SIMULIA

3DEXPERIENCE®

ABAQUS 2025 FD04
ABAQUS/CAE USER'S GUIDE

DASSAULT SYSTEMES

Contents

Trademarks and Legal Notices....16
Abaqus/CAE User's Guide....17
What's New....18
Interacting with Abaqus/CAE....22
Introduction....23
Using this guide....23
Typographical conventions....23
Basic mouse actions....24
The basics of interacting with Abaqus/CAE....26
Starting and exiting Abaqus/CAE....27
The Abaqus/CAE main window....34
What is a module?....47
What is a toolset?....49
Using the mouse with Abaqus/CAE....51
Getting help....52
Understanding Abaqus/CAE windows, dialog boxes, and toolboxes....59
Using the prompt area during procedures....60
Interacting with dialog boxes....64
Understanding and using toolboxes and toolbars....90
Managing objects....93
Working with the Model Tree and the Results Tree....116
Understanding Abaqus/CAE GUI settings....129
Managing viewports on the canvas....130
Understanding viewports....131
Manipulating viewports and viewport annotations....136
Displaying the drawing area in full screen mode....141
Working with viewports....142
Working with viewport arrow and text annotations....151
Linking viewports for view manipulation....173
Working with background images and movies in viewports....177
Manipulating the view and controlling perspective....183
Understanding camera modes and view options....184
Understanding the view manipulation tools....189
The 3D compass....207
Customizing the view triad....214
Controlling perspective....216
Using the view manipulation tools....217
Using the 3Dconnexion motion controllers with Abaqus/CAE....233
Selecting objects within the viewport....236
Understanding selection within viewports....237
Selecting objects within the current viewport....242
Using the selection options....261
Configuring graphics display options....270

Using graphics display options....271
Using display lists....272
Using antialiasing....273
Choosing a highlight method....274
Choosing a translucency mode....275
Controlling drag mode....276
Choosing background colors....277
Printing viewports....278
Understanding printing....279
Controlling the destination and appearance of printed images....287
Working with Abaqus/CAE Model Databases, Models, and Files....303
Understanding and working with Abaqus/CAE models, model databases, and files....304
What is an Abaqus/CAE model database?....305
What is an Abaqus/CAE model?....306
Accessing an output database on a remote computer....310
Understanding the files generated by creating and analyzing a model....314
Abaqus/CAE command files....317
Using the File menu....324
Managing model and output databases....326
Managing models....341
Managing session objects and session options....348
Controlling the input file generated by Abaqus/CAE....352
Managing macros....359
Importing and exporting geometry data and models....361
Importing files into and exporting files from Abaqus/CAE....362
Valid parts, precise parts, and tolerance....377
Controlling the import process....382
Understanding the contents of an IGES file....386
What can you import from a model?....391
A logical approach to successful import of IGES files....399
Importing sketches and parts....400
Importing a model....430
Exporting geometry, model, and mesh data....433
Creating and analyzing a model using the Abaqus/CAE modules....441
The Part module....442
Understanding the role of the Part module....443
Entering and exiting the Part module....444
What is feature-based modeling?....445
How is a part defined in Abaqus/CAE?....454
Copying a part....458
What are orphan nodes and elements?....459
Modeling rigid bodies and display bodies....460
The reference point and point parts....468
What types of features can you create?....471
Using feature-based modeling effectively....487
Capturing your design and analysis intent....489
What is part and assembly locking?....491
What are extruding, revolving, and sweeping?....492

What is lofting?....505
Using the Sketcher in conjunction with the Part module....514
Understanding toolsets in the Part module....515
Using the Part module toolbox....528
Managing parts....529
Using the Create Part dialog box....534
Adding a feature to a part....541
Adding a solid feature....542
Adding a shell feature....556
Adding a wire feature....572
Adding a cut feature....581
Using the Edit Feature dialog box....597
Using the Edit Loft dialog box....599
Blending edges....603
Mirroring a part....607
The Property module....608
Entering and exiting the Property module....609
Understanding properties....610
Which properties can I assign to a part?....619
Understanding the Property module editors....621
Using material libraries....637
Using the Property module toolbox....644
Creating and editing materials....645
Defining general material data....659
Defining mechanical material models....666
Defining thermal material models....852
Defining electrical and magnetic material models....861
Defining other types of material models....868
Creating and editing sections....899
Creating and editing composite layups....951
Assigning sections, orientations, normals, and tangents to a part....977
Using discrete orientations for material orientations and composite layup orientations....995
Creating material calibrations....997
Using the Special menu in the Property module....1011
Using the Query toolset to obtain assignment information....1012
The Assembly module....1014
Understanding the role of the Assembly module....1015
Entering and exiting the Assembly module....1016
Working with part instances....1017
Working with model instances....1026
Creating the assembly....1029
Creating patterns of instances....1043
Performing Boolean operations on part instances....1045
Understanding toolsets in the Assembly module....1052
Using the Assembly module toolbox....1061
Creating and manipulating part and model instances....1062
Applying constraints to part and model instances....1082
Using the Query toolset to query the assembly....1096

The Step module....1097
Understanding the role of the Step module....1098
Entering and exiting the Step module....1099
Understanding steps....1100
Understanding output requests....1107
Understanding integrated, restart, diagnostic, and monitor output....1116
Understanding ALE adaptive meshing....1121
How can I customize the Abaqus analysis controls?....1122
Using the Step module toolbox....1125
Using the Step Manager....1126
Using the step editor....1133
Configuring analysis procedure settings....1137
Defining output requests....1226
Requesting specialized output....1238
Customizing ALE adaptive meshing....1247
Customizing the Abaqus analysis controls....1256
The Interaction module....1264
Understanding the role of the Interaction module....1265
Entering and exiting the Interaction module....1267
Understanding interactions....1268
Understanding interaction properties....1272
Understanding constraints....1274
Understanding contact and constraint detection....1276
Understanding connectors....1299
Understanding connector sections and functions....1300
Understanding Interaction module managers and editors....1305
Understanding symbols that represent interactions, constraints, and connectors....1318
Using the Interaction module toolbox....1319
Using the Interaction module....1320
Using the interaction editors....1350
Using the interaction property editors....1436
Using the constraint editors....1475
Using contact and constraint detection....1495
Using the connector section editors....1516
Using the Query toolset to obtain connector assignment information....1547
The Load module....1548
Understanding the role of the Load module....1549
Entering and exiting the Load module....1550
Managing prescribed conditions....1551
Creating and modifying prescribed conditions....1553
Understanding symbols that represent prescribed conditions....1556
Transferring results between Abaqus analyses....1565
Using the Load module toolbox....1567
Using the Load module....1568
Using the load editors....1579
Using the boundary condition editors....1633
Using the predefined field editors....1674
The Mesh module....1698

Understanding the role of the Mesh module....1699
Entering and exiting the Mesh module....1700
Mesh module basics....1701
Understanding seeding....1715
Assigning Abaqus element types....1726
Verifying and improving meshes....1734
Understanding mesh generation....1746
Structured meshing and mapped meshing....1762
Swept meshing....1774
Free meshing....1789
Bottom-up meshing....1799
Mesh-geometry association....1834
Understanding adaptive remeshing....1836
Advanced meshing techniques....1842
Using the Mesh module toolbox....1852
Seeding a model....1853
Creating and deleting meshes....1867
Controlling mesh characteristics....1871
Obtaining mesh information and statistics....1897
Creating a mesh part....1904
Controlling adaptive remeshing....1905
The Optimization module....1915
Understanding the role of the Optimization module....1916
Entering and exiting the Optimization module....1917
Understanding optimization....1918
Using the Optimization module toolbox....1926
Viewing and troubleshooting an optimization....1927
Creating and configuring an optimization task....1929
Configuring design responses....1966
Creating objective functions....1978
Creating constraints....1979
Configuring geometric restrictions....1981
Creating local stop conditions....2039
The Job module....2041
Understanding the role of the Job module....2042
Understanding analysis jobs....2043
Understanding adaptivity processes....2057
Understanding co-executions....2064
Understanding optimization processes....2069
Restarting an analysis....2080
Creating, editing, and manipulating jobs....2091
Using the job editor....2099
Creating, editing, and manipulating adaptivity processes....2111
Using the adaptivity process editor....2117
Creating, editing, and manipulating co-executions....2124
Creating, editing, and manipulating optimization processes....2130
The Sketch module....2144
Understanding the role of the Sketch module....2145

Entering and exiting the Sketch module....2146
Using the Sketch module....2147
Basic Sketcher concepts....2150
Sketcher geometry....2162
Specifying precise geometry....2166
Controlling sketch geometry....2167
Modifying, copying, and offsetting objects....2174
Customizing the Sketcher....2189
Sketching simple objects....2204
Creating construction geometry....2220
Constraining, dimensioning, and parameterizing a sketch....2229
Editing dimensions....2238
Adding reference geometry....2240
Projecting edges onto a sketch....2241
Moving or copying sketch geometry....2242
Modifying objects....2247
Repairing short edges, gaps, and overlaps....2254
Creating patterns, offsetting, and deleting objects....2259
Undoing and redoing sketching actions....2266
Resetting the view....2267
Managing stand-alone sketches....2268

Modeling techniques....2273

Adhesive joints and bonded interfaces....2274
Modeling adhesive joints and bonded interfaces....2275
Embedding cohesive elements in an existing three-dimensional mesh....2276
Creating a model with cohesive elements using geometry and mesh tools....2277
Defining tie constraints between the cohesive layer and the surrounding bulk material....2279
Assigning cohesive modeling data....2280
Bolt loads....2281
Understanding bolt loads....2282
Creating and editing bolt loads....2285
Composite layups....2287
An overview of composite layups....2288
Creating a composite layup....2290
Understanding composite layups and orientations....2293
Understanding composite layups and distributions....2296
Requesting output from a composite layup....2297
Viewing a composite layup....2299
Connectors....2304
Modeling connectors....2305
What is a connector?....2306
What is a connector section?....2307
What is a CORM?....2308
What are connector behaviors?....2309
Creating the connector geometry, connector sections, and connector section assignments....2310
What is the relationship between reference points and connectors?....2312
Defining connector orientations in connector section assignments....2314
Requesting output from connectors....2315

Applying connector loads and connector boundary conditions....2316
Displaying connectors and connector output in the Visualization module....2317
Continuum shells....2319
Modeling continuum shells....2320
Meshing parts with continuum shell elements....2321
Co-simulation....2322
Overview of co-simulation....2323
What is a co-simulation?....2324
Linking and excluding part instances for a co-simulation....2325
Ensuring matching nodes at the interface regions....2326
Specifying the interface region and coupling schemes....2327
Identifying the models involved and specifying job parameters....2328
Viewing the results of the co-simulation....2329
Display bodies....2330
What is a display body?....2331
Should I mesh a display body?....2333
Displaying display bodies in the Visualization module....2334
Eulerian analyses....2335
Overview of Eulerian analyses....2336
Assembling coupled Eulerian-Lagrangian models in Abaqus/CAE....2338
Defining contact in Eulerian-Lagrangian models....2340
Assigning materials to Eulerian part instances....2341
Using the volume fraction tool....2345
Eulerian mesh motion....2348
Viewing output from Eulerian analyses....2349
Fasteners....2352
About fasteners....2353
Managing fasteners....2369
Creating point-based fasteners....2370
Creating discrete fasteners....2377
Creating assembled fasteners....2379
Fracture mechanics....2384
Seam cracks....2385
Using contour integrals to model fracture mechanics....2388
Using the extended finite element method to model fracture mechanics....2408
Using the virtual crack closure technique to model crack propagation....2420
Managing cracks....2424
Gaskets....2425
Modeling gaskets....2426
Defining materials for gaskets....2429
Assigning gasket elements to a region....2430
Imperfection....2431
Defining imperfections....2432
Managing imperfections....2433
Defining file imperfections....2434
Defining input imperfections....2435
Defining data imperfections....2436
Editing the region to which a file imperfection applied....2437

Inertia....2438
Defining inertia....2439
Managing inertia....2440
Defining point mass and rotary inertia....2441
Defining nonstructural mass....2444
Defining heat capacitance....2446
Editing the region to which inertia is applied....2447
Load cases....2448
What is a load case?....2449
Managing load cases....2450
Load case editors....2451
Viewing load case output....2453
Defining a load case....2455
Midsurface modeling....2457
Understanding midsurface modeling....2458
Understanding the reference representation....2459
Creating shell faces for the midsurface model....2460
Examples of creating a midsurface model....2464
Creating and editing midsurface models....2477
Skin and stringer reinforcements....2482
Defining skin reinforcements....2483
Defining stringer reinforcements....2485
Managing skin and stringer reinforcements....2487
Generating elements on a skin or stringer reinforcement....2488
Assigning element types to skin or stringer reinforcements....2489
Using offset meshes to create skin reinforcements....2490
Assigning surface properties to skins and stringers....2491
Creating and editing skin reinforcements....2492
Creating and editing stringer reinforcements....2494
Springs and dashpots....2496
Defining springs and dashpots....2497
Managing springs and dashpots....2498
Creating springs and dashpots connecting two points....2499
Editing springs and dashpots connecting two points....2502
Creating springs and dashpots connecting points to ground....2503
Editing the region to which springs and dashpots connecting points to ground are applied....2505
Submodeling....2506
Analyzing the global model....2507
Creating a submodel....2508
Removing regions....2509
Creating the submodel boundary condition....2510
Creating the submodel load....2512
Modifying the submodel....2513
Analyzing the submodel....2514
Checking the results from the submodel....2515
Substructures....2516
Using substructures in Abaqus/CAE....2517
Generating a substructure....2520

Specifying the retained nodal degrees of freedom and load cases for a substructure....2521
Importing a substructure into Abaqus/CAE....2522
Using substructure part instances in an assembly....2523
Activating load cases during substructure usage....2524
Recovering field output for substructures....2525
Visualizing substructure output....2526
Viewing results....2527
Visualization module basics....2528
Understanding the role of the Visualization module....2529
Entering and exiting the Visualization module....2532
Understanding plot states and plot customization....2533
Understanding toolsets in the Visualization module....2538
Understanding Visualization module performance....2539
Viewing diagnostic output....2540
Understanding job diagnostics....2541
Generating diagnostic information....2542
Interpreting diagnostic information....2543
Accessing diagnostic information....2552
Selecting model data and analysis results to plot....2553
Selecting results from an output database....2554
Selecting results from the current model database....2555
Selecting the results step and frame....2557
Customizing the display of steps and frames in the results....2560
Selecting the field output to display....2565
Selecting result options....2585
Creating and saving new field output....2603
Creating coordinate systems during postprocessing....2613
Creating node and element sets during postprocessing....2620
Plotting the undeformed and deformed shapes....2621
Understanding undeformed shape plotting....2622
Understanding deformed shape plotting....2623
Using common plot options....2624
Producing an undeformed shape plot....2625
Producing a deformed shape plot....2626
Superimposing deformed and undeformed model plots....2627
Contouring analysis results....2629
Understanding contour plotting....2630
Using contour plot options....2636
Producing a contour plot....2637
Producing a contour plot of linear beam section stresses....2638
Customizing a contour plot....2640
Plotting analysis results as symbols....2659
Understanding symbol plotting....2660
Using symbol plot options....2662
Producing a results symbol plot....2663
Producing a symbol plot of free body nodal forces....2664
Customizing symbol plot appearance....2665
Plotting material orientations....2670

Understanding material orientation plotting....2671
Using material orientation plot options....2672
Producing a material orientation plot....2673
Customizing material orientation plot triads....2674
X–Y plotting....2675
Understanding X–Y plotting....2676
Specifying and saving X–Y data objects....2683
Producing an X–Y plot....2702
Operating on saved X–Y data objects....2704
Customizing X–Y plot axes....2784
Customizing X–Y curve appearance....2799
Customizing X–Y plot appearance....2808
Customizing the X–Y plot title....2814
Customizing the appearance of the X–Y plot legend....2818
Customizing border and fill colors for an X–Y plot....2825
Viewing results along a path....2826
Understanding results along a path....2827
Creating a path through your model....2832
Obtaining X–Y data along a path....2843
Animating plots....2849
Understanding animation....2850
Object-based animation....2854
Saving an animation file....2861
Controlling animation playback....2867
Querying the model in the Visualization module....2876
Understanding the Query toolset in the Visualization module....2877
Using the Query toolset....2879
Probing the model....2884
Understanding probing....2885
Using the Query toolset to probe the model....2887
Calculating linearized stresses....2897
Understanding stress linearization....2898
Stress linearization example....2902
Obtaining linearized stress results....2906
Viewing a ply stack plot....2909
Understanding ply stack plots....2910
Customizing ply stack plots....2912
Generating tabular data reports....2919
Producing a tabular report....2920
Using tabular report options....2921
Selecting report data....2922
Specifying your report file name....2928
Controlling report layout, width, format, and coordinate system....2929
Sorting field output data....2931
Formatting report values....2932
Reporting data values, minimums, maximums, and totals....2934
Customizing plot display....2935
Selecting plot display options....2936

Customizing render style, translucency, and fill color....2940
Customizing element and surface edges....2945
Customizing model shape....2952
Customizing model labels....2956
Displaying multiple plot states....2967
Displaying element and surface normals....2968
Customizing the appearance of display bodies....2969
Customizing camera movement....2970
Controlling the display of model entities....2972
Controlling the display of constraints in the Visualization module....2974
Customizing general model display....2975
Customizing viewport annotations....2994
Customizing the legend....2995
Customizing the title block....2998
Customizing the state block....3000
Using viewport annotation options....3002
Using toolsets....3003
The Amplitude toolset....3004
Understanding the role of the Amplitude toolset....3005
Understanding the amplitude editors....3006
Selecting an amplitude type to define....3007
Using tabular data to define an amplitude curve....3009
Using equally spaced data to define an amplitude curve....3010
Using periodic data to define an amplitude curve....3011
Using modulated data to define an amplitude curve....3012
Using exponential decay data to define an amplitude curve....3013
Defining a solution-dependent amplitude curve....3014
Using smooth step data to define an amplitude curve....3015
Defining a spectrum....3016
Defining a user-defined amplitude curve....3017
Defining a PSD definition....3018
The Analytical Field toolset....3019
Using the Analytical Field toolset....3020
Using analytical expression fields....3021
Using analytical mapped fields....3026
Displaying symbols for interactions and prescribed conditions that use analytical fields....3037
Displaying symbols to visualize mapping source data....3040
Creating expression fields....3041
Creating mapped fields....3043
The Attachment toolset....3053
Understanding attachment points and lines....3054
Understanding the projection methods....3056
Creating attachment points by picking or reading from a file....3057
Creating attachment points by choosing a direction and a spacing....3059
Creating patterns of attachment points based on edges....3061
Creating attachment lines by projecting points....3063
Editing attachment points and lines....3065
Removing attachment points and lines....3069

The CAD Connection toolset....3070
Creating a CAD connection....3071
Updating geometry parameters in an imported model....3074
The Customize toolset....3076
Configuring the visibility of toolbars....3077
Configuring keyboard shortcuts....3078
Custom toolbars....3079
Configuring icons in custom toolbars....3080
Using the Customize toolset....3081
The Datum toolset....3089
Understanding the role of datum geometry....3090
Using the Datum toolset....3092
Why are datum coordinate systems so important?....3093
Understanding a datum as a feature....3094
Datum creation techniques....3095
Creating datum points....3109
Creating datum axes....3118
Creating datum planes....3130
Creating datum coordinate systems....3140
The Discrete Field toolset....3146
Using the Discrete Field toolset....3147
Creating discrete fields....3148
Evaluating discrete fields....3150
Creating discrete fields for material volume fractions....3153
The Edit Mesh toolset....3155
What can I do with the Edit Mesh toolset?....3156
What is the difference between editing an orphan mesh, a meshed part, and a meshed part instance in the assembly?.3165
Meshing strategies and mesh editing techniques....3166
Using the mesh editing tools....3175
Editing nodes....3177
Editing elements....3197
Editing the entire mesh....3217
Editing and refining an orphan mesh....3241
Undoing or redoing a change in the Edit Mesh toolset....3245
The Feature Manipulation toolset....3249
Using the Feature Manipulation toolset....3250
Using the Model Tree to manage features....3252
Tuning feature regeneration....3254
Modifying and manipulating features....3260
The Filter toolset....3269
Filtering field and history data....3270
Applying bounding values to field and history data....3271
Creating a filter....3272
The Free Body toolset....3274
Resultant forces and moments on free body cuts in Abaqus/CAE....3275
Creating or editing a free body cut....3276
Selection methods for free body cross-sections....3279
Displaying, hiding, and highlighting free body cuts....3280

Customizing free body cut display....3281
The Options toolset....3288
Customizing memory limits and regeneration options....3289
Using view manipulation shortcuts....3290
Scaling the size of icons....3291
The Geometry Edit toolset....3292
Editing techniques....3293
Editing techniques....3294
What is stitching?....3302
A strategy for repairing geometry....3303
Creating a part from orphan elements....3305
Editing and repairing edges....3306
Editing and repairing faces....3314
Editing parts....3332
The Partition toolset....3335
Understanding the role of partitions....3336
Using the Partition toolset....3337
Understanding partitions....3338
Partitioning techniques....3342
Partitioning edges....3355
Partitioning faces....3360
Partitioning cells....3377
The Query toolset....3390
Understanding the role of the Query toolset....3391
Querying the model....3400
The Reference Point toolset....3413
What is a reference point?....3414
What is a reference point used for?....3415
Creating a reference point....3416
The Set and Surface toolsets....3417
Understanding the role of the Set and Surface toolsets....3418
Understanding sets and surfaces....3419
Using the Set and Surface toolsets....3428
The Stream toolset....3437
Understanding stream display....3438
Creating a stream....3439
Displaying and hiding streams....3440
Customizing stream display....3441
The Virtual Topology toolset....3444
What is virtual topology?....3445
What can I do with the Virtual Topology toolset?....3448
What can I do with a part or a part instance containing virtual topology?....3450
Why repair a part if I can use virtual topology?....3451
Creating virtual topology based on geometric parameters....3452
Using the Virtual Topology toolset....3457
Customizing model display....3464
Customizing geometry and mesh display....3465
Using geometry and mesh display options....3466

Choosing a render style....3467
Controlling edge visibility....3469
Controlling curve refinement....3472
Defining mesh feature edges....3473
Controlling translucency for substructure parts....3474
Controlling beam profile display....3475
Controlling shell thickness display....3477
Controlling datum display....3479
Controlling the display of individual coordinate systems....3480
Controlling reference point display....3481
Customizing mesh display....3482
Controlling model lighting....3483
Controlling instance visibility....3485
Controlling the display of attributes....3486
Saving your display options settings....3488
Color coding geometry and mesh elements....3489
Understanding color coding....3490
Changing the initial color....3494
Changing the translucency....3495
Coloring geometry and mesh elements....3496
Coloring all geometry in the Visualization module....3497
Coloring nodes or elements in the Visualization module....3498
Coloring constraints in the Visualization module....3500
Customizing the display color of individual objects....3501
Displaying multiple color mappings....3503
Editing the colors in the Auto-Color List....3504
Saving and restoring custom color mappings....3505
Using display groups to display subsets of your model....3506
Understanding display groups....3507
Managing display groups....3512
Overlaying multiple plots....3523
Understanding how to overlay plots....3524
Producing and modifying overlay plots....3526
Cutting through a model....3533
Understanding view cuts....3534
Managing view cuts....3539
Using plug-ins....3554
The Plug-in toolset....3555
What is a plug-in?....3556
Where can I get plug-ins?....3557
How can I get information about a plug-in?....3558
An example of a Python module and a function....3559
What can I do with a GUI plug-in?....3560
How do I make a plug-in available from Abaqus/CAE?....3561
How are kernel plug-ins executed?....3567
Overwriting plug-ins....3568
How are GUI plug-ins executed?....3569
Hiding a plug-in's source code....3570

Displaying exceptions for imported plug-ins at startup....3571
Abaqus/CAE modules and plug-ins....3572
How can I provide information about a plug-in?....3573
Abaqus plug-ins....3574
Running the Getting Started with Abaqus examples....3575
Creating a GUI plug-in....3576
Creating a kernel plug-in....3577
Creating dialog boxes with the Really Simple GUI (RSG) Dialog Builder....3578
Upgrading a script....3579
Displaying modal contribution factors....3582
Displaying the history of the adaptive remeshing error indicator....3593
Generating a report of postprocessing results....3597
Exchanging data between Abaqus/CAE and Microsoft Excel....3599
Importing a model from a file in stereolithography (STL) format....3600
Exporting a part or assembly in stereolithography (STL) format....3601
Plotting amplitude data....3602
Combining data from multiple output databases....3603
Finding the nearest node to a point....3606
Finding the average temperature of a set of elements....3607
Creating a planar constraint....3608
Resetting midside nodes....3609
ODB Reducer/Builder....3610
Additive manufacturing....3613
Appendix....3619
Abaqus keyword browser table....3620
Keyword support from the input file reader....3660
Special element types....3695
Special graphical symbols....3696
Symbols used to represent prescribed conditions....3697
Symbols used to represent interactions, constraints, and connectors....3700
Symbols used to represent special engineering features....3702
Symbols used in the Visualization module....3703
Element and output variable support....3704

Trademarks and Legal Notices

Trademarks

Abaqus, **3D**EXPERIENCE®, the 3DS logo, the Compass icon, IFWE, 3DEXCITE, 3DVIA, BIOVIA, CATIA, CENTRIC PLM, DELMIA, ENOVIA, GEOVIA, MEDIDATA, NETVIBES, OUTSCALE, SIMULIA and SOLIDWORKS are commercial trademarks or registered trademarks of Dassault Systèmes, a European company (Societas Europaea) incorporated under French law, and registered with the Versailles trade and companies registry under number 322 306 440, or its subsidiaries in the United States and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Legal Notices

Abaqus and this documentation may be used or reproduced only in accordance with the terms of the software license agreement signed by the customer, or, absent such agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

Dassault Systèmes or its Affiliates shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

© Dassault Systèmes Americas Corp., 2025.

For a full list of the third-party software contained in this release, please go to the Legal Notices in the Abaqus 2025 HTML documentation, which can be obtained from a documentation installation, or in the SIMULIA Established Products 2025 Program Directory, which is available on *www.3ds.com*.

Abaqus/CAE User's Guide

The Abaqus/CAE User's Guide is a complete reference to using Abaqus/CAE.

This guide is a part of the Abaqus® documentation collection, which describes all the capabilities of the Abaqus finite element analysis technology used in SIMULIA® applications.

Detailed descriptions of how to use Abaqus/CAE for model generation, analysis, and results evaluation and visualization are provided. Abaqus/Viewer users should refer to the information on the Visualization module in this guide.

What's New

This page describes recent changes in Abaqus/CAE.

2025 FD01

Animation File Formats

You can now save and load animations in MP4 and animated GIF formats in Abaqus/CAE.

Benefits: New file formats offer improved compatibility with modern media players and editing software, as well as HTML5 and common applications for producing presentations. The file size can be significantly smaller, with no appreciable loss of quality.
You can save MP4 animations with the MPEG-4 codec. If you have an installation of the FFmpeg program that supports H.264 encoding on your system, you can save MP4 animations using the H.264 codec.

For more information, see *Choosing the animation file format* and *Using background images and movies*.

PDF Guides Available

You can download PDF versions of the Abaqus guides from each guide overview.

Benefits: You can easily access the PDF versions of the guides.
In earlier releases, PDF versions of the guides were updated only for the GA (General Availability) release and could be downloaded from the Dassault Systèmes Knowledge Base. As of 2025 FD01, the PDF guides will be updated with each release of the HTML versions of the guides.

For more information, see *Abaqus/CAE User's Guide*.

2025 GA

AutoCAD DXF Sketch Import

Users can now import polylines, splines, ellipses, and other complex entities from AutoCAD files.

Benefits: The import of `.dxf` files as sketches in Abaqus/CAE is greatly improved.

For more information, see *Importing sketches*.

Wear Modeling in Abaqus/CAE

Abaqus/CAE now supports the wear surface property and its assignment in general contact interactions.

Benefits: The support allows greater flexibility for modeling wear phenomena in Abaqus/CAE without having to use the keyword editor.

For more information, see *Specify wear surface property assignments* and *Defining a wear interaction property*.

Rotordynamic Loads in Abaqus/CAE

Abaqus/CAE now supports rotordynamic loads as a rotational body force type in Abaqus/CAE.

Benefits: This support allows greater flexibility to define a rotordynamic load as a rotational body force in Abaqus/CAE without having to use the keyword editor.

For more information, see *Defining a rotational body force*.

Step Dependence for Fluid Inflator and Fluid Exchange

Abaqus/CAE now supports the activation of fluid inflators and fluid exchange in steps.

Benefits: This support allows greater flexibility for modeling the inflator and cavity exchange in Abaqus/CAE without having to use the keyword editor.
Earlier releases of Abaqus/CAE supported automatic activation in Abaqus/Explicit steps, offering no flexibility to the user.

For more information, see *Defining a fluid inflator activation interaction* and *Defining a fluid exchange activation interaction*.

Modeling Cracks and Seams

Abaqus/CAE now supports the creation of a seam on a dependent part instance.

Benefits: Seam creation on a dependent part instance allows greater flexibility when modeling seams for fracture simulation in Abaqus/CAE.
Earlier releases of Abaqus/CAE supported seam creation only for independent instances.

For more information, see *The Special menu in the Interaction module*.

2024 FD03

State Space Solver for Transient Modal Dynamic Analysis

The new state space solver for transient modal dynamic analysis of systems with nondiagonal modal damping operators is now the default solver in Abaqus/CAE.

Benefits: The state space solver is unconditionally stable and offers improved performance for models with many time increments and a moderate number of modes.
In earlier releases, the conventional solver was the default for transient modal dynamic analysis in Abaqus/CAE. Now, you must use the keywords editor to access the conventional solver in Abaqus/CAE.

For more information, see *Transient Modal Dynamic Analysis*.

Job Execution Using **3D**EXPERIENCE On-Cloud Licensing

When you launch Abaqus/CAE from the **3D**EXPERIENCE Platform Connector for Abaqus/CAE, you can now specify that jobs should execute using the **3D**EXPERIENCE Platform license server instead of a locally installed license server.

Benefits: You can take advantage of available cloud SIM Unit licenses to execute an Abaqus analysis locally.

For more information, see *Setting the license type for the SimUnit license model*.

2024 GA

Python 3

Abaqus Python and Abaqus/CAE are now based on Python 3.10 instead of Python 2.7.

Benefits: This update brings Abaqus Python into conformance with a version of Python that is supported in the open-source community.

For more information, see *https://go.3ds.com/AbaqusPython*.

Render Styles in Plots for Continuum Particle Elements

Abaqus/CAE now supports visualization of element particle edges with wireframe, hidden, filled, and shaded render styles (**View > ODB Display Options**).

Benefits: The ability to assign different render styles makes it easier to distinguish between continuum particle elements.

For more information, see *Output*, *Output*, and *Controlling the display of model entities*.

Step-Dependent General Contact

Abaqus/CAE now supports the definition of general contact in Abaqus/Standard in any analysis step or in the initial step.

Benefits: Defining general contact in analysis steps enables definition of more complex contact behaviors in Abaqus/Standard simulations.

For more information, see *Defining general contact* and *About General Contact in Abaqus/Standard*.

Contact Mass Scaling

Abaqus/CAE now supports the definition of contact mass scaling in Abaqus/Explicit.

Benefits: Contact mass scaling can reduce the number of increments required for Abaqus/Explicit simulations that use nondefault penalty stiffness.
You can apply mass scaling to the contact surfaces or elements involved in the contact definition. Contact mass scaling reduces the impact of higher-than-default contact penalty stiffness on the global stable time increment in an Abaqus/Explicit analysis by increasing the mass associated with the contact surfaces.

For more information, see *Defining contact mass scaling* and **CONTACT MASS SCALING*.

Channel and Hat Section Profiles

You can now define channel and hat section profiles in the Property module.

Benefits: The addition of channel and hat profiles simplifies the specification of these common beam cross-sections in Abaqus/CAE.

For more information, see *Defining a channel profile*, *Defining a hat profile*, *ChannelProfile object*, *HatProfile object*, and *Beam cross-section analysis*.

Beam Section Offset for *BEAM SECTION

Abaqus/CAE now supports the definition of a beam section offset for beam sections integrated during the analysis (*BEAM SECTION).

Benefits: Beam offsets allow greater flexibility when modeling with beam elements in Abaqus/CAE. Earlier releases of Abaqus/CAE supported beam section offsets only for **BEAM GENERAL SECTION*.

For more information, see **BEAM SECTION*, *Using a Beam Section Integrated during the Analysis to Define the Section Behavior*, and *BeamSection object*.

Rotational Couplings

You can now define a rotational coupling type in discrete fasteners and coupling constraints in Abaqus/CAE.

Benefits: Defining the rotational coupling type gives you greater control over the behavior of distributing couplings involving cloud nodes with rotational degrees of freedom.

For more information, see **DISTRIBUTING*, *Creating discrete fasteners*, and *Defining coupling constraints*.

Using the Linearized Contact Capability to Solve for Contact Status and Contact Stresses

Abaqus/CAE now supports the activation of the linearized contact capability to solve for contact status and contact stresses within a static perturbation step with small-sliding, frictionless contact.

Benefits: The Linear Complementarity Problem (LCP) solution technique provides a faster solution for certain classes of contact problem.

For more information, see *Configuring a static, linear perturbation procedure* and **SOLUTION TECHNIQUE*.

Interacting with Abaqus/CAE

This guide is the main reference document for Abaqus/CAE, including Abaqus/Viewer.

Abaqus/CAE

Abaqus/CAE is a complete Abaqus environment that provides a simple, consistent interface for creating, submitting, monitoring, and evaluating results from Abaqus/Standard and Abaqus/Explicit simulations. Abaqus/CAE is divided into modules, where each module defines a logical aspect of the modeling process; for example, defining the geometry, defining material properties, and generating a mesh. As you move from module to module, you build the model from which Abaqus/CAE generates an input file that you submit to the Abaqus/Standard or Abaqus/Explicit analysis product. The analysis product performs the analysis, sends information to Abaqus/CAE to allow you to monitor the progress of the job, and generates an output database. Finally, you use the Visualization module of Abaqus/CAE (also licensed separately as Abaqus/Viewer) to read the output database and view the results of your analysis.

Abaqus/Viewer

Abaqus/Viewer provides graphical display of Abaqus finite element models and results. Abaqus/Viewer is incorporated into Abaqus/CAE as the Visualization module.

This part of the guide introduces you to the Abaqus/CAE working environment.

In this section:

- *Introduction*
- *The basics of interacting with Abaqus/CAE*
- *Understanding Abaqus/CAE windows, dialog boxes, and toolboxes*
- *Managing viewports on the canvas*
- *Manipulating the view and controlling perspective*
- *Selecting objects within the viewport*
- *Configuring graphics display options*
- *Printing viewports*

Introduction

This guide is a complete reference to using Abaqus/CAE.

This section provides information about the contents of this guide and the typographical conventions used.

Using this guide

This guide is a complete reference to using Abaqus/CAE (including Abaqus/Viewer, a subset of Abaqus/CAE that contains only the Visualization module).
In general, any references to the Visualization module throughout this guide apply equally to Abaqus/Viewer.

The Abaqus/CAE user interface is very intuitive and allows you to begin working without a great deal of preparation. However, you may find it useful to read through the tutorials at the end of the *Getting Started with Abaqus/CAE* guide before using the product for the first time. Only *Viewing the Output from Your Analysis* applies if you are running Abaqus/Viewer.

This guide is divided into the following parts:

- *Interacting with Abaqus/CAE* contains general information on the user interface
- *Working with Abaqus/CAE Model Databases, Models, and Files* contains information on the various files created by and used with Abaqus/CAE
- *Creating and analyzing a model using the Abaqus/CAE modules* discusses each of the Abaqus/CAE modules in detail, except the Visualization module
- *Modeling techniques* discusses how to define special engineering features in an Abaqus/CAE model and discusses modeling techniques that span multiple Abaqus/CAE modules.
- *Viewing results* discusses the Visualization module (Abaqus/Viewer) in detail
- *Using toolsets* contains information on the toolsets in all Abaqus/CAE modules except the Visualization module (discussed in *Viewing results*)
- *Customizing model display* contains customization information
- *Using plug-ins* discusses how you can use plug-ins and the Plug-in toolset to extend the capabilities of Abaqus/CAE.

Abaqus keyword browser table and *Keyword support from the input file reader* provide tables that you can use to determine which Abaqus/CAE module embodies the functionality of a particular Abaqus keyword, as well as whether a particular keyword is supported. *Special element types* lists element types used in Abaqus for model features that are not part of the mesh. *Special graphical symbols* explains how to interpret the special graphical symbols used by Abaqus/CAE. *Element and output variable support* lists the Abaqus output variables that are not supported by the Visualization module.

Typographical conventions

This guide adheres to a set of typographical conventions so that you can recognize actions and items.

The following list illustrates each of the conventions:

- Text you enter from the keyboard or that Abaqus/CAE outputs: `crankshaft_steel`, `1.35E10`
- Labels of items on the screen: **Job Manager**
- Hyperlinks: *click here*
- Keyboard actions: **[Shift]**
- Keystroke combinations (two keys that must be pressed simultaneously): **[Alt] + F**
- Compound keyboard/mouse actions: **[Shift] + Click**

- Text indicating that the user has a choice: *odb_file*, **Options->*plot state***
- Menu selections and tabs within dialog boxes:

  View->Graphics Options->Hardware

Basic mouse actions

Figure 1 shows the mouse button orientation for a left-handed and a right-handed 3-button mouse.

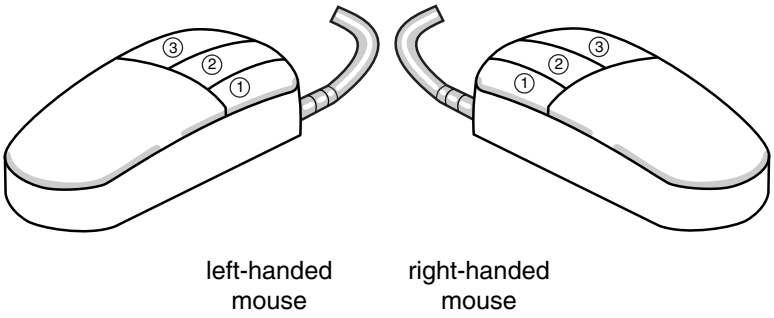


Figure 1: Mouse buttons.

The following terms describe actions you perform using the mouse:

Click

Press and quickly release the mouse button. Unless otherwise specified, the instruction “click” means that you should click mouse button 1.

Drag

Press and hold down mouse button 1 while moving the mouse.

Point

Move the mouse until the cursor is over the desired item.

Select

Point to an item and then click mouse button 1.

[Shift] + Click

Press and hold the **[Shift]** key, click mouse button 1, and then release the **[Shift]** key.

[Ctrl] + Click

Press and hold the **[Ctrl]** key, click mouse button 1, and then release the **[Ctrl]** key.

Abaqus/CAE is designed for use with a 3-button mouse. Accordingly, this guide refers to mouse buttons 1, 2, and 3 as shown in *Figure 1*. However, you can use Abaqus/CAE with a 2-button mouse as follows:

- The two mouse buttons are equivalent to mouse buttons 1 and 3 on a 3-button mouse.
- Pressing both mouse buttons simultaneously is equivalent to pressing mouse button 2 on a 3-button mouse.



Tip: You are instructed to click mouse button 2 in procedures throughout this guide. Make sure that you configure mouse button 2 (or the wheel button) to act as a middle button click.

The basics of interacting with Abaqus/CAE

Before you can begin creating and analyzing a model or interpreting analysis results, it is helpful to become familiar with the basics of interacting with Abaqus/CAE. This chapter introduces you to the user interface.

In this section:

- *Starting and exiting Abaqus/CAE*
- *The Abaqus/CAE main window*
- *What is a module?*
- *What is a toolset?*
- *Using the mouse with Abaqus/CAE*
- *Getting help*

Starting and exiting Abaqus/CAE

This section explains how to start and how to exit Abaqus/CAE.

In this section:

- *Starting Abaqus/CAE (or Abaqus/Viewer)*
- *Exiting an Abaqus/CAE session*
- *Working with abaqus_2025.gpr files*
- *Saving model data from an inactive session*

Starting Abaqus/CAE (or Abaqus/Viewer)

When you create a model and analyze it, Abaqus/CAE generates a set of files containing the definition of your model, the analysis input, and the results of the analysis. In addition, Abaqus/CAE and Abaqus/Viewer generate replay files that reflect all your interactions with the application.
Consequently, before you run either product, you should move to a directory where you have permission to create files.

You execute Abaqus/CAE (or Abaqus/Viewer) by running the **abaqus** execution procedure and specifying the **cae** (or **viewer**) parameter:

abaqus cae or **viewer[database =** *database-file*]**[replay =** *replay-file*]**[recover =** *journal-file*]**[startup =** *startup-file*]**[script =** *script-file*]**[noGUI =** *noGUI-file*]**[noenvstartup][noSavedOptions][noSavedGuiPrefs][noStartupDialog][custom =** *script-file*]**[[guiTester][**GUI-script]]**[guiRecord][guiNoRecord]**

You can include the following options on the command line:

database

This option specifies the name of the model database file or output database file to open. You can open either type of file in Abaqus/CAE; you can open only output database files in Abaqus/Viewer. To specify a model database file, include either the `.cae` file extension or no file extension in your file name. To specify an output database file when running Abaqus/CAE, include the `.odb` file extension in your file name. If you are running Abaqus/Viewer, you can omit the `.odb` file extension.

replay

This option specifies the name of the file from which Abaqus/CAE commands are to be replayed. The commands in *replay-file* will execute immediately upon startup of Abaqus/CAE. You cannot use the `replay` option to execute a script with control flow statements. For more information, see *Replaying an Abaqus/CAE session*.

recover

This option specifies the name of the file from which a model database is to be rebuilt; it is not available if you are running Abaqus/Viewer. The commands in *journal-file* (*model_database_name* `.jnl`) will execute immediately upon startup of Abaqus/CAE. For more information, see *Recreating a saved model database*, and *Recreating an unsaved model database*.

startup

This option specifies the name of the file containing Python configuration commands to be run at application startup. Commands in this file are run after any configuration commands that have been set in the environment file. Abaqus/CAE does not echo the commands to the replay file when they are executed.

Arguments can be passed into the file by entering `--` on the command line, followed by the arguments separated by one or more spaces. These arguments will be ignored by the Abaqus/CAE execution procedure, but they will be accessible within the script.

script

Same as **startup**.

noGUI

This option specifies the name of a file containing Python scripts to be run without the graphical user interface (GUI). This option is useful for automating pre- or post-analysis processing tasks without the added expense of running a display. Since no interface is provided, the scripts cannot include any user interaction. Abaqus/CAE runs the commands in the file and exits upon their completion. If no file extension is given, the default extension is `.py`. If you use the `noGUI` option, Abaqus/CAE ignores any other command line options that you provide.

Arguments can be passed into the file by entering `--` on the command line, followed by the arguments separated by one or more spaces. These arguments will be ignored by the Abaqus/CAE execution procedure, but they will be accessible within the Python script. If you are using the `noGUI` option, you can use an argument to pass in a variable that would otherwise be provided by a command line option. For example, you can pass in the name of a file that would otherwise be specified by the `script` option.

A sample usage of the **noGUI** option is available in *Abaqus/CAE Execution*.

noenvstartup

This option specifies that all configuration commands in the environment files should not be run at application startup. This option can be used in conjunction with the **startup** command to suppress all configuration commands except for those in the **startup** file.

noSavedOptions

This option specifies that Abaqus/CAE should not apply the display options settings (for example, the render style and the display of datum planes) stored in the `abaqus_2025.gpr` file. For more information, see *Working with abaqus_2025.gpr files*, and *Saving your display options settings*.

noSavedGuiPrefs

This option specifies that Abaqus/CAE should not apply the GUI options settings (for example, the size and location of the Abaqus/CAE main window or its dialog boxes) stored in the `abaqus_2025.gpr` file.

noStartupDialog

This option specifies that the **Start Session** dialog box for Abaqus/CAE or Abaqus/Viewer should not be displayed.

custom

This option specifies the name of the file containing Abaqus GUI Toolkit commands. This option executes an application that is a customized version of Abaqus/CAE or Abaqus/Viewer. For more information, see *Introduction*.

guiTester

This option starts a separate user interface containing the Abaqus Python development environment along with Abaqus/CAE or Abaqus/Viewer. The Abaqus Python development environment allows you to create, edit, step through, and debug Python scripts. For more information, see *The Abaqus Python Development Environment*.

You can specify a script as the argument for this option, which prompts Abaqus/CAE or Abaqus/Viewer to run a GUI script. Abaqus/CAE or Abaqus/Viewer closes when the end of the script is reached.

guiRecord

This option enables you to record your actions in the Abaqus/CAE or Abaqus/Viewer user interface in a file named `abaqus.guiLog`. Creating a record of your actions in the GUI can help you capture and replay common activities in Abaqus/CAE or Abaqus/Viewer for demonstration or training purposes. You can replicate all of the actions from a `.guiLog` file in Abaqus/CAE or Abaqus/Viewer by running the file in the Abaqus Python Development Environment (PDE); for more information, see *Running a script*.

If desired, you can set **guiRecord** at startup by using the environment variable `ABQ_CAE_GUIRECORD`. The **guiRecord** option cannot be used with the **guiTester** option.

guiNoRecord

This option enables you to disable user interface recording when the environment variable `ABQ_CAE_GUIRECORD` is set.

Abaqus/CAE begins. If you do not include the **database**, **replay**, **recover**, or **noStartupDialog** options, the **Start Session** dialog box appears. Choose one of the following session startup options:

Create Model Database: With Standard/Explicit Model

Use this option (not available if you are running Abaqus/Viewer) to begin a new Abaqus/Standard or Abaqus/Explicit analysis (equivalent to choosing **File->New Model Database->With Standard/Explicit Model** from the main menu bar).

Create Model Database: With Electromagnetic Model

Use this option (not available if you are running Abaqus/Viewer) to begin an electromagnetic analysis (equivalent to choosing **File->New Model Database->With Electromagnetic Model** from the main menu bar).

Open Database

Use this option to open a previously saved model database or output database file (equivalent to choosing **File->Open** from the main menu bar).

Run Script

Use this option to run a file containing Abaqus/CAE commands (equivalent to choosing **File->Run Script** from the main menu bar). For more information, see *Creating and running your own scripts*.

Start Tutorial

Use this option to begin an introductory tutorial from the online documentation (equivalent to choosing **Help->Getting Started** from the main menu bar).

Recent Files

Use this option to open one of the five model database files or output database files that were most recently opened in Abaqus/CAE (equivalent to choosing one of the recent files listed under the **File** menu).

Exiting an Abaqus/CAE session

You can exit the Abaqus/CAE session at any time by selecting **File->Exit** from the main menu bar. If you made any changes to the current model database, Abaqus/CAE asks if you want to save the changes before exiting the session. Abaqus/CAE then closes the current model or output database and all windows and exits the session.

Abaqus/CAE saves your GUI settings; for example, the size of the main window and the size and location of dialog boxes. For more information, see *Working with abaqus_2025.gpr files*, and *Understanding Abaqus/CAE GUI settings*. In addition, Abaqus/CAE automatically creates a file called `abaqus.rpy` that records your operations during the session; you can use this file to reproduce your operations. For more information on reproducing operations and on recovering interrupted sessions, see *Recreating an unsaved model database*.

Additional information

- *Understanding the files generated by creating and analyzing a model*
- *Using the File menu*

Working with abaqus_2025.gpr files

The abaqus_2025.gpr file in your home directory stores GUI settings (such as the size of the main window) as well as display options settings (such as the render style). You can also store display options settings in an abaqus_2025.gpr file in a directory other than your home directory. If you start Abaqus/CAE with **noSavedOptions** specified, Abaqus/CAE does not apply the display options settings (for example, the render style and the display of datum planes) stored in the abaqus_2025.gpr file. For more information, see *Starting Abaqus/CAE (or Abaqus/Viewer)*.

When you start Abaqus/CAE

- GUI settings are read from the abaqus_2025.gpr file in your home directory.
- Display options settings are read from the abaqus_2025.gpr file in the directory from which you start Abaqus/CAE.
  - If no abaqus_2025.gpr file is present but a .gpr file from an earlier release exists in that directory, Abaqus/CAE attempts to apply the settings specified in that file and creates an abaqus_2025.gpr file to store the settings.
  - If no .gpr file is present in that directory, the display options settings are read from the abaqus_2025.gpr file in your home directory.

During an Abaqus/CAE session

You can use **File->Save Display Options** to save display options settings to the abaqus_2025.gpr file in your home directory or in the current directory. For more information, see *Saving your display options settings*. This save option does not apply to GUI settings.

When you exit Abaqus/CAE

Your GUI settings are saved automatically to the abaqus_2025.gpr file in your home directory. For more information, see *Understanding Abaqus/CAE GUI settings*.

You can edit the abaqus_2025.gpr file using API commands in the Abaqus Scripting Interface; for more information, see *Editing display preferences and GUI settings*. You can also delete the file to restore the default GUI and display options settings.

Saving model data from an inactive session

Abaqus/CAE and Abaqus/Viewer include an inactivity timer. If the applications are left inactive for an extended period of time, the license tokens are returned to the server to make them available to other users. Your session does not end if the server connection is lost or if new license tokens cannot be acquired. Instead, when no licenses are available, a dialog box appears listing your options. For both Abaqus/CAE and Abaqus/Viewer you can attempt to reacquire a license or you can exit the application. For Abaqus/CAE you also have the option to save the current model database. Saving the model allows you to preserve any completed model information that you did not already save; any partially completed information, such as for a procedure that was active at the time the license was lost, is not saved. Once you have saved the model database, only the reacquire and exit options remain in the dialog box. The save option is not provided in Abaqus/Viewer since all changes that affect the output database are saved immediately when you make them.

The default time limit is 60 minutes. You can change the time limit by using the **cae_timeout** environment variable in the Abaqus environment file (`abaqus_v6.env`). For more information on the environment file, see *Using the Abaqus environment files*.

Additional information

- *Saving the current model database without a license*
- *License management parameters*

The Abaqus/CAE main window

This section provides an overview of the main window and explains how to operate and manipulate the elements of the window during a session.

In this section:

- *Components of the main window*
- *Components of the main menu bar*
- *Components of the toolbars*
- *The context bar*
- *Components of the viewport*

Components of the main window

You interact with Abaqus/CAE through the main window, and the appearance of the window changes as you work through the modeling process. *Figure 1* shows the components that appear in the main window.

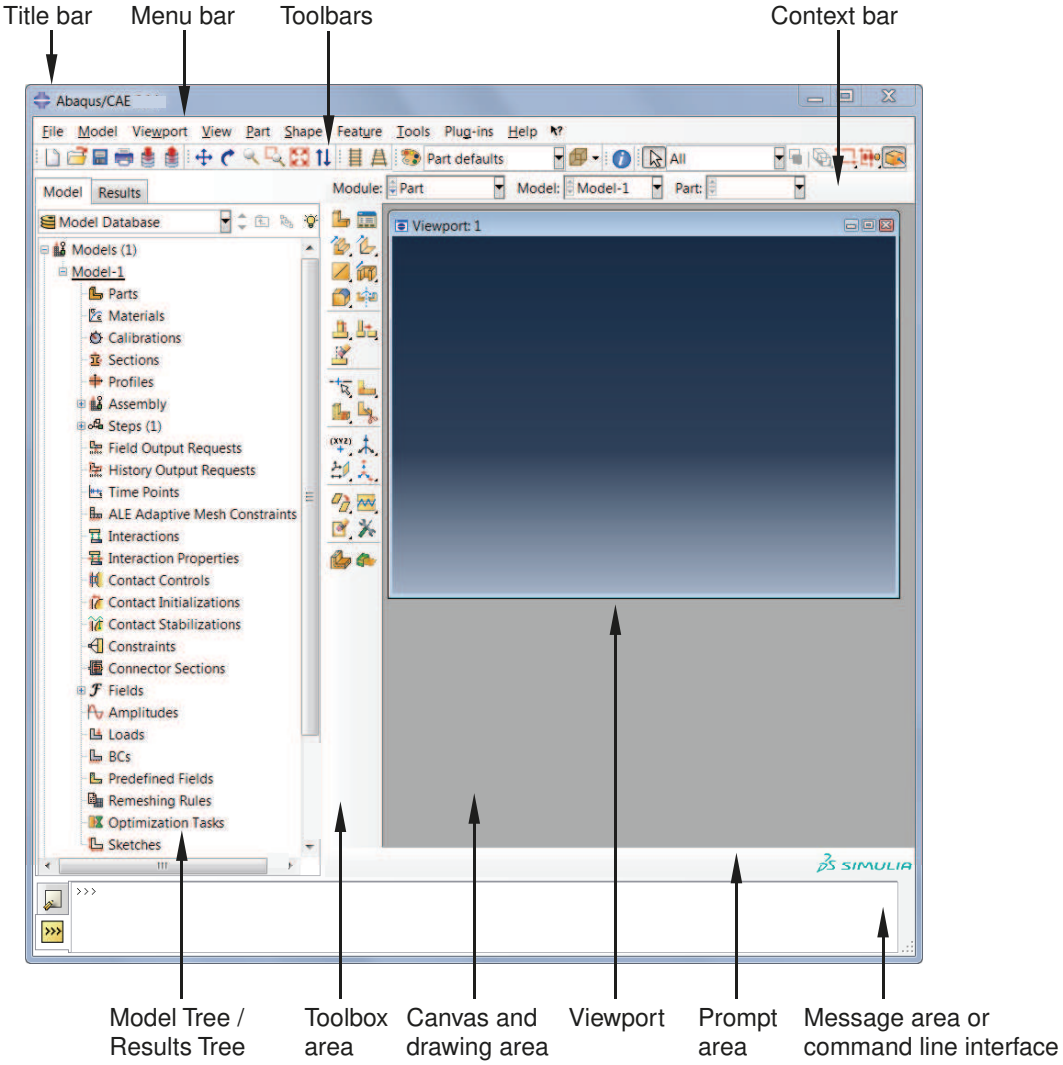


Figure 1: Components of the main window.

The components are:

Title bar

The title bar indicates the release of Abaqus/CAE you are running and the name of the current model database.

Menu bar

The menu bar contains all the available menus; the menus give access to all the functionality in the product. Different menus appear in the menu bar depending on which module you selected from the context bar. For more information, see *Components of the main menu bar*.

Toolbars

The toolbars provide quick access to items that are also available in the menus. For more information, see *Components of the toolbars*.

Context bar

Abaqus/CAE is divided into a set of modules, where each module allows you to work on one aspect of your model; the **Module** list in the context bar allows you to move between these modules. Other items in the context bar are a function of the module you are working in. For example, the context bar allows you to retrieve an existing part while creating the geometry of the model or to change the output database associated with the current viewport. Similarly, in the Mesh module you can choose whether to display the assembly or a particular part. For more information, see *The context bar*.

Model Tree

The Model Tree provides you with a graphical overview of your model and the objects that it contains, such as parts, materials, steps, loads, and output requests. In addition, the Model Tree provides a convenient, centralized tool for moving between modules and for managing objects. If your model database contains more than one model, you can use the Model Tree to move between models. When you become familiar with the Model Tree, you will find that you can quickly perform most of the actions that are found in the main menu bar, the module toolboxes, and the various managers. For more information, see *The Model Tree*.

Results Tree

The Results Tree provides you with a graphical overview of your output databases and other session-specific data such as *X*–*Y* plots. If you have more than one output database open in your session, you can use the Results Tree to move between output databases. When you become familiar with the Results Tree, you will find that you can quickly perform most of the actions in the Visualization module that are found in the main menu bar and the toolbox. For more information, see *The Results Tree*.

Toolbox area

When you enter a module, the toolbox area displays tools in the toolbox that are appropriate for that module. The toolbox allows quick access to many of the module functions that are also available from the menu bar. For more information, see *Understanding and using toolboxes and toolbars*.

Canvas and drawing area

The canvas can be thought of as an infinite screen or bulletin board on which you post viewports; for more information, see *Managing viewports on the canvas*. The drawing area is the visible portion of the canvas. You can display the drawing area full screen using the **View** menu; you can also press **[F11]** to toggle between full screen mode and normal mode.

Viewport

Viewports are windows on the canvas in which Abaqus/CAE displays your model. For more information, see *Managing viewports on the canvas*.

Prompt area

The prompt area displays instructions for you to follow during a procedure; for example, it asks you to select the geometry as you create a set. In the Visualization module a set of buttons is displayed in the prompt area that allow you to move between the steps and the frames of your analysis. For more information, see *Using the prompt area during procedures*.

Message area

Abaqus/CAE prints status information and warnings in the message area. To resize the message area, drag the top edge; to see information that has scrolled out of the message area, use the scroll bar on the right side. The message area is displayed by default, but it uses the same space occupied by the command line interface. If you have recently used the command line interface, you must click  in the bottom left corner of the main window to activate the message area.



Note:

If new messages are added while the command line interface is active, Abaqus/CAE changes the background color surrounding the message area icon to red. When you display the message area, the background reverts to its normal color.

Command line interface

You can use the command line interface to type Python commands and evaluate mathematical expressions using the Python interpreter that is built into Abaqus/CAE. The interface includes primary (>>>) and secondary (`...`) prompts to indicate when you must indent commands to comply with Python syntax. For more information on Python commands, see *The basics of Python*.

The command line interface is hidden by default, but it uses the same space occupied by the message area. Click  in the bottom left corner of the main window to switch from the message area to the command line interface.

Additional information

- *The basics of interacting with Abaqus/CAE*

Components of the main menu bar

When you start a session, the menus listed below appear on the main menu bar. Abaqus/CAE displays additional menu options and provides access to toolsets depending on the current module in use.

File

The items in the **File** menu allow you to create, open, and save model databases; open and close output databases; import and export files; save and load session objects and options; run scripts; manage macros; print viewports; and exit Abaqus/CAE. For more information, see *Using the File menu*.

Model

The items in the **Model** menu allow you to open, copy, rename, and delete the models in the current model database. For more information, see *Managing models*.

Viewport

The items in the **Viewport** menu allow you to create or manipulate viewports and viewport annotations. For more information, see *Managing viewports on the canvas*.

View

The items in the **View** menu allow you to manipulate views, customize certain aspects of the appearance of your model or plots, control display performance, switch to full screen mode, and turn off the display of the Model Tree, the Results Tree, and individual toolbars. Some of the operations available in the view manipulation menu are also available in the View Manipulation toolbar. For more information, see:

- *Working with the Model Tree and the Results Tree*
- *Managing viewports on the canvas*
- *Manipulating the view and controlling perspective*
- *Configuring graphics display options*
- *Customizing plot display*
- *The Customize toolset*
- *Customizing geometry and mesh display*

Plug-ins

The items in the **Plug-ins** menu allow you to access the plug-ins distributed with Abaqus/CAE or plug-ins that you have downloaded or created. For more information, see *The Plug-in toolset*.

Help

The items in the **Help** menu allow you to request context-sensitive help, search or browse the documentation, access the Learning Community, and obtain information about the release and licensing. For more information, see *Getting help*.

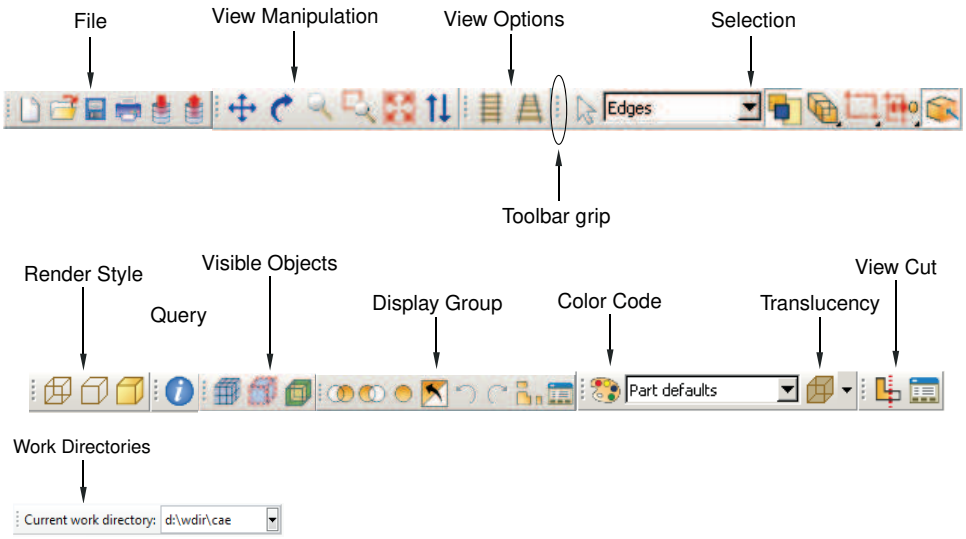
Additional information

- *Components of the main window*

Components of the toolbars

The toolbars contain convenient sets of tools for managing your files, filtering object selection, and viewing your model.
Items in a toolbar are shortcuts to functions that are also available from the main menu bar. By default, Abaqus/CAE displays all of the toolbars in a row underneath the main menu bar. Abaqus/CAE may place some toolbars in a second row depending on your display resolution and the size of the main window.

The toolbars are shown in the following figure:



You can change the location of a toolbar using the toolbar's grip, as indicated in the above figure. Clicking and dragging the grip moves the toolbar around the main window. If you release the toolbar grip while the toolbar is over one of the four available docking regions of the main window (see *Figure 1*), Abaqus/CAE “docks” the toolbar; a docked toolbar has no title bar and does not obstruct any other portion of the main window.

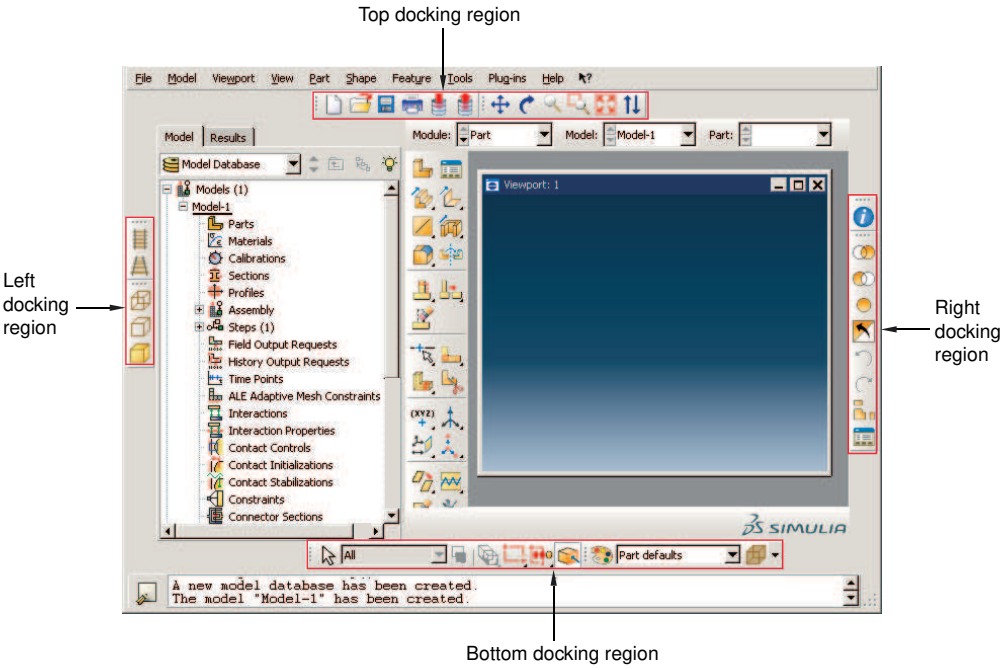


Figure 1: Available docking regions for toolbars.

If you release the toolbar grip while the toolbar is not near a docking region, Abaqus/CAE creates a floating toolbar with a title bar. A floating toolbar obstructs other items in the main window (see *Figure 2*); however, a floating toolbar can be positioned outside of the Abaqus/CAE main window.

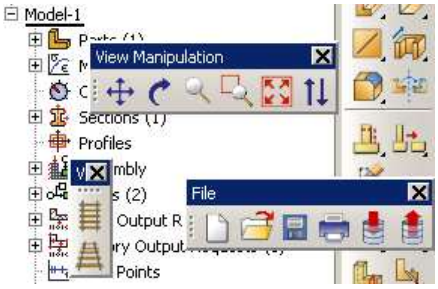


Figure 2: Floating toolbars.

Clicking mouse button 3 on a toolbar grip displays a menu that lets you specify the location and format of the toolbar:

- Select **Top** to dock the toolbar in the top docking region.
- Select **Bottom** to dock the toolbar in the bottom docking region.
- Select **Left** to dock the toolbar in the left docking region.
- Select **Right** to dock the toolbar in the right docking region.
- Select **Float** to change a docked toolbar into a floating toolbar; this option is available only for docked toolbars.
- Select **Flip** to change the orientation of a floating toolbar from horizontal to vertical, or vice versa; this option is available only for floating toolbars.

You can also hide toolbars and create custom toolbars that include shortcuts to additional functions. For more information, see *The Customize toolset*.

To obtain a short description of a tool in a toolbar, place the cursor over that tool for a moment; a small box containing a description, or "tooltip," will appear. To obtain the name of a toolbar, place the cursor over the toolbar grip for a moment.

The Abaqus/CAE toolbars contain the following functionality:

File



The File toolbar allows you to create, open, and save model databases; to open output databases; to print viewports; and to save and load session objects and options. For more information, see *Working with Abaqus/CAE Model Databases, Models, and Files*; *Printing viewports*; and *Managing session objects and session options*.

Work Directories



The Work Directories toolbar allows you to change the current working directory. The list provided in the toolbar contains the five most recently used work directories. For more information, see *Setting the work directory*.

Viewport



The Viewport toolbar allows you to create and align viewports, link viewports, and create viewport annotations. For more information, see *Managing viewports and viewport annotations from the Viewport toolbar*. The Viewport toolbar is not displayed by default.

View Manipulation



The View Manipulation toolbar allows you to specify different views of the model or plot. For example, you can pan, rotate, or zoom the model or plot using these tools. For more information, see *Manipulating the view and controlling perspective*.

View Options



The View Options toolbar allows you to specify whether or not perspective is applied to your model. For more information, see *Controlling perspective*.

Views



The Views toolbar allows you to apply a custom view to the model in the viewport. For more information, see *Custom views*. The Views toolbar is not displayed by default.

Render Style



The Render Style toolbar allows you to specify whether the wireframe, hidden line, or shaded render style will be used to display your model. In the Visualization module the Render Style toolbar also includes the filled render style  tool. For more information, see *Choosing a render style*.

Visible Objects



The Visible Objects toolbar allows you to switch between displaying the geometry of an Abaqus/CAE native part and the meshed representation of the same part, to toggle the display of seeds on and off, and to toggle the display of the reference representation on or off if the meshed representation and reference representation exist. For more information, see *Displaying a native mesh*, *What are mesh seeds?*, and *Understanding the reference representation*.

Selection



The Selection toolbar allows you to enable or disable object selection by toggling on the arrow icon. You can use the list to the right of the arrow to limit the types of objects that you can select. The **Selection** toolbar is available only when there are no active procedures running in a viewport. For more information, see *Selecting objects before choosing a procedure*.

Query



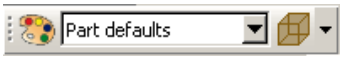
The Query toolset allows you to obtain information about the geometry and features of your model, to probe model and *X*–*Y* plots for output data, and to perform stress linearization on your results. For more information, see *The Query toolset*; *Probing the model*; and *Calculating linearized stresses*.

Display Group



The Display Group toolbar allows you to selectively plot one or more model or output database items. For example, you can create a display group that contains only the elements belonging to specified sets in your model. For more information, see *Using display groups to display subsets of your model*.

Color Code



The Color Code toolbar allows you to customize the colors of items in the viewport and change the degree of their translucency.

For color coding, you can create color mappings that assign unique colors to different elements of a display. For example, when using a part instance color mapping, each part instance in a model will appear as a different color. For more information, see *Color coding geometry and mesh elements*.

For translucency, you can click the arrow to the right of the  tool to reveal a slider, which you can drag to make the display colors more transparent or more opaque. For more information, see *Changing the translucency*.

View Cut



The View Cut toolbar allows you to toggle the display of view cuts in modules other than the Visualization module and to customize their definition and display. For more information, see *Cutting through a model*. The View Cut toolbar is displayed by default; in the Visualization module, view cut options are available in the toolbox.

Field Output



The Field Output toolbar allows you to control two aspects of field output variable display:

- You can select the field output variable that you want to display in the current viewport. Selections include the type of field output variable (**Primary**, **Deformed**, or **Symbol**), the variable name, and if available, the invariants and components for the selected primary variable.
- For changes in variable type, you can control whether Abaqus/CAE automatically synchronizes the plot state in the current viewport with the new selection of variable type. If the tool is toggled on, Abaqus/CAE synchronizes the plot state if the newly selected field output variable requires a change in plot state; if this option is toggled off, Abaqus/CAE still updates the output variable displayed in the viewport but does not change the plot state in the current viewport.

The selections in the toolbar are limited, but the tool provides access to the **Field Output** dialog box, if needed. For more information about the options in the toolbar, see *Using the field output toolbar*.

Additional information

• *Components of the main window*

The context bar

The context bar is located above the canvas and drawing area; you can use it to do the following:

Select the current module

The **Module** list on the context bar allows you to move between modules. (For more information, see *What is a module?*.) *Figure 1* shows the context bar. To move to a different module, you can choose from the list (the arrow on the right) or click the up and down arrows (on the left) to move to the previous or next module.



Figure 1: The context bar in the Part module.

Note:

Abaqus/Viewer contains only the Visualization module.

Select module-specific items

As you move between modules, Abaqus/CAE displays additional items on the context bar that help you select the context of your current operations. For example, when you are in the Part module or Mesh module, Abaqus/CAE displays the **Part** list in the context bar. The **Part** list contains every part in your model; you can use it to retrieve a particular part. These lists also include the up and down navigation arrows that allow you to move to the previous or next item in the list.

The context bar also allows you to move between models in the model database or to change the output database associated with the current viewport. The additional items in the context bar are a function of the module in which you are working.

The items displayed in the context bar always refer to the current viewport, which is indicated by a dark gray title bar. For example, if you have different parts displayed in different viewports, the context bar indicates the name of the part displayed in the current viewport.

Additional information

- *What is a module?*
- *What is an Abaqus/CAE model?*

Components of the viewport

Figure 1 shows the components of the viewport in the Visualization module.

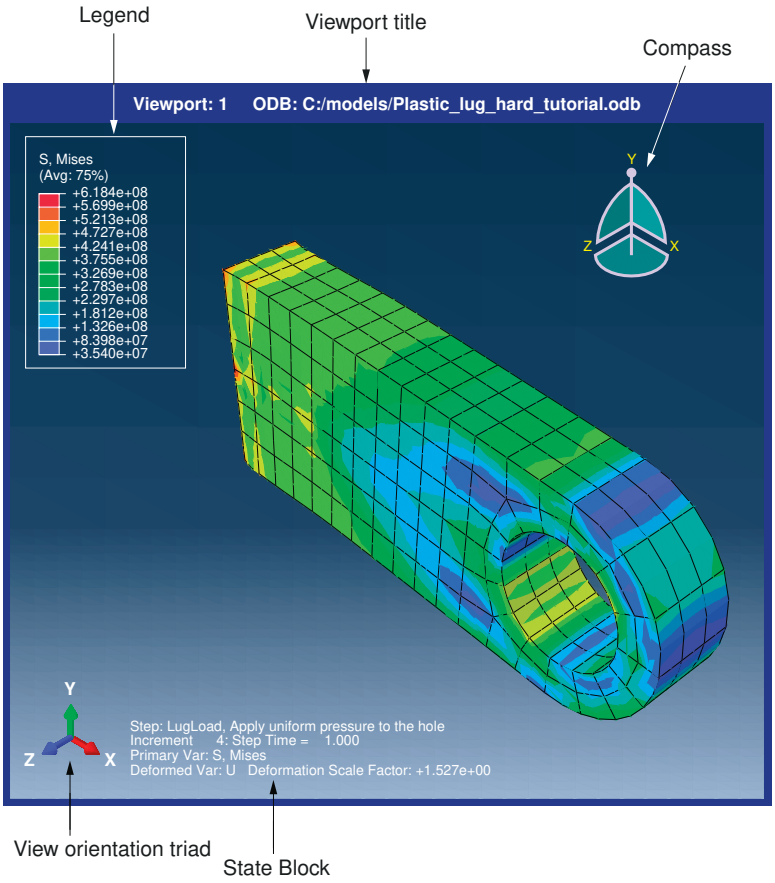


Figure 1: Components of the viewport.

The viewport title and the border around the viewport are called the viewport decorations. The legend, state block, title block, view orientation triad, and 3D compass are called the viewport annotations. The view orientation triad and 3D compass indicate the orientation of the model currently being displayed. You can change the view of the model by clicking and dragging on the 3D compass; the three perpendicular axes on the view orientation triad rotate with the compass to indicate the current view orientation. For more information, see *The 3D compass*, and *Customizing the view triad*. The legend, state block, and title block identify results you display using the Visualization module. For more information, see *Customizing viewport annotations*.

Additional information

- *Managing viewports on the canvas*
- *Customizing viewport annotations*

What is a module?

Abaqus/CAE is divided into functional units called modules. Each module contains only those tools that are relevant to a specific portion of the modeling task. For example, the Mesh module contains only the tools needed to create finite element meshes, while the Job module contains only the tools used to create, edit, submit, and monitor analysis jobs. Abaqus/Viewer is a subset of Abaqus/CAE that contains only the Visualization module.

You can select a module from the **Module** list in the context bar. Alternatively, you can select a module by switching to the context of a selected object in the Model Tree; for more information, see *The Model Tree*. The order of the modules in the menu and in the Model Tree corresponds to the logical sequence you follow to create a model. In many circumstances you must follow this natural progression to complete a modeling task; for example, you must create parts before you create an assembly. Although the order of the modules follows a logical sequence, Abaqus/CAE allows you to select any module at any time, regardless of the state of your model.

The following list of the modules available within Abaqus/CAE briefly describes the modeling tasks you can perform in each module. The order of the modules in the list corresponds to the order of the modules in the context bar's **Module** list and in the Model Tree:

Part

Create individual parts by sketching or importing their geometry. For more information, see *The Part module*.

Property

Create section and material definitions and assign them to regions of parts. For more information, see *The Property module*.

Assembly

Create and assemble part instances. For more information, see *The Assembly module*.

Step

Create and define the analysis steps and associated output requests. For more information, see *The Step module*.

Interaction

Specify the interactions, such as contact, between regions of a model. For more information, see *The Interaction module*.

Load

Specify loads, boundary conditions, and fields. For more information, see *The Load module*.

Mesh

Create a finite element mesh. For more information, see *The Mesh module*.

Optimization

Create and configure an optimization task. For more information, see *The Optimization module*.

Job

Submit a job for analysis and monitor its progress. For more information, see *The Job module*.

Visualization

View analysis results and selected model data. For more information, see *Viewing results*.

Sketch

Create two-dimensional sketches. For more information, see *The Sketch module*.

Modules can be classified by the objects that are displayed in the viewport. Parts are displayed when you are in the Part and Property modules; the assembly is displayed when you are in the Assembly, Step, Interaction, Load, Mesh, and Job modules; and output database results are displayed when you are in the Visualization module.

The contents of the main window change as you move between modules. Selecting a module from the **Module** list on the context bar or by switching to the context of a selected object in the Model Tree causes the context bar, module toolbox, and menu bar to change to reflect the functionality of the current module.

When you move between modules, Abaqus/CAE associates the current viewport with the module you select. You can have multiple viewports, and different viewports can be associated with different modules. As you select a viewport and make it current, the module associated with the viewport becomes the current module. For more information on moving between viewports, see *Selecting viewports*.

Additional information

- *The context bar*
- *What is a viewport?*

What is a toolset?

A toolset is a functional unit that allows you to perform a specific modeling task.
When you enter most modules, a **Tools** menu appears in the main menu bar containing all of the toolsets relevant to that module.

In most cases the objects that you create with a toolset in one module are useful in other modules. For example, you can use the Set toolset to create sets in the Assembly module and then apply boundary conditions to those sets in the Load module. Most of the toolsets include manager menus and manager dialog boxes that allow you to edit, copy, rename, and delete the objects you create with the toolset.

The following toolsets are available in Abaqus/CAE:

- The Amplitude toolset allows you to define arbitrary time or frequency variations of load, displacement, and other prescribed variables. For more information, see *The Amplitude toolset*.
- The Analytical Field toolset allows you to create analytical fields that you can use to define spatially varying parameters for selected interactions and prescribed conditions. For more information, see *The Analytical Field toolset*.
- The Attachment toolset allows you to create attachment points and lines that you can use to define point-based and discrete fasteners, connector points for a connector, and regions for a coupling definition, point mass, load, or boundary condition. For more information, see *The Attachment toolset*.
- The CAD Connection toolset allows you to create a connection that you can use for associative import of parts into Abaqus/CAE from CATIA and third-party CAD systems. For more information, see *The CAD Connection toolset*.
- The Color Code toolset allows you to customize the edge and fill color of individual elements. For more information, see *Color coding geometry and mesh elements*.
- The Coordinate System toolset allows you to create local coordinate systems for use in postprocessing. For more information, see *Creating coordinate systems during postprocessing*.
- The Customize toolset allows you to control the appearance of Abaqus/CAE toolbars, to create customized toolbars, and to specify keyboard shortcuts for many Abaqus/CAE features. For more information, see *The Customize toolset*.
- The Datum toolset allows you to create datum points, axes, planes, and coordinate systems for a variety of modeling tasks. For more information, see *The Datum toolset*.
- The Discrete Field toolset allows you to create a spatially varying field where values are associated with nodes or elements. For more information, see *The Discrete Field toolset*.
- The Display Group toolset allows you to selectively plot one or more model or output database items. For more information, see *Using display groups to display subsets of your model*.
- The Edit Mesh toolset allows you to modify a mesh to improve mesh quality. For more information, see *The Edit Mesh toolset*.
- The Feature Manipulation toolset allows you to modify and manage the existing features in your model. For more information, see *The Feature Manipulation toolset*.
- The Field Output toolset allows you to perform operations on the field output available in an output database. For more information, see *Creating and saving new field output*.
- The Filter toolset allows you to remove extraneous output data—noise—during the analysis of a model without a loss of resolution in the desired data range. For more information, see *The Filter toolset*.
- The Free Body toolset allows you to create and customize free body cuts in the Visualization module of Abaqus/CAE. For more information, see *The Free Body toolset*.
- The Geometry Edit toolset allows you to repair invalid and imprecise imported parts. For more information, see *The Geometry Edit toolset*.
- The Partition toolset allows you to divide a part or assembly into regions. For more information, see *The Partition toolset*.

- The Path toolset allows you to specify a path through your model along which you can obtain and view *X*–*Y* data. For more information, see *Viewing results along a path*.
- The Query toolset allows you to obtain general information about your model and to probe model and *X*–*Y* plots for output data. For more information, see *The Query toolset*.
- The Reference Point toolset allows you to create reference points associated with a part or assembly. For more information, see *The Reference Point toolset*.
- The Set toolset and the Surface toolset allow you to define sets and surfaces from regions of a model. For more information, see *The Set and Surface toolsets*.
- The Stream toolset allows you to display streamlines to investigate velocity or vorticity in a fluid flow analysis. For more information, see *The Stream toolset*.
- The Virtual Topology toolset allows you to ignore details, such as very small faces and edges, when you are meshing a part or a part instance. For more information, see *The Virtual Topology toolset*.
- The XY Data toolset allows you to create and operate on *X*–*Y* data objects. For more information, see *X–Y plotting*.

Using the mouse with Abaqus/CAE

Many of the procedures in the Abaqus/CAE documentation involve using one or more of the three mouse buttons. The following list explains the importance of each mouse button when interacting with Abaqus/CAE:

Mouse button 1

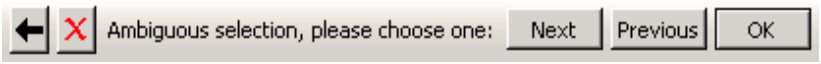
You use mouse button 1 to select objects in the viewport, to expand pull-down menus, and to select items from menus. The instructions “click,”“select,” and “drag” in the documentation refer to mouse button 1.

Mouse button 2

Clicking mouse button 2 in the viewport signifies that you have finished the current task. For example:

- Selecting entities from the model: when you create a node set, you select the nodes to include in the set. Clicking mouse button 2 indicates that your selection is complete and you are ready to create the set.
- Using a tool: click mouse button 2 to indicate that you have finished with a view manipulation tool.

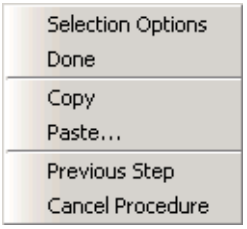
In addition, clicking mouse button 2 in the viewport is equivalent to clicking the highlighted button in the prompt area. For example, if you tried to select nodes from your model and Abaqus/CAE displayed the following prompt, clicking mouse button 2 would have the same effect as clicking **OK**:



If your mouse has a wheel as mouse button 2, you can scroll the wheel vertically to manipulate your view of the model or plot in the viewport. Scroll downward to magnify your view of the contents of the viewport, or scroll upward to reduce your view of the contents of the viewport.

Mouse button 3

You press and hold mouse button 3 to access a popup menu that contains shortcuts to functions related to the current procedure. For example, when you press mouse button 3 in a viewport while creating a geometry set, Abaqus/CAE displays the following menu:



If you use mouse button 3 in a viewport, most of the items in the popup menu duplicate the buttons in the prompt area. The mouse button 3 shortcut is also available for selections from the Model Tree and Results Tree, as described in *Using popup menus in the Model Tree and the Results Tree*.

Getting help

The Abaqus/CAE HTML documentation is available through the **Help** menu on the main menu bar. This section provides a brief description of the HTML online documentation and explains how to use the **Help** menu to find information.

The features described in this section apply only to the HTML documentation, not the PDF-format guides.



Note:

- On Windows platforms, the help system uses your default web browser to display the online documentation.
- On Linux platforms, the help system searches the system path for Firefox. If the help system cannot find Firefox, an error is displayed.

  The **browser_type** and **browser_path** variables can be set in the environment file to modify this behavior. For more information, see *System customization parameters*.

In this section:

- *Displaying context-sensitive help*
- *Browsing and searching the HTML guides*
- *Finding special sections of the online documentation*
- *Finding information about keywords*
- *Accessing the Learning Community*
- *Obtaining information about the release and licensing*

Displaying context-sensitive help

You can display detailed HTML help on any icon, menu, or dialog box that you use in Abaqus/CAE.

You can use the help tool  on the main menu bar to display detailed HTML help on any icon, menu, or dialog box that you use in Abaqus/CAE. When you click the help tool and then click an item in the Abaqus/CAE window, a help window appears containing the section from the online documentation that is relevant to that item.

Display help on an item in the main window or in a dialog box

1. Click the help tool  on the main menu bar.

    **Tip:** You can also select **Help->On Context** from the main menu bar.

   The cursor changes to a question mark.

2. Position the cursor over the item about which you need help, and click mouse button 1.

   A help window appears. The window contains the appropriate online documentation and links to associated topics.

Display help using the [F1] key

Alternatively, you can use the **[F1]** key to display help on a particular item. In most cases you can gain access to context-sensitive help by using the **Help** menu, the help tool icon, or the **[F1]** key. However, you must use **[F1]** if you are seeking information about menu items or dialog boxes that do not allow access to the help tool.

1. Click the feature in the Abaqus/CAE window that you want help with. If the feature is part of a menu, do not release the mouse button.
2. Press **[F1]**.

   A help window appears. The window contains the appropriate online documentation and links to associated topics. If you selected a menu item without releasing the mouse button, that menu disappears.

 **Note:**

Abaqus/CAE also provides brief “tooltips” that describe the function of tools in toolboxes and in the toolbars. To see a “tooltip,” position the cursor over a tool and leave it stationary for a short time.

Browsing and searching the HTML guides

You can open and browse and search the entire HTML collection using the **Help** menu.

1. From the main menu bar, select **Help->Search & Browse Guides**.

   The documentation displays in your web browser open to a topic that contains an overview of the guides in the documentation.

2. To view a particular guide, click **Abaqus** in the table of contents and click the title of interest.

   The guide that you selected opens in your browser window.

3. Navigate through the guide's contents using any of the following techniques:

Browsing

Expand and collapse the table of contents to vary the level of detail displayed. Click the topic of interest. You can also use the web browser functions to return to recently viewed pages.

Searching

Use the search panel located in the navigation frame to search for specific words or phrases.

Using hyperlinks

Use hyperlinks to move from one part of a guide to another or from one guide to another guide.

Finding special sections of the online documentation

The following **Help** menu items allow you to display sections of the HTML documentation that you may find useful:

On Module

Select **Help->On Module** to display the *Abaqus/CAE User's Guide* opened to the beginning of the chapter that describes the current module. If you have not yet entered a module, the guide will be opened to a description of the module concept. In either case, you are then free to read additional information as needed and to conduct text searches through the entire guide.

On Help

Select **Help->On Help** to display the *Abaqus/CAE User's Guide* opened to the section that describes how to use the help system. You are also free to read additional information as needed and to conduct text searches through the entire guide.

Getting Started

Select **Help->Getting Started** to display a section that provides basic information on how to work in the Abaqus/CAE window. This section also contains links to helpful tutorials in the *Getting Started with Abaqus/CAE* guide.

Release Notes

Select **Help->Release Notes** to display the *Abaqus Release Notes*. Release notes detail new features of the software and provide a list of updates and enhancements.

Finding information about keywords

The keyword browser is a scrollable table that contains the following information:

- The purpose of each keyword.
- The Abaqus/CAE module or toolset that contains the functionality associated with each keyword.

To view the keyword browser, select **Help->Keyword Browser** from the main menu bar. For example, you could use the keyword browser to verify that the **ELASTIC* option allows you to specify elastic material properties and that the Property module is the Abaqus/CAE module associated with this keyword.

The keyword browser also contains hyperlinks to relevant sections in the online documentation. You can click a particular keyword in the table to display detailed information concerning the function of that keyword. You can also click the name of a module or toolset in the table to view related documentation in the *Abaqus/CAE User's Guide*.

1. From the main menu bar, select **Help->Keyword Browser**.

   The *Abaqus/CAE User's Guide* is opened to a table of Abaqus keywords and their associated modules.
2. In the **Keyword** column, click the keyword of interest to view online documentation describing that keyword.
3. In the **Module** column, click the module or toolset name of interest to view online documentation concerning that module or toolset.

Accessing the Learning Community

You can access the Learning Community at *https://www.3ds.com/products-services/simulia/simulia-academic-program/learning-community/*. The Learning Community contains online tutorials and technical content. The community also hosts a question-and-answer area that enables the global community of users to share their expertise and learn how to leverage the latest features and enhancements available in the SIMULIA portfolio.

Obtaining information about the release and licensing

The following **Help** menu items allow you to obtain additional information:

About Abaqus

Select **Help->About Abaqus** to determine which release of Abaqus/CAE you are currently using. Abaqus also provides the location of release information for open source software used by Abaqus/CAE; for example, Python.

About Licensing

Select **Help->About Licensing** to determine product license information. Abaqus displays your site identification and the name of your license server along with your license number and the total number of licenses available from your site.

Understanding Abaqus/CAE windows, dialog boxes, and toolboxes

This chapter explains how to interact with the various windows, dialog boxes, and toolboxes that appear throughout the Abaqus/CAE application.

In this section:

- *Using the prompt area during procedures*
- *Interacting with dialog boxes*
- *Understanding and using toolboxes and toolbars*
- *Managing objects*
- *Working with the Model Tree and the Results Tree*
- *Understanding Abaqus/CAE GUI settings*

Using the prompt area during procedures

This section explains how to make use of the procedural steps that Abaqus/CAE displays in the prompt area.

In this section:

- *What is a procedure?*
- *Following instructions and entering data in the prompt area*
- *Using mouse shortcuts with procedures*

What is a procedure?

Many tasks within Abaqus/CAE are broken into step-by-step procedures. For example, creating an arc in the Sketcher is a three-step procedure:

1. Pick the center point for the arc.
2. Pick the start point.
3. Pick the end point.

Abaqus/CAE displays each step of a procedure in the prompt area near the bottom of the main window so that you do not need to remember all the steps and their order.

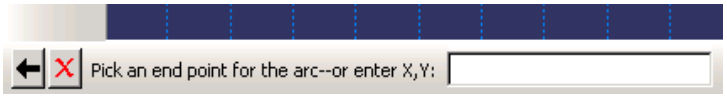
Additional information

- *Using the prompt area during procedures*

Following instructions and entering data in the prompt area

To use a procedure, simply follow the directions that appear in the prompt area near the bottom of the main window.

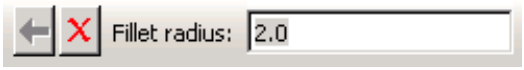
For example, follow the directions as shown here:



The button marked X in the above figure is the Cancel button; click this button to cancel the entire procedure at any time. The arrow to the left of the Cancel button is the Previous button; click it to end the current step of the procedure and return to the previous one. (The Previous button appears dimmed during the first step of any procedure.) If you prefer, you can place the cursor over the canvas and press mouse button 3; then select **Previous Step** or **Cancel Procedure** from the menu that appears.

A **Stop** button appears in the prompt area during certain time-consuming operations, such as part healing or meshing or the extraction of *X–Y* data from history for large models. You can click **Stop** to interrupt and cancel the operation.

Many procedures require textual or numeric data; for example, when creating a fillet using the Sketch module, you must first specify the fillet radius. When textual or numeric data are required, Abaqus/CAE displays a text field in the prompt area for you to fill in; usually the text box will already contain a default value, as shown here:



Position your cursor over the viewport, and enter data into the text field as follows:

- To accept the default value, press either **[Enter]** or mouse button 2.
- To replace the default value, simply begin typing; you need not click the text field before typing. The default value disappears as soon as you begin to type.
- To change a portion of the default value, first click the text field; then use the **[Delete]** key and the other keys on your keyboard to change the value.
- To commit any changes, press **[Enter]** or mouse button 2.
- You can also enter an expression in a text field in the prompt area. For more information, see *Entering expressions*.

Some procedures require you to choose from a number of options. For example, the Datum toolset may ask you to choose a principal axis. Such options are represented by buttons in the prompt area, as shown here:



Click the appropriate button to select the desired option.

In some procedures a default option is indicated by a border around the corresponding button; in the above example the border is drawn around the **X-Axis** button. To select the default option, click mouse button 2.

Additional information

• *Using the prompt area during procedures*

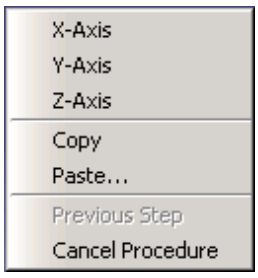
Using mouse shortcuts with procedures

Mouse shortcuts are available for many of the actions that take place in the prompt area. To use the shortcuts, first make sure that the cursor is in the current viewport.

- To commit the contents of any text field that appears in the prompt area, click mouse button 2.
- To accept any default option depicted by a highlighted button in the prompt area, click mouse button 2.
- To reveal a menu containing options identical to those in the prompt area, click mouse button 3. For example, given the following prompt:



Clicking mouse button 3 will reveal the following menu:



Items above the horizontal line correspond to the option buttons on the right side of the prompt area, while items below the line correspond to the Previous and Cancel buttons.

Additional information

• *Using the prompt area during procedures*

Interacting with dialog boxes

This section explains how to use the various dialog box components that appear within Abaqus/CAE.

In this section:

- *Using basic dialog box components*
- *Entering expressions*
- *Using dimmed dialog box and toolbox components*
- *Disabling warning dialog boxes*
- *Understanding the OK, Apply, Defaults, Continue, Cancel, and Dismiss buttons*
- *Using dialog boxes separated by tabs*
- *Entering tabular data*
- *Customizing fonts*
- *Customizing colors*
- *Using file selection dialog boxes*
- *Selecting multiple items from lists and tables*
- *Using keyboard shortcuts*

Using basic dialog box components

Dialog box components include text fields, numeric fields, combo boxes, radio buttons, check boxes, scroll bars, and sliders.

The following types of components are present in dialog boxes throughout Abaqus/CAE:

Text fields

Text fields are areas in dialog boxes in which you can enter information. For example, when you save a display group, you must enter its name in the text field shown below:



If you are entering a floating point number, most text fields allow you to enter an expression; for example, `cos(2.5/(4.9*pi))`. The expression can be any valid Python expression. For more information, see *Entering expressions*.

Text fields are available whenever you need to name an object (such as a part, material, set, path, or *X*–*Y* data) or provide a description for an object (such as a material or step). In general, you should avoid using an asterisk (*) in an object name or description.

Object names must adhere to the following rules:

- Part, model, instance, set, surface, feature, and job names can have up to 80 characters; other object names can have up to 38 characters. Instance names of models that have been instantiated as model instances in another model still have a 38-character limit. For imported sets/surfaces, parts, and model instances, the names are generated internally in Abaqus/CAE by combining part/instance/set names. You must ensure that the combined length will not exceed 80 characters; otherwise, the data check analysis will fail.
- The name can include spaces and most punctuation marks and special characters.
- The name must not begin with a number.
- The name must not begin or end with an underscore or a space.
- The name must not contain a period or double quotes.
- The name must not contain a backslash.
- The name cannot be `Assembly`, which is reserved for internal use by Abaqus/CAE.

Additional restrictions apply to model names, part names, and job names.

- When you name a model or a job, the name can begin with a number.
- When you name a model, you cannot use the following characters:

  ```
  $&*~!()[]{}|;'`",.?/\><
  ```

- When you name a part, the name should not be the same as the model name.
- When you name a job, you cannot use the following characters:

  ```
  <space>$&*~!()[]{}|:;'`",.?/\><
  ```

  In addition, a job name cannot begin with a dash −.

The material evaluation procedure (*Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*) generates jobs with the same names as the materials; therefore, these material names must adhere to the same

rules as job names. In general, when you are specifying a name that will be used external to Abaqus/CAE, such as a file name, you should avoid any character that may have a reserved meaning on your platform.



Note:

Abaqus/CAE retains the case of any text you enter in a text field. For example, if you create a material called `Steel Alloy` in the **Edit Material** dialog box in the Property module, the material will appear as `Steel Alloy` in the graphical user interface (material manager, section editor, Model Tree, etc.). In the graphical user interface, object names are case insensitive. For example, you cannot create a second material called `steel alloy`. Conversely, Python (which is used in the command line interface) is case sensitive, but you should not rely on this behavior to distinguish between objects.

Numeric fields

Numeric fields are specialized text fields for integer input values. They have two opposing arrows directly to the right of the text area. You can enter a numeric value into the text field, or you can use the arrows to cycle up and down through a list of fixed values.



Unlike other text fields, numeric fields do not accept text or special characters.

Numeric fields often have upper and lower limits. If the value you enter exceeds the limits, Abaqus/CAE changes the entry to the closest acceptable value when you move to another field or try to apply the value.

Combo boxes

Combo boxes are fields having an arrow directly to the right of the field. If you click this arrow, a list of the possible choices that you can enter in the field appears. For example, if you click the arrow to the right of the **Module** field in the context bar, a list of all the Abaqus/CAE modules appears, and you can select the module of your choice from the list.



Radio buttons

Radio buttons present a mutually exclusive choice. When an option is controlled by radio buttons, you can choose only one of the buttons at a time.



Check boxes

You can toggle a check box to turn a particular option off or on.

For example, the visibility of the triad in the current viewport depends on the status of the **Show triad** check box. If the box is toggled on, as shown below, the triad appears in the viewport.



If the box is toggled off, as shown below, the triad does not appear in the viewport.

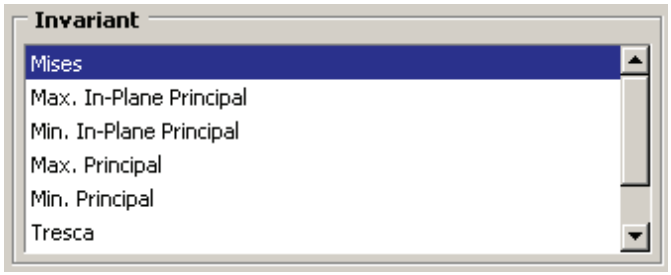


In some cases the option controlled by a check box can apply to more than one object. For example, a single **Show line** check box in the **XY Curve Options** dialog box individually controls the display of all *X–Y* curve lines in an *X–Y* plot. If you have toggled **Show line** on for some curves and off for others, that check box appears gray with a darker gray check mark, as shown below.



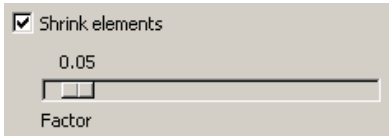
Scroll bars

Scroll bars appear in lists whose contents are too big to display; they allow you to scroll through the visible contents of the list as well as any contents that are hidden. Scrolling is often necessary when numerous items must be listed, as shown below.



Sliders

Sliders allow you to set the value of an option that has a continuous range of possible values. An example of a slider is shown in the following figure:



Additional information

• *Interacting with dialog boxes*

Entering expressions

If a field in a dialog box or in the prompt area is expecting a floating point number or a complex number, you can enter an arithmetic expression, as shown in *Figure 1*.

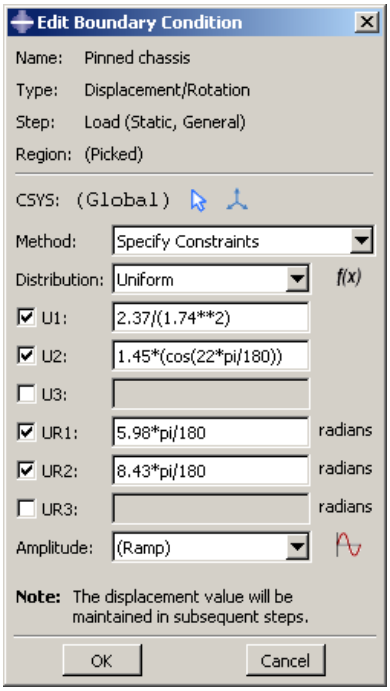


Figure 1: An expression in a text field.

The expression is evaluated by the Python interpreter that is built into Abaqus/CAE. The arithmetic expression is replaced by its value; if you reopen a dialog that contained expressions, only the values are available. Variables like `pi` and functions like `sin()` are available because Abaqus/CAE imports the Python math module when you start a session. As a result, you can enter any expression that can be evaluated by Python's built-in functions or by the Python math module. For more information, see the documentation for built-in functions and the math module accessible from the official Python home page (*http://www.python.org*).

To make sure that your expression is evaluated as expected, you should be aware of the following:

- If you enter numbers as integers, Python will perform integer division and round down any remainder. For example, Python will interpret `3/2` as `1` and `1/2` as `0`. In contrast, Python interprets `3./2` as `1.5` and `1/2.` as `0.5`.
- Python interprets numbers with leading zeros as octal numbers (for example, `0123` is interpreted as `83.0`). However, Abaqus/CAE will ignore leading zeros in numbers in text fields before Python interprets them; such numbers are evaluated as decimals.
- Python interprets `e` as the base of the natural logarithm; that is, `e` equates to `2.71828182846` and `e+2` equates to `4.71828182846`.
- If the "e" character is preceded by a number, Python interprets it as an exponent, not a natural logarithm. For example, Python interprets `2e+2` as 2×10^2 and equates it to `200`.
- Python interprets `2e+` as 2×10^0 and equates it to `2`. Similarly, Python interprets `2e++11` as 2×10^0 `+ 11` and equates it to `13`.

If you are unsure how Python will interpret your expression, you can enter the expression on the command line; Abaqus/CAE will print the resulting interpreted value in the message area. To access the command line interface, click 

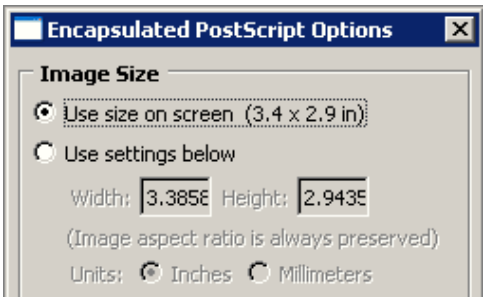
in the bottom left corner of the main window. For more information, see *Components of the main window*.

You can also test how Abaqus/CAE interprets an expression by entering `abaqus python` at an operating system prompt and entering the expression at the Python prompt that appears. The prompt line and some dialog boxes do not allow you to enter an expression. As an alternative, you can enter the expression on the command line or at the Python prompt and paste the resulting value in the prompt line or dialog box.

Using dimmed dialog box and toolbox components

Some objects in dialog boxes and toolboxes are available only under certain circumstances. When an object is unavailable, it appears dimmed in the dialog box.

Items are usually dimmed as a result of some other setting in the dialog box. For example, if **Use settings below** is not selected, the image size options below it are not available and appear dimmed, as shown below.



Context-sensitive help is available even for dimmed options, although tooltips are not.

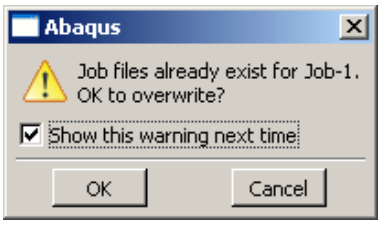
Additional information

• *Interacting with dialog boxes*

Disabling warning dialog boxes

Some dialog boxes can be disabled so that they will not appear again during the current Abaqus/CAE session.

For example, if you submit a job for analysis and job files with the same name already exist, Abaqus/CAE displays a dialog box asking if it is OK to overwrite the job files, as shown below.



If you toggle off **Show this warning next time**, the dialog box will be disabled for the remainder of the current Abaqus/CAE session.

Additional information

• *Interacting with dialog boxes*

Understanding the OK, Apply, Defaults, Continue, Cancel, and Dismiss buttons

When you are finished working with a dialog box, you can specify how to proceed by using different action buttons. For example, if you enter data in a dialog box, you can save the data and apply them by clicking **OK**. If the dialog box is part of an intermediate step of a procedure, you can click **Continue** to move on to the next step.

The following action buttons can appear in a dialog box:

OK

Click **OK** to commit the current contents of a dialog box and to close the dialog box.

Apply

When you click **Apply**, any changes you have made in the dialog box take effect, but the dialog box remains displayed. This button is useful if you make changes in a dialog box and would like to see the effects of these changes before closing the dialog box.

Defaults

If you want to revert back to the predefined default values after entering data or specifying preferences in a dialog box, you can click **Defaults**. This button affects only the information entered in the dialog box. It does not apply your changes or close the dialog box; therefore, to see the effect of reverting to the default values, you must click **Apply** or **OK**.

Cancel

Click **Cancel** to close a dialog box without applying any of the changes that you made. If the dialog box appears in the middle of a procedure, clicking **Cancel** usually also cancels the procedure. In some cases clicking **Cancel** returns you to the previous step in the procedure.

Continue

Dialog boxes that appear in the middle of a procedure contain **Continue** buttons. When you click **Continue**, you indicate that you have finished entering data in the current dialog box and would like to move on to the next step of the procedure. **Continue** causes the dialog box to be closed and all data in it to be saved unless you click **Cancel** at some point later in the procedure.

Dismiss

Dismiss buttons appear in dialog boxes that contain data that you cannot modify. For example, some managers contain lists of objects that exist but no fields in which you can enter data or specify preferences. **Dismiss** buttons also appear in message dialog boxes. When you click **Dismiss**, the dialog box closes.

To close a toolbox or a dialog box that does not have a **Cancel** or **Dismiss** button, click the close button in the upper right corner of the toolbox or dialog box. Alternatively, you can close an active toolbox or dialog box by pressing **[Esc]**.



Note:

On Linux platforms, depending on your settings, **[Esc]** may be the only way to close a toolbox or dialog box. For more information, see *Linux settings that affect Abaqus/CAE and Abaqus/Viewer*.

Additional information

• *Interacting with dialog boxes*

Using dialog boxes separated by tabs

For the sake of organization and convenience, some dialog boxes are separated by tabs. Only one dialog box is visible at a time. To view a particular dialog box, click its labeled tab.

For example, *Figure 1* displays the **Common Plot Options** dialog boxes.

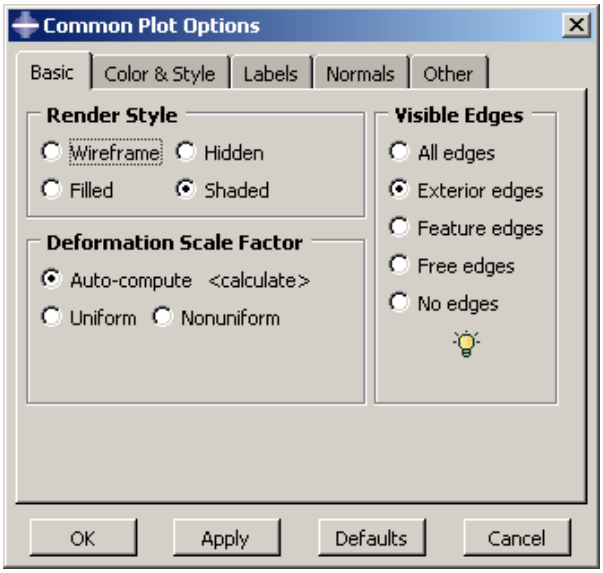


Figure 1: Dialog boxes separated by tabs.

If you click the **Color & Style** tab, the dialog box containing the color and edge attributes options comes forward, obscuring the other four dialog boxes, as shown in *Figure 2*.

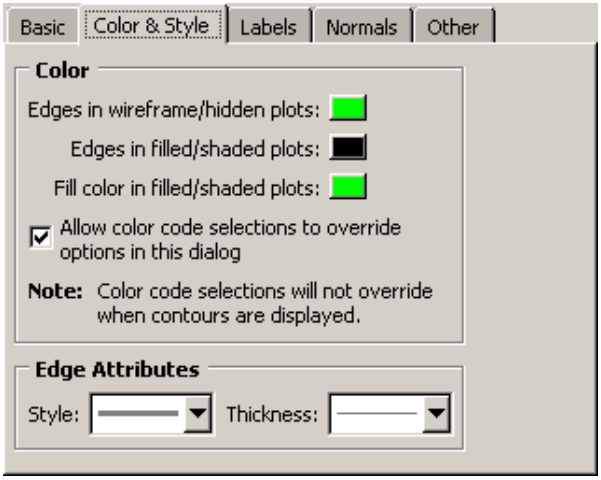


Figure 2: Using tabs to display particular dialog boxes.

In addition, separated dialog boxes can exist within a single dialog box. In this case the tabs of the separated dialog boxes are aligned vertically but work the same way as tabs aligned horizontally. In *Figure 3* the **Other** dialog box contains two dialog boxes separated by tabs: **Scaling** and **Translucency**.

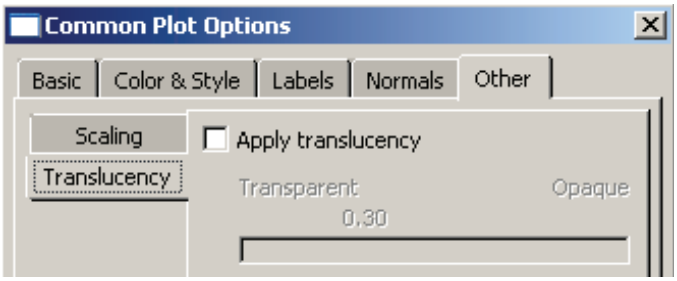


Figure 3: Dialog box containing additional dialog boxes.

The action buttons in a dialog box apply to the whole set of dialog boxes, not just the one you are currently viewing. If you click **Cancel**, all of the unapplied changes you have made in the set of dialog boxes are canceled, not just those in the current dialog box. Likewise, clicking **OK** saves all changes that you have made in any of the dialog boxes.

Additional information

• *Interacting with dialog boxes*

Entering tabular data

Some operations require the entry of tabular data. For example, the XY Data toolset can produce plots of data that you enter in the dialog box shown in *Figure 1*.

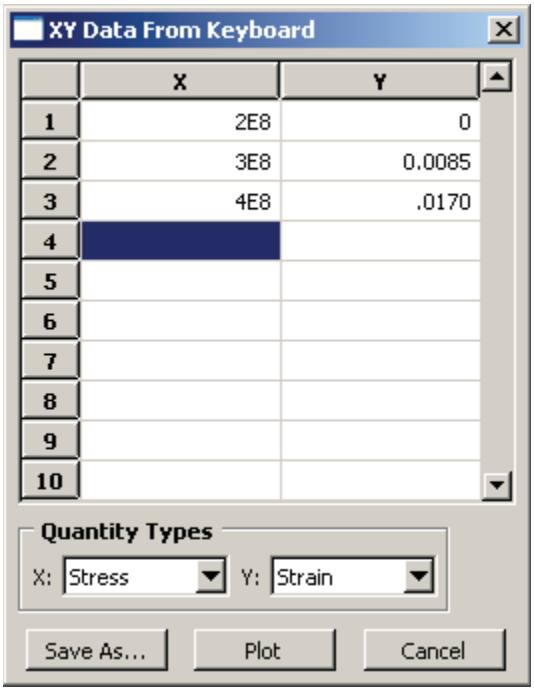


Figure 1: *X–Y* data table.

Data tables are composed of input boxes, or cells, organized into rows and columns. You can type data into a table using the keyboard, or you can read data in from a file.

The following list describes techniques for entering and modifying tabular data:

Entering data

Click any cell, and type the required data. You can press **[Enter]** to commit the data in a particular cell.

Abaqus/CAE does not allow you to enter character data in tables requiring numeric data; the program beeps if you attempt to enter character data in a numeric field. (The letter E that denotes scientific notation, as in 12.E6, is an exception to this rule.)

Adding new rows

Use the menu that appears when you click mouse button 3 to add a new row before or after an existing row. Click mouse button 3 while holding the cursor over the row of interest; then select the item of your choice from the menu that appears:

- Select **Insert Row Before** to add a blank row above the current row.
- Select **Insert Row After** to add a blank row below the current row.

Alternatively, you can add a blank row to the end of the table by clicking the cell in the last row and in the last column of the table and then pressing **[Enter]**.

Reading data from a file

You can enter data by reading it in from an ASCII file. Data fields within the file can be delimited by any combination of spaces, tabs, or commas; each space, tab, or comma is considered a single field delimiter. To enter data from a file, click mouse button 3 while holding the cursor over the target cell; then select **Read From File** from the menu that appears. The **Read Data from ASCII File** dialog box appears. In this dialog box, specify the following:

- In the **File** text field, enter the name of the file to read.
- Specify the row number and column number of the target cell in the **Start reading values into table row** and **Start reading values into table column** fields, respectively. (By default, Abaqus sets these fields to the cell your cursor was over when you clicked mouse button 3.)

Click **OK**. Abaqus reads data values from the file into the table according to your specifications.

Moving from cell to cell

Use the **[Enter]** key to move from left to right between the cells in a row. When you have reached the end of the row, press **[Enter]** to move the cursor to the first cell in the following row.

In addition, you can use a combination of the **[Tab]** key and the up and down arrow keys to move from cell to cell. Use **[Tab]** to move to the right and **[Shift][Tab]** to move to the left; use the up and down arrows to move up and down. You can also simply click the cell of interest.

Changing data

If a cell already contains data, clicking the cell highlights the data; as soon as you begin typing, the highlighted contents of the cell disappear and are replaced by whatever you type. You can also use the **[Backspace]** or **[Delete]** keys to delete highlighted data in a cell.

After clicking the cell once, you can click a second time to remove the highlighting and position the cursor within the cell. Use the **[Backspace]** key and the other keys on your keyboard to modify the data.

Cutting, copying, and pasting data

Use the menu that appears when you click mouse button 3 to cut, copy, and paste data from one location in a table to another. You can cut or copy data in single cells, in rows or parts of rows, in columns or parts of columns, and in series of consecutive rows or columns.

First, drag the mouse over the cells containing the data that you want to cut or copy. All of the selected cells will become highlighted except the cell that you selected first. This cell becomes highlighted when you move the cursor outside the data table window or if you click mouse button 3.

Once you have selected the cells of interest, click mouse button 3 while holding the cursor over the selection; then select either **Cut** or **Copy** from the menu that appears. To paste the data, select the target cell, click mouse button 3, and select **Paste** from the menu that appears.

Sorting data

Some data tables offer a sorting feature. (To determine if sorting is available for a particular table, hold the cursor over the table; then click mouse button 3. If it is available, **Sort** is listed in the menu that appears.)

To sort table data, click mouse button 3 while holding the cursor over the table; then click **Sort**. The **Sort Table** dialog box appears. In this dialog box, choose the following:

- In the **Sort by** text field, choose the column by which to sort.
- Choose **Ascending** or **Descending** sort order.

Click **OK** or **Apply**. Abaqus sorts all rows according to data values in the specified column.

Expanding and contracting columns

You can change the size of the columns in some tables. To expand or contract a column, move the cursor to the line that divides the headings of the columns you want to resize; a resize cursor will appear. Drag this cursor to the left or right to resize the two columns on either side of the dividing line.

You can also resize the last column in some tables by horizontally enlarging the dialog box that contains the table.

Viewing data that extend beyond the edge of the dialog box

Use the horizontal and vertical scroll bars to view portions of a table that are outside the boundaries of the dialog box. In some cases scroll bars may not be available; instead, increase the size of the dialog box to display more data.

Deleting rows of data

Click any cell within the row you want to delete, or select multiple cells in consecutive rows. Then, while holding the cursor over the dialog box containing the table, click mouse button 3 and select **Delete Rows** from the menu that appears. The row or rows disappear; if the rows are numbered, Abaqus/CAE automatically renumbers the remaining rows.

You cannot delete rows from tables that display matrices or tensors of fixed size, such as those used in the orthotropic or anisotropic elasticity data input forms in the Property module.

Creating X–Y data from table data

While you are creating a material in the Property module, you can use the data in a table to create *X*–*Y* data. You can then use the Visualization module to plot the *X*–*Y* data and to visually check its validity. To create an *X*–*Y* data object, click mouse button 3 while holding the cursor over the table; then select **Create XY Data** from the menu that appears. The **Create XY Data** dialog box appears. In this dialog box, do the following:

- Enter the name of the *X*–*Y* data to create.
- Specify the column number containing the *X*-values and the column number containing the *Y*-values.
- Click **OK**. Abaqus reads the data values from the table into the *X*–*Y* data. Abaqus/CAE retains saved *X*–*Y* data only for the duration of the session.

To view the *X*–*Y* data, do the following:

- From the module list on the context bar, select **Visualization**.
- From the main menu bar, select **Tools->XY Data->Plot**, and select the *X*–*Y* data from the pull-right menu.

For more information, see *X–Y plotting*.

Clearing the table

You can delete all data from a table. While holding the cursor over the table, click mouse button 3 and select **Clear Table** from the menu that appears. The table data disappear.

Additional information

• *Interacting with dialog boxes*

Customizing fonts

The **Select Font** dialog box allows you to customize the font of certain kinds of text; for example, you can use this dialog box to customize the font that appears in viewport annotations. A similar dialog box is used to customize the font of the Visualization module labels and titles.

The **Select Font** dialog box allows you to specify and preview the following:

- Proportional or fixed fonts.
- The font family.
- The font size, in points.
- Regular, bold, or italic font.

The available options vary depending on which fonts are installed on your system.

1. Display the **Select Font** dialog box for the text that you want to customize. For more information, see the following sections:
   - *Customizing X–Y plot axes*
   - *Customizing the X–Y plot legend*
   - *Customizing viewport annotations*”
   - *Setting the label font*
2. Select the desired font and properties.

   A preview of the selected font appears in the **Sample** area of the **Select Font** dialog box.
3. In the **Apply To** field of the **Select Font** dialog box, toggle on the items to which the selected font will apply. The **Apply To** field does not appear unless there are multiple items to which the font can apply.
4. Click **OK** to accept your changes and to close the **Select Font** dialog box.

Additional information

• *Customizing viewport annotations*
• *Setting the label font*

Customizing colors

The **Select Color** dialog box allows you to customize the color of many objects in Abaqus/CAE. For more information about the objects that you can change, see the following sections:

- *Customizing the view triad*
- *Choosing background colors*
- *Selecting overall element and surface edge color*
- *Coloring elements with no results*
- *Customizing the legend*
- *Customizing the title block*
- *Customizing the state block*

The current color is displayed on the left side of the **Select Color** dialog box, below the eyedropper tool. You can use the methods in the **Select Color** dialog box to update the displayed color. The color is not updated elsewhere until you click **OK** to accept your changes and to close the **Select Color** dialog box. You can choose from the following methods of color selection:

Color palette

Twenty-four common colors are displayed in boxes along the bottom of the dialog box. Click a color to select it; you cannot modify the palette to show different colors.

Eyedropper tool

The eyedropper tool is located on the left side of the dialog box. When you click the eyedropper tool, the cursor changes to crosshairs. The next time you click mouse button 1 anywhere on the computer screen, Abaqus/CAE selects the color at the cursor position.



Note:

The cursor returns to its normal form if you move it outside the Abaqus/CAE application window, but you can still select a color by clicking mouse button 1.

Color wheel

The color wheel and brightness control are located in the **Wheel** tab. A black dot indicates the position of the currently selected color, regardless of the method that was used to select it. Click anywhere on the wheel to select a new color. Move the vertical slider to change the brightness; as you move the slider downward, Abaqus/CAE adds black to the selected color.

RGB controls

RGB (Red, Green, and Blue) controls are located in the **RGB** tab. The RGB settings match the color displayed on the left side of the **Select Color** dialog box, regardless of the method that was used to select it. You can move the sliders or enter values from 0 to 255 to mix the three colors of light and produce the full color spectrum. 0, 0, 0 is black (no light); and 255, 255, 255 is white (full intensity, full spectrum light).

HSV controls

HSV (Hue, Saturation, and Value) controls are located in the **HSV** tab. The HSV settings match the color displayed on the left side of the **Select Color** dialog box, regardless of the method that was used to select it. The Hue control ranges from 0 to 360, and changing the setting corresponds to moving the black dot around the perimeter of the color wheel (0 and 360 are both red). The Saturation control ranges from 0 to 100 and varies the amount of the selected color added to the background color. The Value control indicates the background color; 0 is black, 100 is white.

CMY controls

CMY (Cyan, Magenta, and Yellow) controls are located in the **CMY** tab. The CMY settings match the color displayed on the left side of the **Select Color** dialog box, regardless of the method that was used to select it. You can move the sliders or enter values from 0 to 255 to mix the three colors of tint and produce the full color spectrum. The CMY controls work like adding tint to paint; 0, 0, 0 is white (no tint), and 255, 255, 255 is black (all tint).

Color list

The color list is located in the **List** tab. You can choose from several hundred colors, including shades of gray. The color list provides you with a more extensive range of colors than the color palette, but it does not provide you with the full color spectrum.

1. Open the dialog box that contains the settings for the object that you want to change.
2. Click the color sample  for the object that you want to customize.

   Abaqus/CAE displays the **Select Color** dialog box.
3. Use one of the methods in the dialog box to select a new color.

   A preview of the selected color appears on the left side of the **Select Color** dialog box, below the eyedropper tool.
4. Click **OK** to accept your changes and to close the **Select Color** dialog box.

   Abaqus/CAE returns you to the originating dialog box and updates the color sample to display the color that you selected.

When you click **OK** or **Apply** in the originating dialog box, Abaqus/CAE updates the color in the viewport.

Additional information

- *Customizing model labels*
- *Customizing viewport annotations*

Using file selection dialog boxes

File selection dialog boxes allow you to select files from lists that are filtered based on file type or location. To use a file selection dialog box, you first choose the type of file to open and then specify the directory to list. Abaqus/CAE refreshes the dialog box to list only files that meet your criteria. From this list, you select the file to open.

The dialog box for selecting model databases or output databases is shown in *Figure 1*.

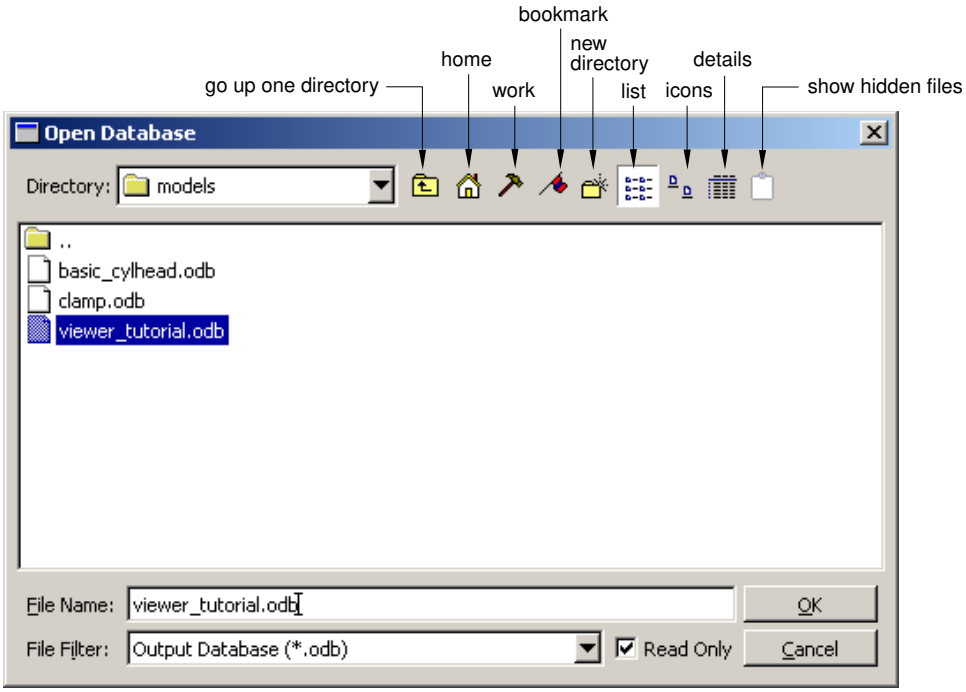


Figure 1: Selecting a model database or an output database.

Note:

In Abaqus/Viewer you can open only output database files; therefore, **Output Database (*.odb)** is the only type available in the **File Filter** field.

Similar file selection dialog boxes appear when you perform other **File** menu functions, such as importing a part or printing to a file.

Use the following techniques to select the file of your choice:

Filtering the file list according to file type

File selection dialog boxes contain **File Filter** fields, which allow you to select the file extension of interest. For example, the **File Filter** selection in *Figure 1* is **Output Database (*.odb)**. Therefore, only files with the extension `.odb` appear in the list in the center of the dialog box.

Using wildcards to search for a file name

You can use a wildcard filter to search for partial names of files. A wildcard search is helpful when you have a large number of files stored in the same directory. Wildcard searches also override the file extensions (**File Filter** field, as described above), allowing you to open files saved with nonstandard file extensions.

To use a wildcard search, enter a partial file name into the **File Name** field using one of the following forms:

?

matches a single character

*****

matches zero or more characters

abc

matches a single character, but it must be one of the characters listed

^abc or !abc

matches a single character, but it must *not* be one of the characters listed

a-zA-Z

matches a single character, but it must be within the ranges provided

^a-zA-Z or !a-zA-Z

matches a single character, but it must *not* be within the ranges provided

pat1|pat2 or pat1,pat2

matches either pat1 or pat2

(pat1|pat2) or (pat1,pat2)

matches either pat1 or pat2, and the patterns may be nested

You can combine several wildcard filters to further narrow your search. For example, entering `[abc]*.(cae,odb)` will list all files beginning with `a`, `b`, or `c` and having a `.cae` or `.odb` file extension.

Specifying the directory from which to select a file

By default, the **Directory** field shows the directory in which you started Abaqus/CAE. If you want to view a list of files from a different directory, you can click the directory name in the list to view directories within the current path or you can click the arrow next to the **Directory** field to access other paths that are available on your system. In addition, icons at the top of the dialog box allow you to do the following (keyboard shortcuts are shown in parentheses when available):

- Go up one directory level (**[Backspace]**).
- Access your system default, or **Home**, directory (**[Ctrl] + H**).
- Access the **Work** directory (**[Ctrl] + W**). The work directory is the directory from which you started Abaqus/CAE unless you specified the directory using **File->Set Work Directory**.
- Set or use **Bookmarks** to any directory on your system.
- Create a new directory (**[Ctrl] + N**).

The **Directory** field includes a **Network connectors** item. If you have created and started a network ODB connector, you can use this item to access a remote directory and to open a remote output database. For more information, see *Creating a network ODB connector*.

Selecting a file

To select and open a file, double-click the file name of interest from the list. You can also begin typing the file name; the cursor will reposition to the matching location in the file list, and the first file starting with the letters you typed will be selected. Alternatively, you can enter the entire directory path and file name of interest directly in the **File Name** field and then click **OK**. Icons at the top of the dialog box allow you to change the displayed file format to one of the following (keyboard shortcuts are shown in parentheses):

- A list (**[Ctrl] + `S`**).
- Icons (**[Ctrl] + `B`**).
- A detailed list (**[Ctrl] + `L`**).

The icon farthest to the right allows you to display or suppress “hidden” files.

Selecting multiple items from lists and tables

In some Abaqus/CAE dialog boxes it is necessary to select an item from a list or a table before you can perform certain functions. For example, if you want to plot *X*–*Y* data, you must first select the data object of your choice from the list in the **XY Data Manager**, shown in *Figure 1*, and then click **Plot**.

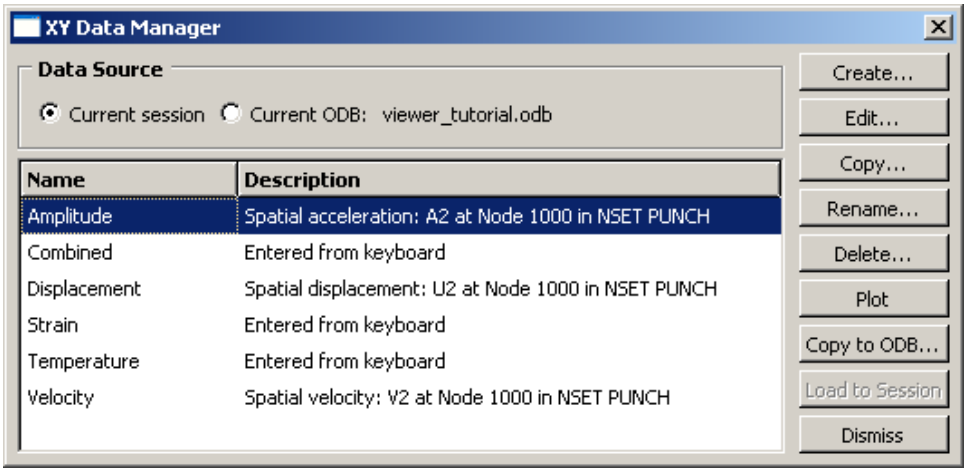


Figure 1: Single item selected.

Some functions allow you to operate on more than one item. For example, if you wanted to delete the first two data objects in the manager shown in *Figure 1*, you could select them both and then click **Delete**.

To select a single item from a list, you need only click that item in the dialog box. To select a single item from a table, click the table row heading. To select multiple items, you can use the following techniques:

Selecting consecutive items from a list or table

Click the first item of interest from a list or row heading from a table and then, while continuing to hold down mouse button 1, drag the cursor over the remaining items. Release the mouse button when all of the items of interest are selected. For example, consecutive items are selected in *Figure 2*.

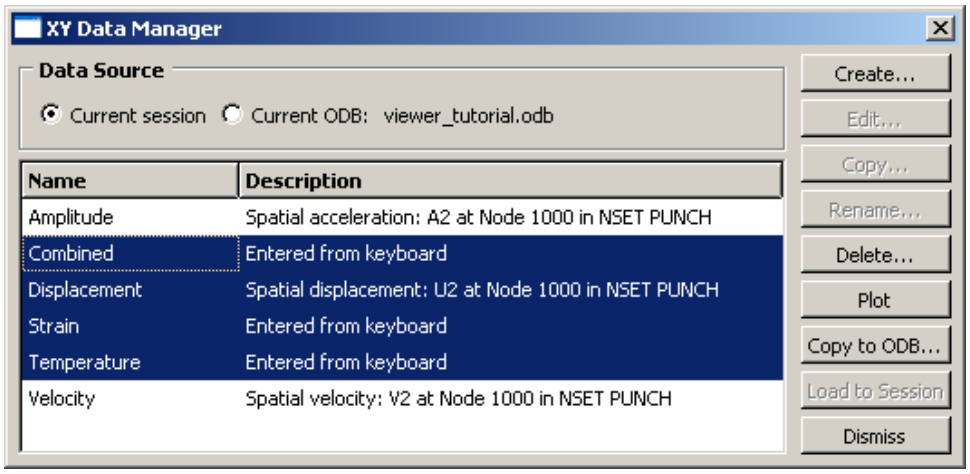


Figure 2: Consecutive items selected.

Another way to select consecutive items is to click the first item of interest from a list or row heading from a table and then **[Shift] + Click** the last item of interest. All items between the first and the last are selected automatically.

Selecting nonconsecutive items from a list or table

Click the first item of interest from a list or row heading from a table and then **[Ctrl] + Click** any other items you want to select. For example, nonconsecutive items are selected in *Figure 3*.

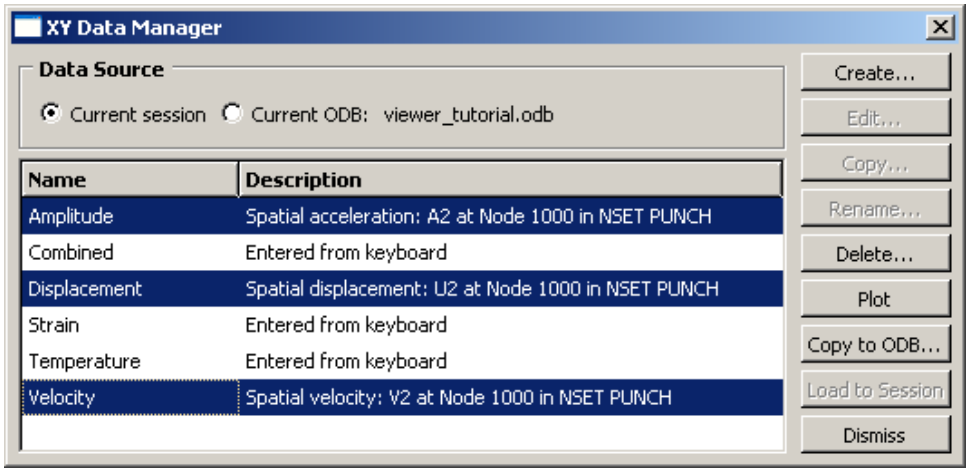


Figure 3: Nonconsecutive items selected.

Canceling a selection

You can **[Ctrl] + Click** previously selected items to remove them from your selection. For example, if you **[Ctrl] + Click**`Displacement` in the list shown in *Figure 3*, that data object is no longer selected, as shown in *Figure 4*.

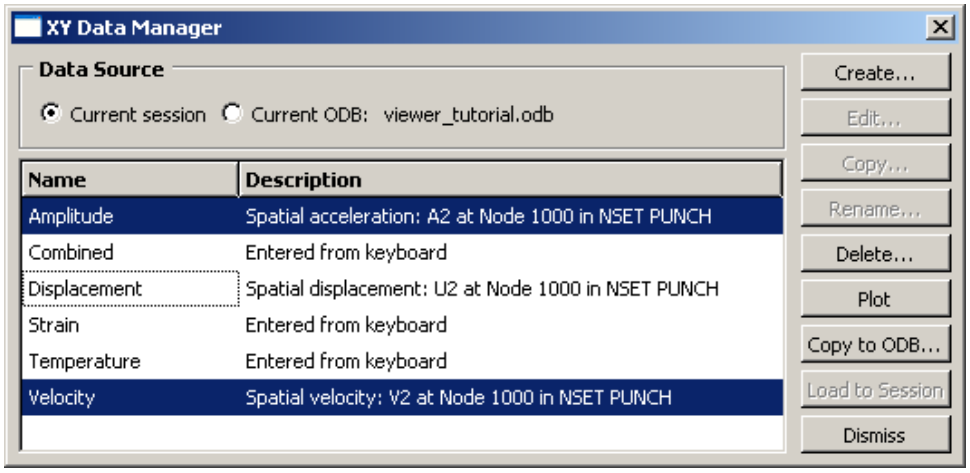


Figure 4: Individual item removed from selection.

Certain functions in a dialog box may become unavailable when you select multiple items. For example, the **Edit**, **Copy**, and **Rename** functions in the **Data Manager** shown in *Figure 4* are valid only for individual data objects. When you select multiple data objects, these three functions become unavailable.

Using keyboard shortcuts

You can use the keyboard instead of the mouse to perform most actions within the Abaqus/CAE main window and dialog boxes. The following actions have keyboard shortcuts:

Context-sensitive help

Press **[F1]** to display context-sensitive help concerning the currently selected object in the Abaqus/CAE main window or dialog box. For more information on using **[F1]** for context-sensitive help, see *Displaying context-sensitive help*.

Menus

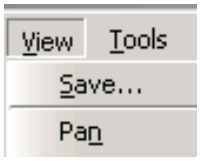
You can display a particular menu by pressing the **[Alt] +** key in combination with the underlined character in that menu's name. For example, the letter `V` is underlined in the **View** menu in the main menu bar:



Therefore, you can type **[Alt] + V** to display the **View** menu.

Menu items

Once the menu is displayed, you can select a particular menu item by continuing to press the **[Alt] +** key and pressing the underlined character in that menu item's name. For example, the letter `n` is underlined in **Pan** in the **View** menu:



Therefore, you can type **[Alt] + V** to display the **View** menu and then, without releasing the **[Alt] +** key, type `n` to select **Pan**.

Model Tree and Results Tree

The Model Tree and Results Tree contain keyboard shortcuts that allow you to navigate through the tree and toggle its display on and off. For more information, see *The Model Tree*.

Understanding and using toolboxes and toolbars

This section explains how to use the toolbox windows and toolbars to perform common functions within a module or toolset or on the canvas.

In this section:

- *What are toolboxes and toolbars?*
- *Using toolboxes and toolbars that contain hidden icons*

What are toolboxes and toolbars?

Toolboxes and toolbars are collections of icons that provide quick access to commonly used Abaqus/CAE functions. For example, the Visualization module toolbox contains icons representing the tools used to generate different kinds of plots. The Visualization module toolbox is shown in *Figure 1*. All module toolboxes are available immediately to the left of the drawing area as soon as you enter the module.



Figure 1: The Visualization module toolbox.

Toolbars also contain collections of icons to access Abaqus/CAE functions. Toolbars provide access to supporting functions that help you save, manipulate, and make selections from a model; whereas toolboxes contain functions critical to creating or changing a model. In addition to tool icons, toolbars may also contain lists of options related to a tool. For example, the color mapping list in the **Color Code** toolbar contains various methods for coloring the objects displayed in the current viewport. You can also customize toolbar contents, move toolbars to new locations, or close them (for more information, see *The Customize toolset*). Toolboxes cannot be moved or hidden.

To obtain a short description of a tool, place the cursor over that tool for a moment; a small box containing a description, or "tooltip," will appear. Tooltips are not available for icons that appear dimmed; to get information on those icons, use *context-sensitive help* instead.

Additional information

- *Understanding and using toolboxes and toolbars*

Using toolboxes and toolbars that contain hidden icons

In some toolboxes, such as the Job module toolbox, all tool icons are immediately visible; however, most toolboxes contain hidden icons to conserve space. Since there is more space above the canvas, and since you can move or hide toolbars to meet your needs, most toolbars do not contain hidden icons.

Any icon that includes a small triangle in its lower right corner conceals a group of icons whose function is closely related to that of the visible icon.

1. Click and hold any icon that includes a triangle in its lower right corner.

   Icons for all the tools that are closely related to the original icon appear. For example, *Figure 1* shows the top portion of the Part module toolbox with all of the icons revealed that are used for creating round or chamfered corners.

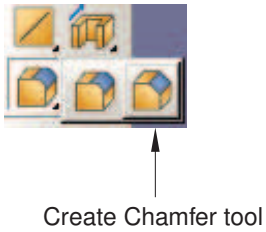


Figure 1: Part module toolbox with round and chamfer icons displayed.

2. Drag the cursor to the desired icon, and release the mouse button.

   The selected icon replaces the icon that was visible originally, and you can begin using the corresponding tool immediately.

Additional information

• *Understanding and using toolboxes and toolbars*

Managing objects

Managers are dialog boxes you use to manage all objects of a given type associated with the current model or session; examples of such objects include materials, parts, steps, display groups, and *X*–*Y* data objects.

In addition, you can use the **Model Manager** to manage the models contained in the current model database. This section describes basic and step-dependent managers and how you can use them in Abaqus/CAE.

In this section:

- *What are basic managers?*
- *What are step-dependent managers?*
- *Suppressing and resuming objects*
- *Understanding the status of an object in a step*
- *Terms describing object status*
- *Modifying the history of a step-dependent object*
- *Understanding modified step-dependent objects*
- *What happens when deleted objects are referred to?*
- *Managing objects using manager dialog boxes*
- *Managing objects using manager menus*
- *Copying step-dependent objects using manager dialog boxes*
- *Changing the status of an object in a step*
- *Editing step-dependent objects*

What are basic managers?

Basic managers consist of a list of objects and a series of buttons; you use the buttons to perform tasks on the objects you select from the list or to add new objects to the list.

Figure 1 shows the **Material Manager**, which is an example of a basic manager used in Abaqus/CAE.

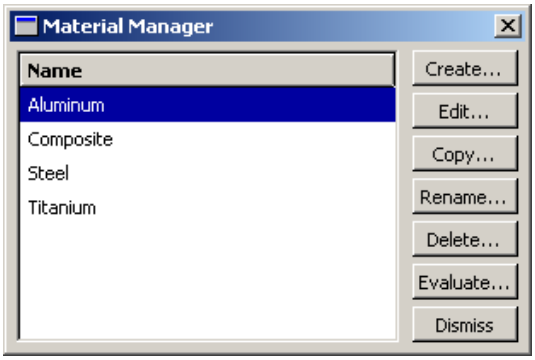


Figure 1: The Material Manager.

The list box on the left shows all the materials that you have defined within the context of the current model. You use the buttons on the right to create new material definitions and to edit, copy, rename, and delete existing material definitions. The **Dismiss** button is used to close the manager dialog box.

Often, the manager provides more information about an object than just its name; for example, in the Job module, the **Job Manager** provides information about currently executing jobs and provides buttons that allow you to write input files, submit jobs, monitor the analysis, or view output files for a given job. The **Job Manager** is shown in *Figure 2*.

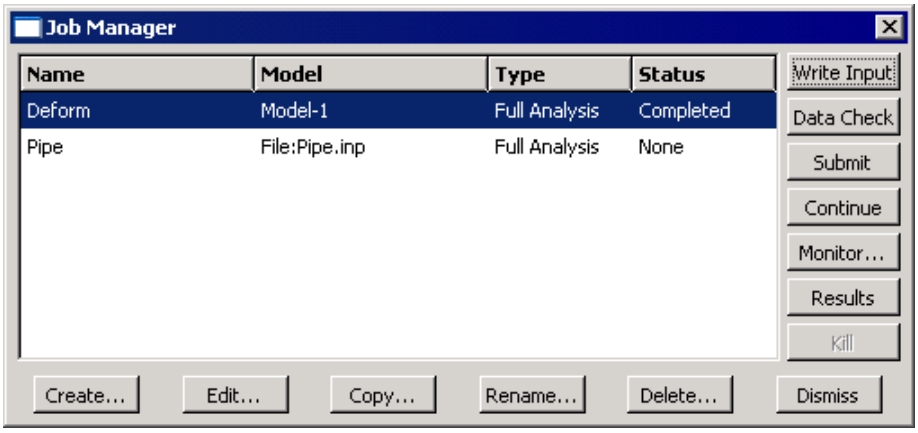


Figure 2: The Job Manager.

Most tasks you can perform with a manager can also be performed using the pull-down menus available from the main menu bar; for example, *Figure 3* shows the menu items that correspond to the **Job Manager**.

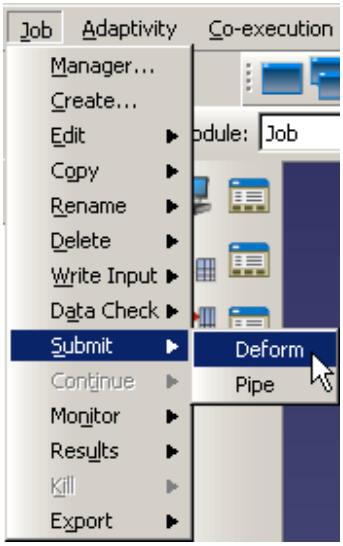


Figure 3: Menu items that correspond to the Job Manager.

After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box. In addition, most of the tasks you can perform with a manager can be performed by clicking mouse button 3 on an object in the Model Tree. For more information, see *Working with the Model Tree and the Results Tree*.

The decision whether to use menus, dialog boxes, or the Model Tree is yours. In general, menus are more convenient if you are performing isolated operations; the advantages of manager dialog boxes become apparent when you are performing several operations in sequence, when you need to browse through a long list of objects, or when you need quick access to the additional information that is displayed by some managers. The Model Tree provides you with a graphical overview of your model and allows you to perform operations without changing modules. In addition, the Model Tree allows you to use drag-select to select multiple items; for example, you can select multiple sets to merge or multiple parts to delete.

Additional information

- *Managing objects*
- *What are step-dependent managers?*

What are step-dependent managers?

Like the basic managers described in *What are basic managers?*, step-dependent managers contain a list of all of the objects of a certain type that you have created, as well as **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons that you can use to manipulate existing objects and to create new ones.

However, the types of objects that appear in step-dependent managers are those that you can create and, in some cases, modify, suppress, and deactivate in particular analysis steps. Therefore, unlike basic managers, step-dependent managers contain additional information concerning the history of each object listed in the manager. Step-dependent managers display how these objects propagate from one step to another during the course of an Abaqus analysis. (For information on steps and multiple-step analyses, see *Defining an Analysis*.)

The following step-dependent managers exist in Abaqus/CAE:

In the Load module:

- **Load Manager**
- **Boundary Condition Manager**
- **Predefined Field Manager**

In the Interaction module:

- **Interaction Manager**

In the Step module:

- **Field Output Requests Manager**
- **History Output Requests Manager**
- **Adaptive Mesh Constraint Manager**

For example, the **Load Manager** is shown in *Figure 1*.

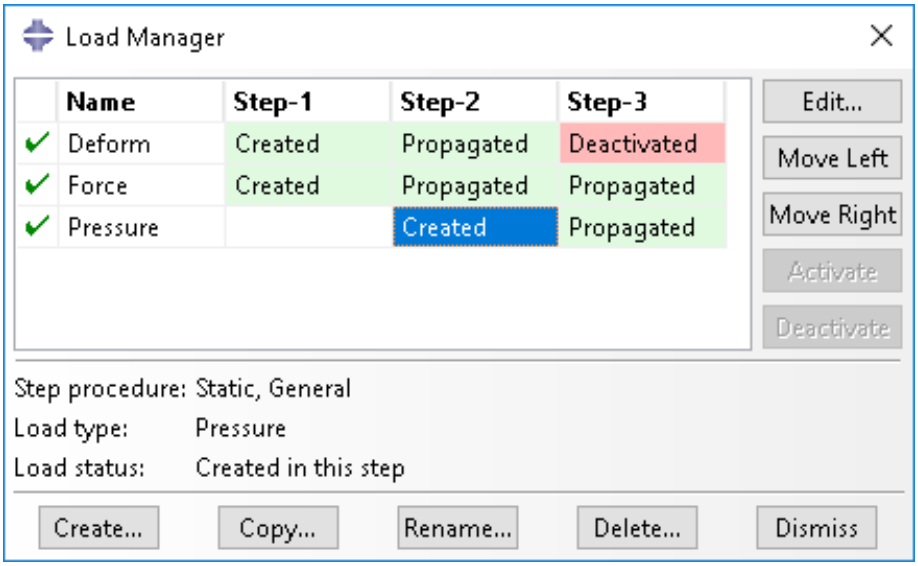


Figure 1: The Load Manager.

This manager displays an alphabetical list of existing loads along the left side of the dialog box. The names of all the steps in the analysis appear along the top of the dialog box in the order of execution. The table formed by these two lists displays the status of each load in each step. (For information on creating and deleting steps, see *The Step module*.)

If you click one of the cells in the table, that cell becomes highlighted, and the following information related to the cell appears in the legend at the bottom of the manager:

- The type of analysis procedure carried out in the step in that column.
- Information about the step-dependent object in that row.
- The status of the step-dependent object in that step (the same information that appears in the cells of the table except in more detail in some cases).

You can use the icons in the column along the left side of the manager to suppress objects or to resume previously suppressed objects for an analysis. For more information, see *Suppressing and resuming objects*.

The buttons along the right side of the manager allow you to manipulate objects in the steps that you select. For example, if you click **Edit** in the Load Manager shown above, an editor appears in which you could modify the load named `Force` in `Step-1`. The other buttons—**Move Left**, **Move Right**, **Activate**, and **Deactivate**—allow you to change the status of an object in a particular step.



Note: The **Activate** and **Deactivate** buttons are not available in the **Predefined Field Manager**.

For more information, see *Modifying the history of a step-dependent object*, *Changing the status of an object in a step*, and *Editing step-dependent objects*.

You can resize the columns of the table by dragging the dividers between the column headings to the right or left. You can also increase the size of the dialog box by dragging the sides of the box. If the analysis includes many steps or many step-dependent objects, increasing the size of the dialog box allows you to view more rows and columns without having to use the scroll bars.

Additional information

- *The output request managers*
- *Managing objects in the Interaction module*
- *Managing prescribed conditions*

Suppressing and resuming objects

When performing an analysis, you may want to study the effects of different combinations of objects, such as loads, or you may want to temporarily exclude an object from the model, such as excluding a boundary condition or a constraint in a design analysis. You can create a model that includes all of the objects and then suppress the objects that you want to exclude from the model prior to the analysis. Suppressed objects are not written to the input file and are treated as deleted objects. You should review your model for any references to suppressed objects. For more information, see *What happens when deleted objects are referred to?*.

You can suppress step-dependent objects, constraints (in the Interaction module), section assignments (in the Property module), and features. After you create a suppressible object, the manager dialog box displays a green check mark in the column along the left side of the manager next to the name of the object. You can suppress an object from the manager by clicking the green check mark next to the object. For example, if you click the green check mark to the left of the load named `Force` in the Load Manager shown in *Figure 1*, the icon changes to a red "X" and the cells displaying the status of the load in each step are shaded gray to indicate that the load is suppressed, as shown in *Figure 1*.



Note:

There is no manager associated with features; you can suppress or resume features using the popup menus in the Model Tree.

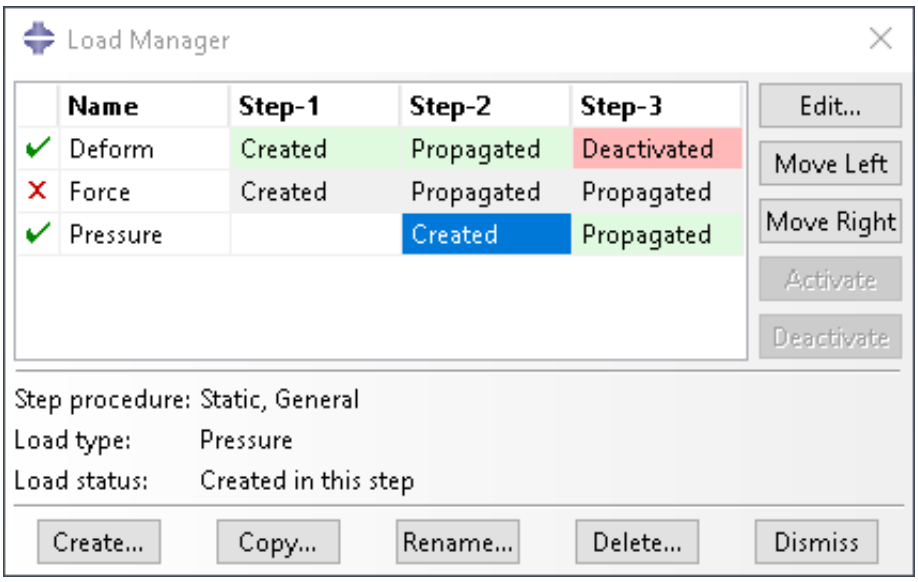


Figure 1: The Load Manager indicates that the load named `Force` is suppressed.

You can also select **Suppress->*object*** in the appropriate menu from the main menu bar to suppress an object. For example, to suppress the load named `Force` shown in *Figure 1*, you would select **Load->Suppress->Force** from the main menu bar of the Load module.

You cannot edit suppressed objects; however, you can copy, rename, and delete them. Symbols for suppressed objects are not displayed in the viewport.

You can resume an object that was previously suppressed. If you attempt to resume an object that is not valid for a given procedure type, Abaqus/CAE displays an error message. You can use the manager or the **Resume** menu item from the main menu bar to resume the object. In the manager, click the red "X" to change the icon back to a green check mark and to remove the cell shading. Symbols for resumed objects are displayed in the viewport.

You can also use the Model Tree to suppress or resume an object by clicking mouse button 3 on the object and selecting **Suppress** or **Resume** from the menu that appears. The Model Tree displays a red "X" next to an object to indicate that it is suppressed. For more information, see *The Model Tree*.

Additional information

- *What are step-dependent managers?*
- *The output request managers*
- *Managing objects in the Interaction module*
- *Managing prescribed conditions*

Understanding the status of an object in a step

A model can contain a sequence of analysis steps. When you create an object in a step, that object may or may not continue to be active in any of the following steps. The activity (or inactivity) of an object in any particular step is called its “status” in that step.

For example, *Figure 1* shows the status of a load in a series of general static analysis steps.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Load 1	Created	Propagated	Modified	Propagated	Deactivated	Deactivated

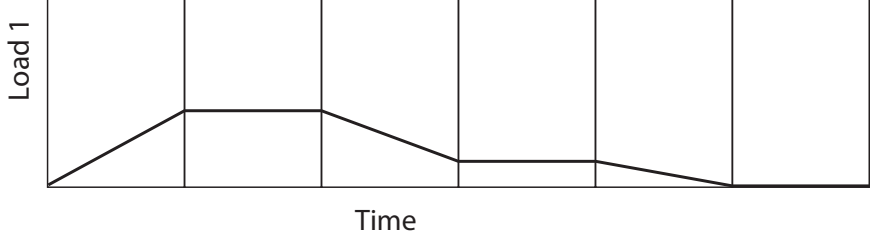


Figure 1: The analysis history of a load.

The load in this example is created in Step 1; therefore, the status of the load in Step 1 is `Created`. Since Step 1 is a general static step, the load's magnitude is ramped up over the course of the step. If the load continues to be active in Step 2, its status in Step 2 is `Propagated` and its magnitude remains constant throughout that step. If you edit the load in Step 3, its status in Step 3 becomes `Modified` and its magnitude ramps to the new value over the course of the step. If the modified version of the load continues to be active in Step 4, its status in Step 4 (as in Step 2) is `Propagated` and the value is constant. If you deactivate the load in Step 5, its status in Step 5 is `Deactivated` and its magnitude ramps down to zero. The load remains deactivated in Step 6.

For detailed explanations of the terms used to describe object status, see *Terms describing object status*.

Additional information

- *What are step-dependent managers?*

Terms describing object status

Abaqus/CAE uses the following terms to describe the status of objects in particular steps:

Created

The object was created and becomes active in this step. The point in the step at which a prescribed condition becomes active depends on the amplitude variation associated with that step. For more information, see "Prescribed conditions" in *Defining an Analysis*.

Computed

The analysis products will compute the value of the object in this step.

Modified

The definition of the object has been modified in this step. The variation of a prescribed condition over the course of the step depends on the amplitude variation associated with that step.

Propagated

The object was created, modified, or computed in an earlier step of the analysis and continues to be active in this step.

Deactivated

The object has been deactivated in this step or in a previous step. It will remain deactivated in all subsequent steps until you reactivate it. You cannot deactivate an object in the step in which it was created. The point in the step at which a prescribed condition becomes inactive depends on the amplitude variation associated with that step. For more information, see "Prescribed conditions" in *Defining an Analysis*.

You cannot deactivate predefined fields; a deactivated status for a predefined field means that the field has been reset to the value specified in the initial step. The point in the step at which an object resumes its initial value depends on the amplitude variation associated with that step. For more information, see "Prescribed conditions" in *Defining an Analysis*.

N/A

The object does not have any effect on the calculations for this step.

The following terms apply only in linear perturbation steps:

Built into base state

Any active object created in a preceding general analysis step will be part of the base state and cannot be changed during the linear perturbation step.

Propagated from base state

Objects created in a previous general step will be part of the base state for this procedure but can be modified or deactivated by the user.

Deactivated from base state

Objects created in a previous general step are deactivated in this linear perturbation step. The deactivated state applies only to the linear perturbation step and does not propagate to the remaining steps.

For information on linear perturbation steps, see *General and Perturbation Procedures*.

The following term applies only in modal dynamics steps:

Built into modes

Boundary conditions that are active in a preceding frequency analysis step are used in the calculation of modes and will therefore be built into the modes for mode-based linear perturbation procedures and subspace dynamic procedures. During these mode-based and subspace dynamic procedures the boundary condition cannot be changed.

For information on modal dynamics steps, see *Transient Modal Dynamic Analysis*.

Figure 1 shows the color codes assigned to various object statuses. The table cells in step-dependent managers are colored accordingly to indicate the object status.

Color codes	Statuses
	Created
	Computed
	Modified
	Propagated
	Deactivated
	N/A
	Built into base state
	Modified from base state
	Propagated from base state
	Deactivated from base state
	Built into modes

Figure 1: Color codes for object statuses.

Modifying the history of a step-dependent object

You can modify the analysis history of an object by using the five buttons aligned along the right side of the step-dependent manager: **Edit**, **Move Left**, **Move Right**, **Activate**, and **Deactivate**. (For information on how to use these buttons, see *Changing the status of an object in a step*.) The use of these buttons may be restricted depending on the nature of each step and the status of the object in the steps.

The following list describes the rules for modifying the history of a step-dependent object:

Changing the step in which an object becomes active

You can change the step in which an object becomes active by moving the `Created` status to that step. You can move the `Created` status of an object to any previous general step, or you can move the `Created` status to the following general step if its status in the following step is `Propagated`.

For example, you could select the `Created` status of Load1 in the load manager table below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1		Created	Propagated	Propagated	Propagated

If you moved the `Created` status to Step 1, the table would change as shown below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1	Created	Propagated	Propagated	Propagated	Propagated

If you moved the `Created` status to Step 3, the table would change as shown below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1			Created	Propagated	Propagated



Note:

If an object is created in a linear perturbation step, its `Created` status cannot be moved.

Modifying an object

You can modify an object when its status is `Propagated`; the object's status in that step changes to `Modified`.

Moving the modifications of an object to another step

You can transfer the modifications of an object to another step by moving the object's modified status to that step. You can move the `Modified` status of an object to the previous general step or to the following general step if the status of the object in those steps is `Propagated`.

For example, you could select the `Modified` status of Load1 in the load manager table below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1		Created	Propagated	Modified	Propagated

If you moved the `Modified` status to Step 3, the table would change as shown below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1		Created	Modified	Propagated	Propagated

If you moved the `Modified` status to Step 5, the table would change as shown below.

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1		Created	Propagated	Propagated	Modified

Deactivating an object

You can deactivate an object when its status is `Propagated` or `Modified`; the object's status in that step is shown as `Deactivated`, and the object is inactive in any following steps.



Note:

You cannot deactivate predefined fields using the **Predefined Field Manager**; you must select **Reset to initial** in the predefined field editor (for example, see *Defining a temperature field*).



Warning:

If you deactivate an object in a step in which its status is `Modified`, the modifications to the object are lost. If you later reactivate the object in that step, the original propagated version of the object becomes active in that step and in all subsequent steps.

Reactivating an object

You can reactivate an object that has a `Deactivated` status; however, the **Activate** button is available only in the step in which the object is first deactivated (for example, Step 3 in the following table).

	Step 1	Step 2	Step 3	Step 4	Step 5
Load1	Created	Propagated	Deactivated		

When you reactivate the load in the example above, its status in Step 3 and in all following steps changes to `Propagated`.

The following rules apply to linear perturbation steps:

Deactivating a boundary condition whose status is `Propagated from base state`

You can deactivate an object whose status is `Propagated from base state`; the object's status in the linear perturbation step changes to `Deactivated from base state`. The status `Propagated from base state` cannot be moved to other steps.

Reactivating a boundary condition whose status is `Deactivated from base state`

You can reactivate an object whose status is `Deactivated from base state`; the object's status in the linear perturbation step changes to `Propagated from base state`. The status `Propagated from base state` cannot be moved to other steps.

Objects whose status is `Built into base state`

The status `Built into base state` cannot be changed directly.

For information on linear perturbation steps, see *General and Perturbation Procedures*.

For information on the propagating behavior of output requests from general and linear perturbation analysis steps, see *Propagation of output requests*.

You can use the Model Tree to view the status of a step-dependent object, to edit the object, and to deactivate and reactivate the object. However, you must use the step-dependent manager to modify the history of an object by moving it right or left in the sequence of steps. For more information, see *The Model Tree*.

Additional information

- *What are step-dependent managers?*

Understanding modified step-dependent objects

When you edit an object in the step in which it was created, you change the definition of the object in all of the steps in which it is active. In some cases you can also edit an object in steps in which its status is `Propagated` or `Modified`. In these cases the object's definition varies according to the analysis step.

The effects of editing a step-dependent object are summarized below.

If the status of the object is `Created` in the selected step:

- Modifications that you make to the object in this step become effective in this step and propagate through all subsequent steps in which the condition is active unless you modify the object again in a later step.
- The status of the object remains `Created` in the selected step and also remains unchanged in all subsequent steps. For more information, see *Understanding the status of an object in a step*.

If the status of the object is `Propagated` or `Modified` in the selected step:

- Modifications that you make to the object in this step become effective in this step and propagate through all subsequent steps in which the object is active.
- The status of the object becomes (or remains) `Modified` in this step and remains unchanged in all other steps. (In other words, if the status of the object in the following step was `Propagated` before modification, its status in the following step remains `Propagated` after modification.) For example, the load applied over a sequence of general static analysis steps in *Figure 1* has been modified in Step 3; the modifications remain in effect in Step 4 even though the status in Step 4 is `Propagated`. For more information, see *Understanding the status of an object in a step*.
- When you modify the data in any editor other than the Interaction editor, Abaqus/CAE indicates in the editor which data have been modified. These indications disappear if you change the data in the editor back to their original values.

In some cases you cannot edit a particular aspect of an object's definition because it must be consistent for the analysis to proceed correctly. For example, although you can modify the magnitude of a load in any analysis step, you cannot modify the region to which the load is applied. The areas in an editor that specify this kind of restricted data are unavailable in all steps except the one in which the object was created.

Additional information

• *What are step-dependent managers?*

What happens when deleted objects are referred to?

You should take care when deleting or renaming objects, such as materials and amplitudes, that may be referred to by other objects. For example, if you delete or rename a material, the sections that refer to the material become inconsistent. To resolve the missing reference, you can edit the section and refer to a new material, or you can create a new material with the same name as the deleted material.

Table 1 lists objects that are commonly referred to by other objects.

Table 1: Objects that are commonly referred to by other objects.

This object	Can be referred to by these types of objects
Material	Section
Profile	Section, skin
Section	Section assignment
Interaction	Output request, contact controls
Interaction property	Interaction
Amplitude	Load, predefined field, boundary condition, interaction
Connector section	Connector section assignment
Region (set or surface)	Boundary condition, predefined field, load, interaction, constraint, connector section assignment, output request, section assignment, beam section orientation, material orientation, output request, DOF monitor, adaptive mesh domain
Load	Load case, output request
Boundary condition	Load case
Datum coordinate system	Boundary condition, connector section assignment, material orientation, constraint
Datum plane	Load
Datum axis	Load
Datum point	Constraint
Part instance	Constraint
Part	Part instance

Parts and part instances behave slightly differently. If you delete a part after you have instanced the part in the Assembly module, Abaqus/CAE suppresses the part instance in the assembly. You can delete the instance from the assembly. Alternatively, if you then create a new part that uses the same name, you can unsuppress the part instance to include it in the assembly. In addition, if you rename a part or a datum, objects that refer to the part or datum refer to the new name; and, as a result, the reference does not become inconsistent.

Managing objects using manager dialog boxes

Abaqus/CAE provides you with a set of managers that list all the objects defined in the current model or session such as parts, stand-alone sketches, materials, sections, steps, display groups, and *X–Y* data objects. In addition, the **Model Manager** lists all the models defined in the current model database.



Note:

For more information on specific managers and where they are located, see the documentation for the particular module in which you are interested.

Use the buttons in the manager dialog box to manage the list of objects.

1. To display a manager, do one of the following:
   - To display a manager associated with a module, select **Manager** from the appropriate menu on the main menu bar. For example, to display the **Section Manager** while you are working in the Property module, select **Section->Manager** from the main menu bar.
   - To display a manager associated with a toolset, select **Tools->*Toolset*->Manager** from the main menu bar. For example, to display the **Set Manager**, select **Tools->Set->Manager** from the main menu bar.
   - To display a step-dependent manager associated with an object in the Model Tree, click mouse button 3 on the object and select **Manager** from the menu that appears.
   - To display the **Model Manager**, select **Model->Manager** from the main menu bar.

   The manager appears and displays a list of objects in the current model or session. The list contains the name of each object and, in some cases, information about each object. For example, the **Part Manager** lists the name of each part, its status, its type, and the modeling space in which it was created.
2. To manage an existing object, select the object or objects of interest from the list in the manager and then click the appropriate button. (For example, to delete an object, select that object's name from the list and then click **Delete**.)

   In most cases a dialog box appears; for example, when you click **Rename**, the dialog box asks for the new name of the selected object.
3. If a dialog box appears, provide the requested information and click **OK**.
4. Click **Dismiss** to close the manager.



Tip: You can also use the menus in the main menu bar to manage objects. For more information, see *Managing objects using manager menus*.

Additional information

• *Managing objects using manager menus*

Managing objects using manager menus

Like managers, pull-down menus from the main menu bar allow you to manage all the objects defined in the current model or session.

1. From the main menu bar, select one of the following:

   - To manage objects associated with a module, select the manager menu items in the appropriate menu in the main menu bar. For example, to edit a material in the Property module you would select **Material**->**Edit**->***material of your choice*** from the main menu bar.
   - To manage objects associated with a toolset, select the appropriate manager menu items in the **Tools** menu. For example, to delete a set, you would select **Tools**->**Set**->**Delete**->***set of your choice*** from the main menu bar.
   - To manage all the models defined in the current model database, select the manager menu items in the **Model** menu in the main menu bar. For example, to copy a model, you would select **Model**->**Copy**->***model of your choice*** from the main menu bar.

   In most cases a dialog box appears; for example, when you rename an object, a dialog box appears that asks for the new name of the object.

2. If a dialog box appears, provide the requested information and click **OK**.



Tip: You can also use manager dialog boxes to manage objects. For more information, see *Managing objects using manager dialog boxes*.

Additional information

• *Managing objects using manager dialog boxes*

Copying step-dependent objects using manager dialog boxes

You use the **Copy** button in manager dialog boxes to copy a step-dependent object such as a load, boundary condition, interaction, predefined field, output request, or adaptive mesh constraint. You can copy an object from any step to any other valid step.

1. Open the appropriate manager, as described in *Managing objects using manager dialog boxes*.
2. Select the object you want to copy by clicking in the corresponding table cell.
3. Click **Copy**.

   The corresponding **Copy *Object*** dialog box appears.
4. Enter a name for the new object.
5. Choose the destination step from the **Step** list. By default, the object is copied into the step in which it was created.
6. Click **OK**.

   The object is copied to the selected destination step.

If the destination step type is different from the type of step in which the object was originally created, Abaqus/CAE may need to modify the object to make it compatible with the new step. For example, if a load was created in a static (general) step and you copy it into a steady-state dynamics (linear perturbation) step, Abaqus/CAE will add an imaginary component to the load value. Conversely, if a load from a steady-state dynamics step is copied to a static step, the imaginary component will be removed.

In addition, when you copy an object to a step other than the step in which it was created, any modifications made to the original object in one of its propagated states will be ignored in the copy.

Abaqus/CAE prevents you from copying an object into a step type where the object type is invalid. Abaqus/CAE also prevents you from copying a suppressed object or copying to a suppressed step.

Changing the status of an object in a step

Step-dependent managers contain buttons that you can use, under certain circumstances, to alter the status of an object in a particular step. These buttons are labeled **Move Left**, **Move Right**, **Activate**, and **Deactivate**.

Whether or not you can change the status of an object in a step depends on the step's procedure and the object's status in the step. The manager allows you to make only valid changes to the history of an object. If the operation of one of the buttons would cause an invalid change in status, that button becomes unavailable. For more information, see *Modifying the history of a step-dependent object*.

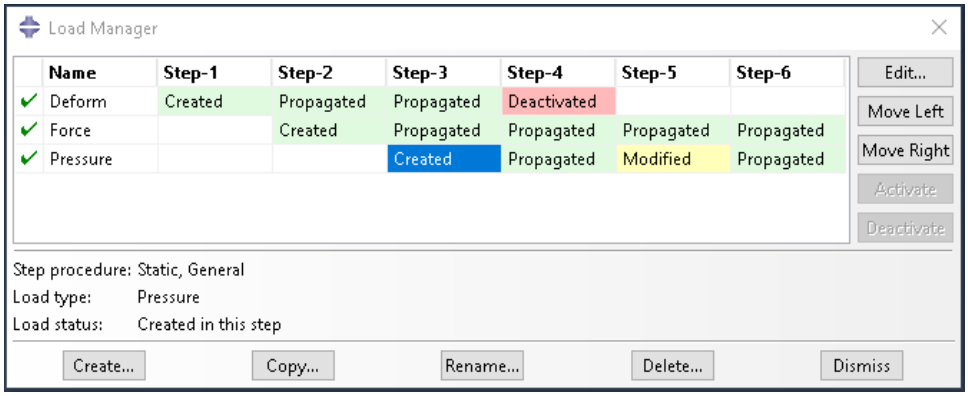
The following list describes techniques for manipulating the status of a step-dependent object:

To select the status that you want to change:

Click in the cell that is located in the row of the object of interest and in the column of the step of interest.

The status of the object in that step becomes highlighted, and, in most cases, some or all of the buttons on the right side of the dialog box become available. The availability of the buttons depends on the status of the object in the current step, in the preceding step, and in the following step.

For example, the `Created` status of **Pressure** in `Step-3` is selected in the figure below:

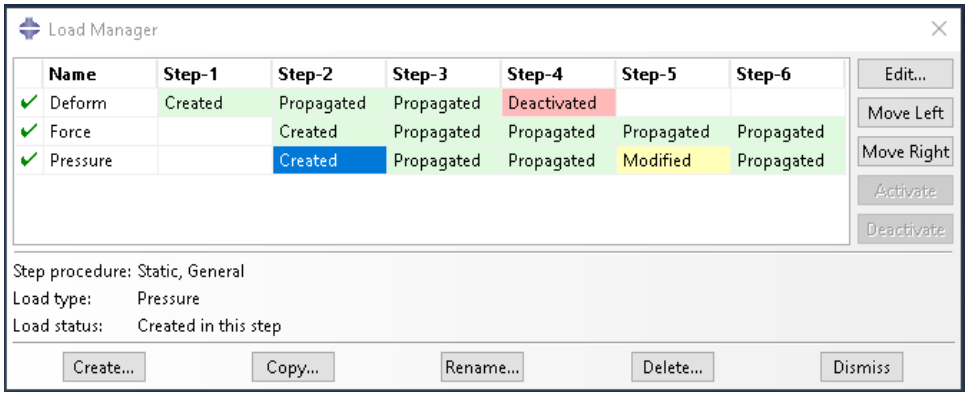


Use the buttons that become available to manipulate the status of the object in the step that you have chosen, as described below.

To move the status in the selected step to the preceding step:

Click **Move Left** to move the highlighted status from the selected to the preceding step.

For example, the `Created` status of **Pressure** in `Step-3` is selected in the history shown above. If you clicked **Move Left**, the history would change as shown below:

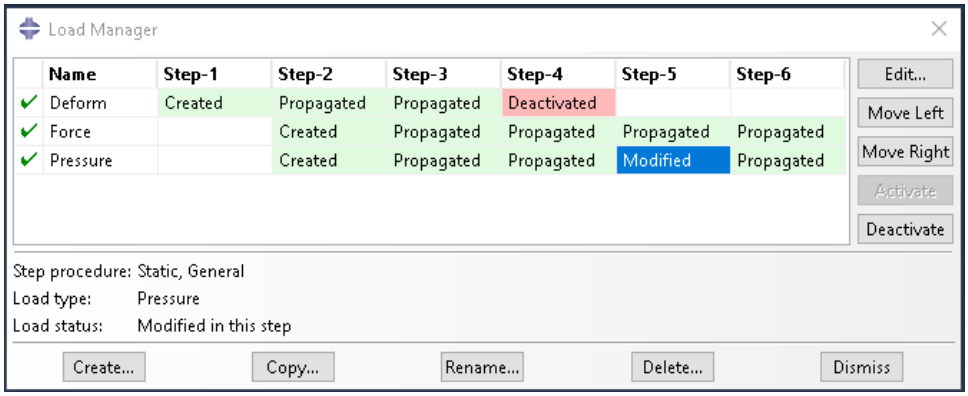


The `Created` status of **Pressure** moves to `Step-2` and is replaced by `Propagated` in `Step-3`.

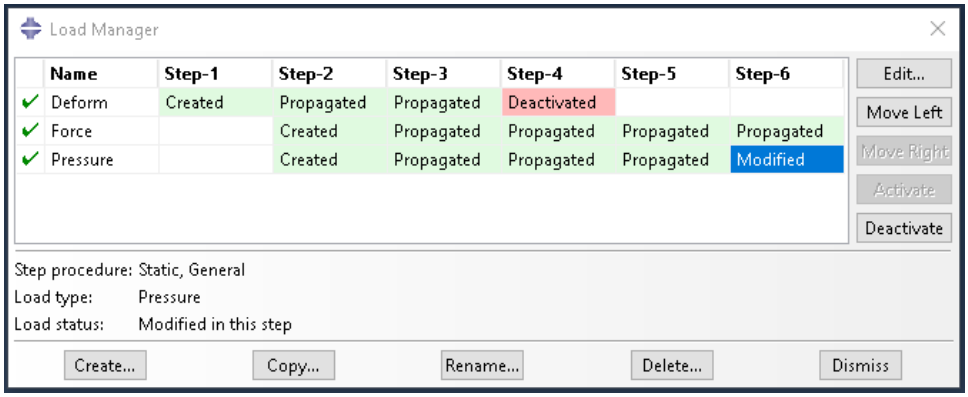
To move the status in the selected step to the following step:

Click **Move Right** to move the highlighted status from the selected step to the following step.

In the history shown below, for example, the `Modified` status of **Pressure** in `Step-5` is selected.



If you clicked **Move Right**, the history would change as shown below:

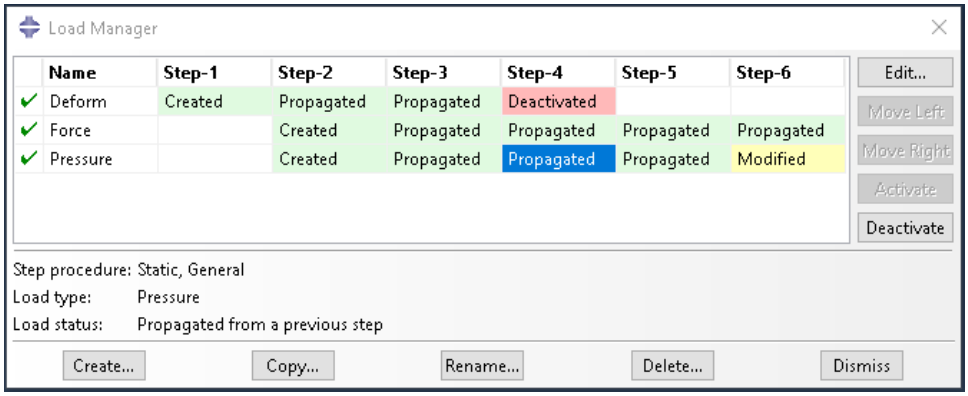


The `Modified` status of **Pressure** moves to `Step-6` (indicating that the modifications to **Pressure** become effective in `Step-6`), and `Modified` is replaced by `Propagated` in `Step-5`.

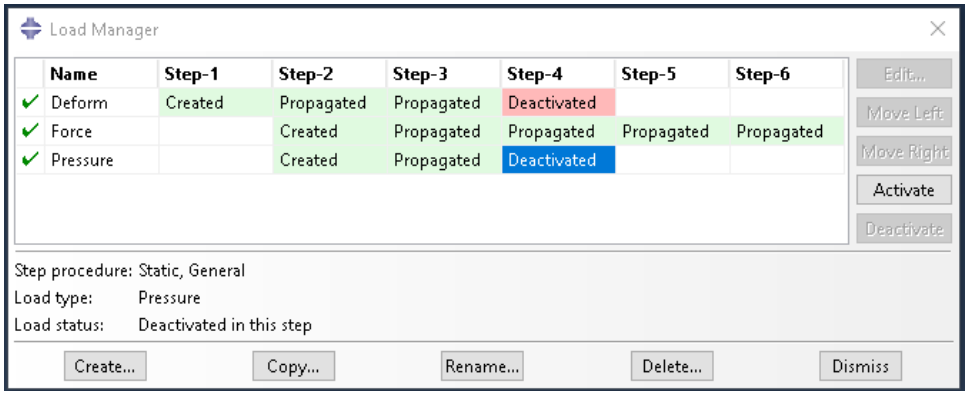
To deactivate the object in the selected step:

Click **Deactivate** to deactivate the object in the selected step.

In the history shown below, for example, the `Propagated` status of **Pressure** in `Step-4` is selected.



If you clicked **Deactivate**, the history would change as shown below:



The `Propagated` status of **Pressure** in `Step-4` changes to `Deactivated`, and the pressure is inactive in all subsequent steps.

Note:

You cannot deactivate predefined fields using the **Predefined Field Manager**; you must select **Reset to initial** in the predefined field editor (for example, see *Defining a temperature field*).

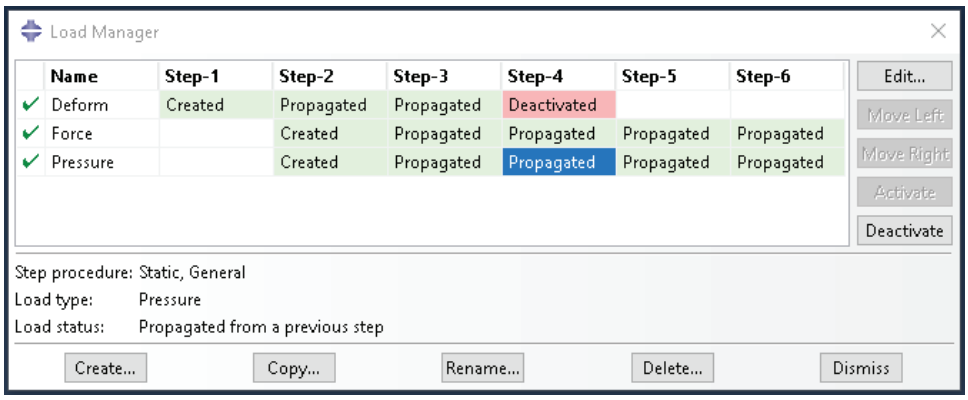
Warning:

If you deactivate an object in a step in which its status is `Modified`, the modifications to the object are lost. If you later reactivate the object in that step, the original, unmodified version of the object becomes active in that step and in all subsequent steps.

To reactivate the object in the selected step:

Click **Activate** to reactivate the object in the selected step.

In the history shown above, for example, the `Deactivated` status of **Pressure** in `Step-4` is selected. If you clicked **Activate**, the history would change as shown below:



The `Deactivated` status of **Pressure** changes to `Propagated` in `Step-4` and in any following steps.

Note:

The **Activate** button is available only in the step in which an object is first deactivated.

You can use the Model Tree to view the status of a step-dependent object, to edit the object, and to deactivate and reactivate the object. However, you must use the step-dependent manager to modify the history of an object by moving it right or left in the sequence of steps. For more information, see *The Model Tree*.

Additional information

- *What are step-dependent managers?*

Editing step-dependent objects

You can use either menus, managers, or the Model Tree to edit step-dependent objects in a particular step.
You cannot edit suppressed objects; you must resume the object before editing. For more information, see *Suppressing and resuming objects*. (For information about the status of modified objects, see *Understanding modified step-dependent objects*.)

Additional information

- *What are step-dependent managers?*
- *Suppressing and resuming objects*
- *Understanding modified step-dependent objects*
- *Understanding symbols that represent prescribed conditions*
- *Editing the region to which a prescribed condition is applied*

Edit step-dependent objects using menus

1. From the main menu bar, select **Edit->*object of your choice*** from the appropriate menu. For example, if you want to edit a load in the Load module, select **Load->Edit->*load of your choice***.

   The appropriate editor appears. The region to which the object is applied becomes highlighted in the current viewport.

2. In the editor, modify the object definition as desired, and click **OK**.

Edit step-dependent objects using managers

1. In the manager, locate the cell of interest. The cell is located in the row of the object that you want to modify and in the column of the step of interest.
2. In the manager, double-click the cell.

   Note:

   Alternatively, you can click the cell of interest and then click **Edit**.

   The appropriate editor appears. Abaqus/CAE highlights the region to which the object is applied.

3. In the editor, modify the object definition as desired, and click **OK**.

Edit step-dependent objects using the Model Tree

1. In the Model Tree select the object of interest.
2. Click mouse button 3 on the object and select **Edit** from the menu that appears

   The appropriate editor appears. Abaqus/CAE highlights the region to which the object is applied.

3. In the editor, modify the object definition as desired, and click **OK**.

Working with the Model Tree and the Results Tree

The Model Tree and Results Tree are convenient tools for navigating and managing your models and analysis results.

You can use the Model Tree to view a model and the items that it contains, and you can use the Results Tree to display analysis results from output databases as well as session-specific data such as *X*–*Y* plots. Both trees provides shortcuts to much of the functionality of the main menu bar, the module toolboxes, and the various managers. This section describes both the Model Tree and Results Tree.

In this section:

- *The Model Tree*
- *The Results Tree*
- *Using popup menus in the Model Tree and the Results Tree*
- *Changing the view of the model*

The Model Tree

The Model Tree provides a visual description of the hierarchy of items in a model.
For example, *Figure 1* shows the appearance of the Model Tree after completing the tutorial for a cantilever beam in *Creating and Analyzing a Simple Model in Abaqus/CAE*.

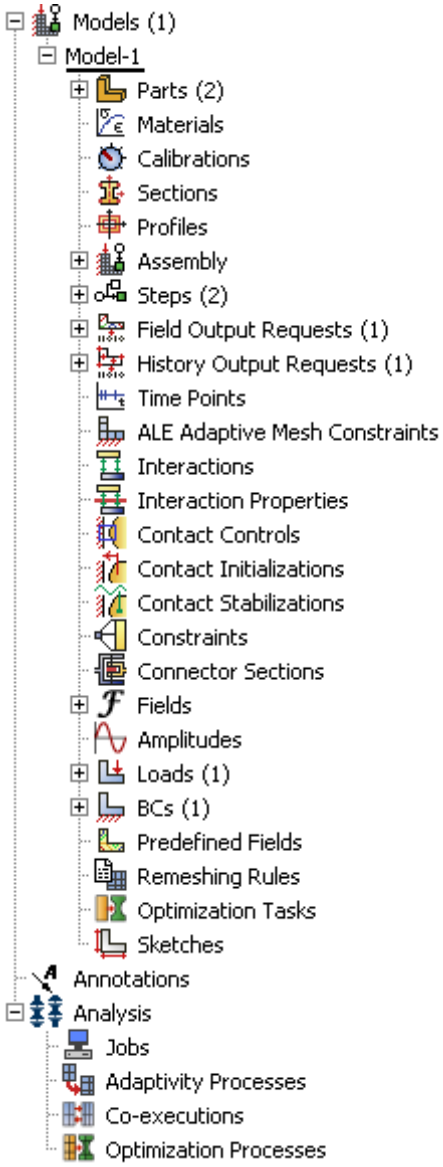
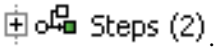


Figure 1: The Model Tree after completing the cantilever beam tutorial.

The Model Tree shares the left side of the Abaqus/CAE interface with the Results Tree and, in the Property module only, a material library. You can click the **Model**, **Results**, or **Material Library** tab to toggle the display between the Model Tree, the Results Tree, and a material library. See *The Results Tree*, and *Using material libraries*, respectively, for more information about the Results Tree and material libraries. In addition, the tip button at the top of the Model Tree provides a quick summary of the functionality of the Model Tree and Results Tree along with a summary of the keyboard shortcuts described at the end of this section.

A complete Abaqus/CAE model contains all of the information required to perform an analysis; for example, all of the parts, materials, steps, and loads and the meshed representation of the assembly. A model also contains the jobs that are submitted to the Abaqus analysis products. For more information, see *What does an Abaqus/CAE model contain?*. All of these items are represented in the Model Tree.

Items in the Model Tree are represented by small icons; for example, the **Steps** icon, 

. In addition, parentheses next to an item indicate that the item is a container, and the number in the parentheses indicates the number of items in the container. You can click on the "plus" and "minus" signs in the Model Tree to expand and collapse a container. The right and left arrow keys perform the same operation.

For example, the **Steps** container contains all the steps in your model. In the example shown in *Figure 1* expanding the **Steps** container reveals that the model contains two steps—the **Initial** step and the **BeamLoad** step. Expanding the **BeamLoad** step, as shown in *Figure 2*, reveals that the step has four containers, each of which contains a single item—**Field Output Requests**, **History Output Requests**, **Loads**, and **BCs**.

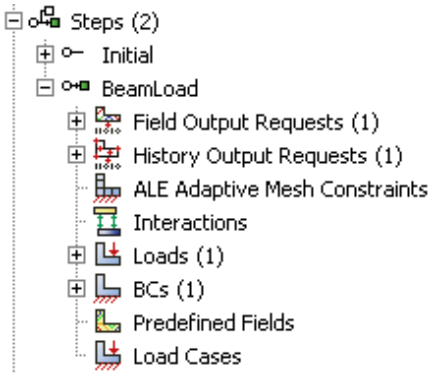


Figure 2: Containers in the BeamLoad step.

In addition, the step contains four empty containers—**ALE Adaptive Mesh Constraints**, **Interactions**, **Predefined Fields**, and **Load Cases**. You cannot delete an empty container from the Model Tree, although you can hide empty containers from view (see *Changing the view of the model*). Finally, expanding the **Loads** container, as shown in *Figure 3*, reveals a single load called **Pressure** that was created in this step.

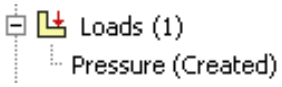


Figure 3: The load in the Loads container.

The arrangement of the containers and items in the Model Tree reflects the order in which you are likely to create your model. A similar logic governs the order of modules in the module menu—you create parts before you create the assembly, and you create steps before you create loads. This arrangement is fixed—you cannot move items in the Model Tree. For more information, see *What is a module?*.

Abaqus/CAE underlines the current objects in the Model Tree and displays them in the context bar. The model you are working on is a current object. The current part or the current step is also a current object. When you select an item in the Model Tree, Abaqus/CAE highlights that item in the current viewport if the selected item belongs to the current objects. For example, if you select a load, Abaqus/CAE highlights the load in the current viewport if it was applied in the current step of the current model. Containers are not highlighted.

You can select multiple items in the Model Tree, and Abaqus/CAE highlights each of those items if they belong to the current objects. For example, you can select an interaction and a load in the current step of the current model, and Abaqus/CAE highlights both the interaction and the load in the assembly. As you move the cursor over an item, the Model Tree displays some information about the item, as shown in *Figure 4*. In most cases the same information is available from the item's manager.

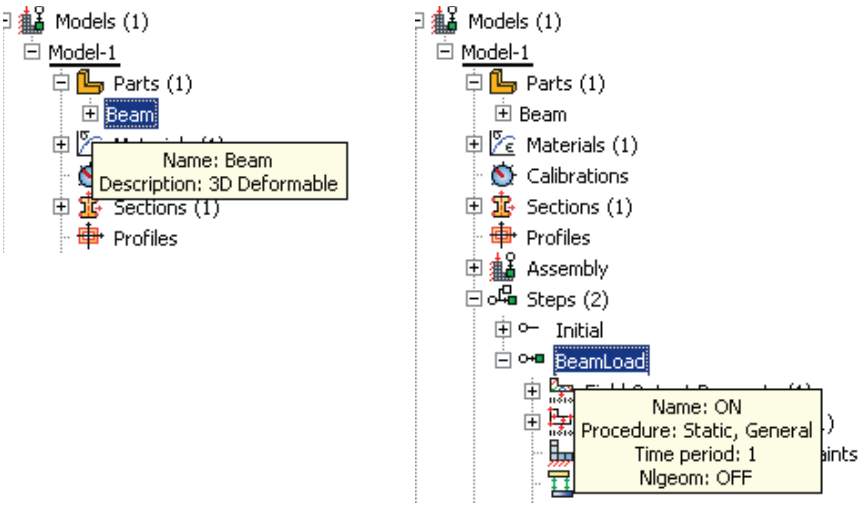


Figure 4: The Model Tree displays information about the item under the cursor.

Pressing an alphabetic key (a–z) when the cursor is in the Model Tree selects the first item in the tree with a name beginning with that character. Pressing subsequent keys continues to match characters in an item's name. *Table 1* describes all of the keyboard shortcuts that are available for navigation in the Model Tree; you can use these shortcuts to navigate in the Results Tree as well.

Table 1: Keyboard shortcuts in the Model Tree and Results Tree.

Keyboard shortcut	Action
[Home]	Go to top of Model Tree or Results Tree
[End]	Go to bottom of Model Tree or Results Tree
Up arrow	Move up one item
Down arrow	Move down one item
Right arrow	Expand branch or move down one item
Left arrow	Collapse branch or move up one item
[Del]	Delete item
[F2]	Apply a *filter* to a container

The Model Tree provides most of the functionality of the main menu bar and the module managers. For example, if you double-click on the **Parts** container, you can create a new part (the equivalent of selecting **Part->Create** from the main menu bar). If you double-click on a part's feature, you can edit the feature (the equivalent of selecting **Feature->Edit** from the main menu bar).

You can drag the divider between the Model Tree and the canvas to change the width of the Model Tree. In addition, you can toggle off the display of the Model Tree by selecting **View->Show Model Tree** from the main menu bar. Pressing **[Ctrl] + T** has the same effect. To switch to the Results Tree, click the **Results** tab.

Step-dependent objects are objects that can be propagated between steps; for example, loads and interactions. For more information, see *What are step-dependent managers?*. Text next to a step-dependent object in the Model Tree, such as **(Created)** and **(Propagated)**, indicates the status of the object. You can use the Model Tree to change the status of a step-dependent object by clicking mouse button 3 on the object and selecting an action from the menu that appears. The actions correspond to those available in the step-dependent managers. For more information, see *Understanding modified step-dependent objects*.

You can use the Model Tree to suppress a feature, a constraint (in the Interaction module), a section assignment (in the Property module), or a step-dependent object by clicking mouse button 3 on the item and selecting **Suppress** from the menu that appears. A red "X" appears next to the item in the Model Tree to indicate that it is suppressed. You can resume the item by clicking mouse button 3 on the item and selecting **Resume**. Abaqus/CAE removes the red "X" from the Model Tree to indicate that the item is no longer suppressed. The same information is displayed in the managers. For more information, see *Suppressing and resuming objects*.

The Results Tree

The Results Tree provides a visual description of the output data available in your session, including all open output databases and session-specific data such as *X–Y* data and *X–Y* plots. In addition, the Results Tree enables you to navigate to viewable content in the current model database, such as the loads specified in one step of a particular model. This tool shares the left side of the Abaqus/CAE interface with the Model Tree and, in the Property module only, a material library. You can click the **Model**, **Results**, or **Material Library** tab to toggle the display between the Model Tree, the Results Tree, and a material library. (For more information on material libraries, see *Using material libraries*.) The Results Tree also uses all of the same keyboard and navigational shortcuts as the Model Tree; see *Table 1* for more information.

Figure 1 shows the appearance of the Results Tree after completing an analysis of the tutorial for the hinge model in *Using Additional Techniques to Create and Analyze a Model in Abaqus/CAE*. The **Output Databases** container displays all the output database files that are currently open in your session. In the example shown in *Figure 1* the **Output Databases** container is expanded and reveals that only one output database is open—the **PullHinge** output database.

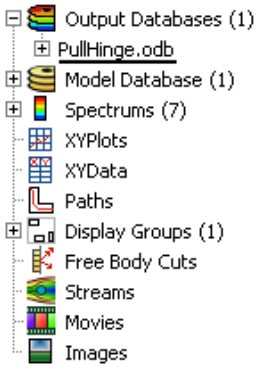


Figure 1: The Results Tree after completing an analysis of the hinge model tutorial.

Expanding **PullHinge**, as shown in *Figure 2*, reveals that this output database has the following containers: **History Output**, **Steps**, **Instances**, **Materials**, **Sections**, **Element Sets**, **Node Sets**, and **Surface Sets**. In addition, the output database contains two empty containers—**Session Coordinate Systems** and **ODB Coordinate Systems**. You cannot delete an empty container from the Results Tree, although you can hide empty containers from view (see *Changing the view of the model*).

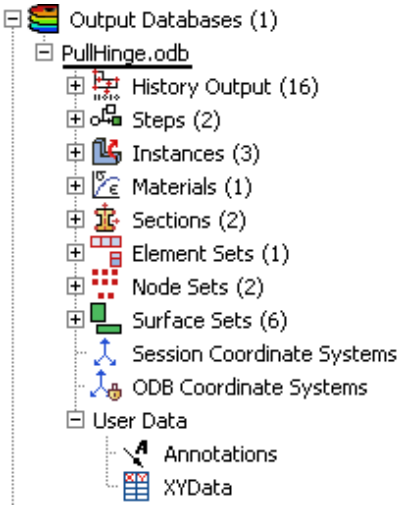


Figure 2: The containers in the PullHinge output database.

Expanding the **History Output** container, as shown in *Figure 3*, reveals the sixteen output variables for which history output was requested in this analysis. Each variable listing also describes the region for which history output was requested; in this example every history output request was made for the whole model. You can click any of the history output variables in the Results Tree to plot the selected variable in the current viewport.

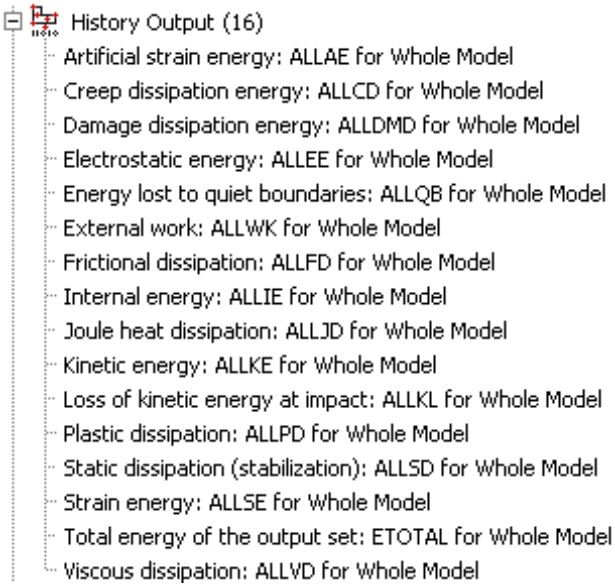


Figure 3: The History Output container of the Results Tree.

Each output database also includes a **Steps** container, which includes containers for each step in the output database and within the steps, every frame in the output database. You can use the Results Tree to display the model at any frame of the analysis, to activate or deactivate steps or frames in the analysis, or to display field output at the selected frame.

The **Model Database** container displays all the models in the current model database. You can expand each model to select the step with data that you want to investigate and to display or hide individual part instances. *Figure 4* shows the hinge model with its **Steps** and **Instances** containers expanded.

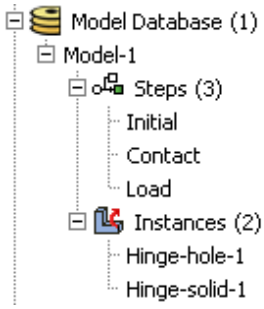


Figure 4: The Model Database container of the Results Tree.

The other containers in the Results Tree provide shortcuts to data that persist only during your session. By using these shortcuts, you can create and manage contour spectrums; create and edit *X–Y* data and display *X–Y* plots, create and manage paths and display groups; and upload and display background images and movies.

Using popup menus in the Model Tree and the Results Tree

Much of the power of the Model Tree and Results Tree comes from the popup menu that appears when you click mouse button 3 on an item. For example, *Figure 1* shows the effect of clicking mouse button 3 on the **Parts** container in the Model Tree.

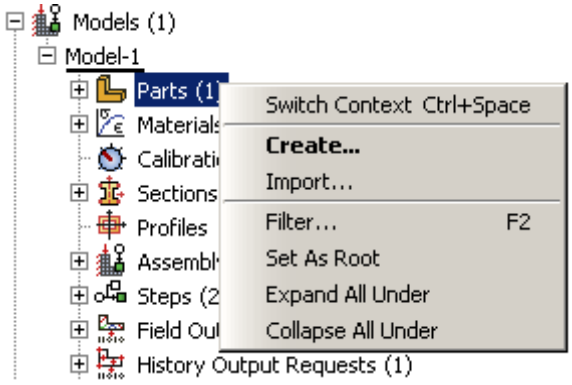


Figure 1: Clicking mouse button 3 on the Parts container in the Model Tree.

The **Create** menu item appears in bold font in *Figure 1* because it is the default action. Double-clicking an item or selecting an item and pressing **[Enter]** invokes the default action. In most cases if an item is a container, the default action is to create a new item in the container. Similarly, if an item is not a container, the default action is to edit the item. For example, if you double-click the **Parts** container, Abaqus/CAE displays the **Create Part** dialog box and allows you to create a new part; if you double-click one of the parts within the container, Abaqus/CAE displays the **Edit Part** dialog box and allows you to edit the part you selected.

Some of the commands in the popup menu appear with all items in the Model Tree and Results Tree; other commands appear only with specific items or with items in one of the two trees. For example, the **Switch Context** command appears with all items in the Model Tree and Results Tree, including containers. The following commands appear with all containers in the Model Tree and Results Tree:

Switch Context

If you select **Switch Context**, Abaqus/CAE makes the item current. In the Model Tree, where appropriate, Abaqus/CAE also switches to the module in which you can edit the item. For example, if you click mouse button 3 on the **Materials** container and select **Switch Context**, Abaqus/CAE switches to the Property module. For more information, see *What is a module?*. In the Results Tree you can also switch the context to an item in another output database.

Selecting a container and pressing **[Ctrl][Space]** have the same effect as selecting **Switch Context** from either tree.

Filter

When you select **Filter**, Abaqus/CAE prompts you for a string of characters next to the container's name. After you press **[Enter]**, Abaqus/CAE filters the contents of the container and displays only those items that match the specified character string. The filter is case-sensitive. For details on valid filtering syntax, click the tip button



at the top of the Model Tree or Results Tree. *Figure 2* shows the effect of filtering a container.

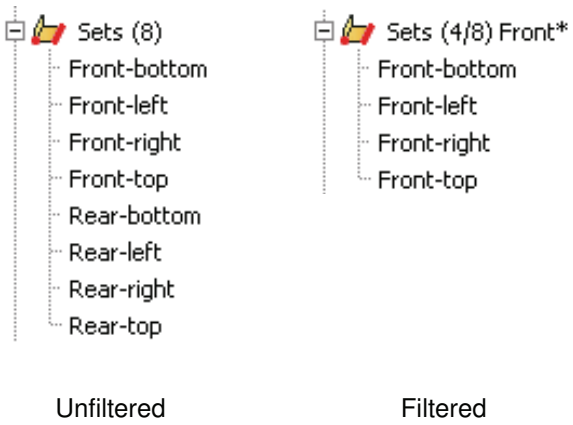


Figure 2: Filtering a container in the Model Tree.

Items that are hidden by a filter cannot be manipulated from the Model Tree or Results Tree; however, these items are not removed from the model or output database. When a filter is in effect, the numbers in parentheses next to the container name indicate the number of visible items in the container followed by the total number of items (visible and hidden) in the container (see *Figure 2*). The filter string appears to the right of these numbers. To remove a filter, select **Filter** for the appropriate container, delete the filter string, and press **[Enter]**.

Filters can be applied only to individual containers and persist only during your session. Selecting a container and pressing **[F2]** have the same effect as selecting **Filter** from either tree.

Set As Root

If you select **Set As Root**, Abaqus/CAE moves the container to the pull-down menu above the Model Tree or Results Tree and displays everything under the selected container. For more information, see *Changing the view of the model*.

Expand All Under

If you select **Expand All Under**, Abaqus/CAE expands all of the containers and items inside the selected container.

Collapse All Under

If you select **Collapse All Under**, Abaqus/CAE collapses all of the containers and items inside the selected container.

Group Children

When a container includes more than 30 items, Abaqus/CAE automatically groups the items into sets of 30. If you toggle off the **Group Children** option, Abaqus/CAE removes the groupings and lists all of the items on the same level in the container.

Selecting a container and pressing **[Ctrl][G]** have the same effect as selecting **Group Children** from either tree.

Many popup menu commands appear only with specific items. In the Model Tree these commands mirror the actions that you can perform with that item's manager; for example, create, edit, delete, rename, suppress, and resume. In the Results Tree some popup menus provide Boolean operators that enable you to control the display of items in the current viewport. These Boolean operators are the same five commands that are available for controlling display groups: replace, add, remove, intersect, and either. See *Understanding display group Boolean operations*, for more information.

Some menu commands are specific to one container in the Model Tree or Results Tree; for example, clicking mouse button 3 on a step allows you to toggle the **Nlgeom** setting, clicking mouse button 3 on a job allows you to submit the job for analysis, and clicking mouse button 3 on a history output variable allows you to add another variable to the existing plot. When you become familiar with the Model Tree and Results Tree, you will find that you can quickly perform most of the actions that are found in the main menu bar, the module toolboxes, and the various managers.

Changing the view of the model

If you select **Set As Root** from a container's popup menu, Abaqus/CAE displays everything under the selected container in the Model Tree or Results Tree and displays the name of the container in the menu above the tree. This option is useful if, for example, you have a complex model or output database and correspondingly complex Model Tree or Results Tree. You can use **Set As Root** to simplify the Model Tree or Results Tree by displaying only the portion that you are working on. For example, *Figure 1* shows a view of the Model Tree on the left and contrasts it with the effect of setting the **Materials** container as the root.

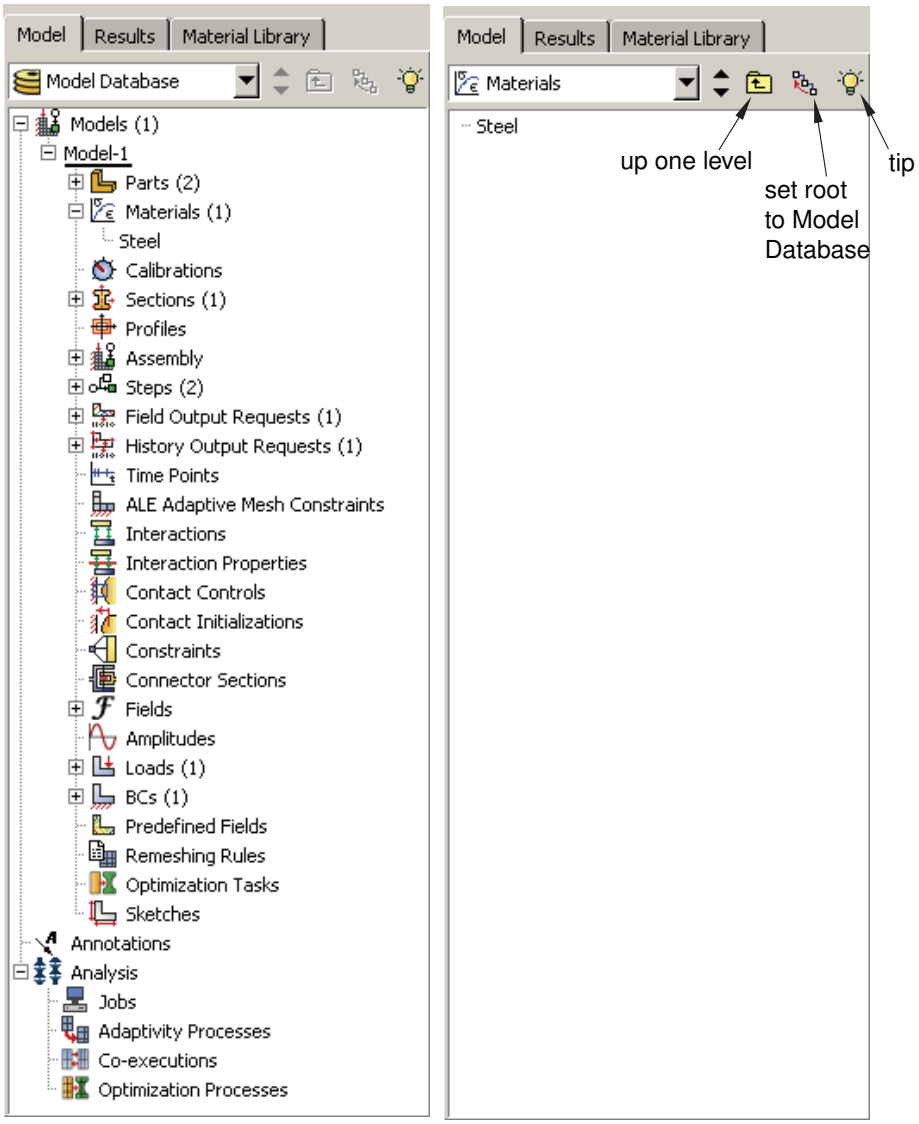


Figure 1: The effect of Set As Root on the Model Tree.

When you change the default root container, you can use the menu above the Model Tree and Results Tree to move up through its levels. In addition, Abaqus/CAE activates two icons above the Model Tree or Results Tree, as shown in *Figure 1*: the **Set Root to Model Database** icon in the Model Tree, the **Set Root to Session Data** icon in the Results Tree, and the **Up One Level** icon in both trees.

- The **Set Root to Model Database** icon  returns the Model Tree to the default view that shows **Model Database** at the top of the tree. The **Set Root to Session Data** icon  returns the Results Tree to the default view that shows **Session Data** at the top of the tree.
- The **Up One Level** icon  moves the root of the Model Tree or Results Tree up one level; for example, from the **Materials** container up one level to the **Beam** model that contains the **Materials** container.

If you click mouse button 3 on the background of the Model Tree or Results Tree, Abaqus/CAE displays a popup menu with the following options:

Show Empty Containers

By default, Abaqus/CAE displays all of the containers in the Model Tree and Results Tree, whether or not they have items in them. By turning off the **Show Empty Containers** option, you can suppress the display of containers without any items in them. If you perform an action in Abaqus/CAE that adds an item to a previously empty container (for example, creating an interaction using the Interaction module toolbox), the container and item will reappear in the Model Tree or Results Tree. For a container to be suppressed from view, it must be completely empty; even if all items in a container are hidden because of a filter (see *Using popup menus in the Model Tree and the Results Tree*), that container is not suppressed by the **Show Empty Containers** option.

The state of the **Show Empty Containers** option persists between Abaqus/CAE sessions.

Expand All

If you select **Expand All**, Abaqus/CAE expands all of the containers and items in the Model Tree or Results Tree.

Collapse All

If you select **Collapse All**, Abaqus/CAE collapses all of the containers in the Model Tree or Results tree, leaving only top-level containers and items visible.

Set Root to Displayed Object

If you select **Set Root to Displayed Object**, the container corresponding to the part visible in the current viewport becomes the root of the Model Tree (as described above). If an assembly is visible in the current viewport, the **Assembly** container for the appropriate model becomes the root of the Model Tree.

This option is available only in the Model Tree.

Set Root to Model Database

If you select **Set Root to Model Database**, Abaqus/CAE returns the Model Tree to the default view that shows **Model Database** at the top of the tree. This option has the same effect as the **Set Root to Model Database** icon .

This option is available only in the Model Tree.

Set Root to Session Data

If you select **Set Root to Session Data**, Abaqus/CAE returns the Results Tree to the default view that shows **Session Data** at the top of the tree. This option has the same effect as the **Set Root to Session Data** icon  . This option is available only in the Results Tree.

Understanding Abaqus/CAE GUI settings

GUI settings are always saved automatically to a binary file in your home directory called `abaqus_2025.gpr` when you exit Abaqus/CAE. For more information, see *Working with abaqus_2025.gpr files*.

These GUI settings include the following:

- The size and location of the main window.
- The size and location of a particular dialog box; for example, the **Open Database** and **Create Part** dialog boxes.
- The location, orientation, and visibility of individual toolbars.
- Custom toolbars.
- Customized keyboard shortcuts.
- The size of the message area and command line interface.
- Whether the Model Tree and Results Tree are displayed. The width of the tree area is also stored.
- Bookmarks to directories that you created when opening a file.

You cannot edit the `abaqus_2025.gpr` file; however, you can delete it to restore the default GUI and display options settings.



Warning:

Deleting the `abaqus_2025.gpr` file resets all of the GUI settings listed above. You cannot restore the settings from a deleted `abaqus_2025.gpr` file except by recreating them manually in Abaqus/CAE.

Managing viewports on the canvas

The canvas can be thought of as an infinite screen or bulletin board on which you post viewports; you can imagine the canvas extending beyond the main window and your monitor.

The visible portion of the canvas is called the drawing area, and you can increase its size by increasing the size of the main window. You can display the drawing area full screen using the **View** menu; you can also press **[F11]** to toggle between full screen mode and normal mode.

You can position viewports anywhere on the canvas, and you can drag them outside the drawing area. When viewports are positioned outside the drawing area, you can cascade or tile the viewports to bring them back into view. Viewports are not part of a model and are not saved between sessions.

This chapter explains how to create and manipulate viewports, text annotations, and arrow annotations.

In this section:

- *Understanding viewports*
- *Manipulating viewports and viewport annotations*
- *Displaying the drawing area in full screen mode*
- *Working with viewports*
- *Working with viewport arrow and text annotations*
- *Linking viewports for view manipulation*
- *Working with background images and movies in viewports*

Understanding viewports

Viewports are areas on the canvas where you can display models or analysis results. You can add arrow and text annotations to draw attention to or explain features within a viewport. You can create and manipulate viewports, text, and arrows using the **Viewport** menu.

In this section:

- *What is a viewport?*
- *What are arrow and text annotations?*

What is a viewport?

While the canvas can be thought of as an infinite screen or bulletin board, viewports are simply display areas posted onto that screen on which you can display models or analysis results. You can have many viewports on the canvas. A viewport is similar to other windows on your workstation in that it can be moved, resized, minimized, and maximized; and it can overlap other viewports on the canvas. For more information, see *Working with viewports*.

You can easily create and delete viewports and control their size, position, and appearance. *Figure 1* illustrates how you might use several viewports to view the results from your analysis.

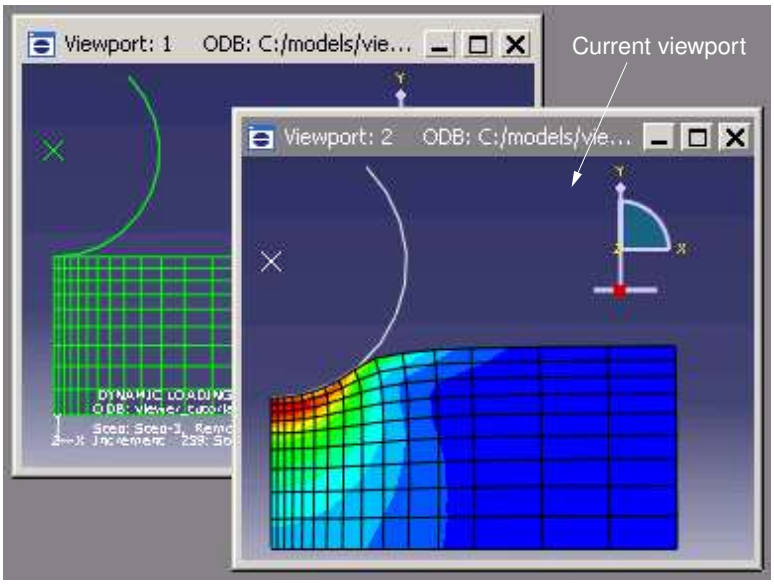


Figure 1: Working with multiple viewports.

The *view manipulation tools*, such as zoom and rotate, operate on the viewport that contains the cursor. Other operations interact with the current viewport or with all viewports on the canvas.

The current viewport

To change the contents of a viewport, you must first designate the desired viewport as current. The current viewport is indicated by a dark gray title bar. All work takes place within the current viewport. To choose another viewport as the current viewport, click on the border or title bar. The selected viewport moves in front of other viewports on the canvas, and the title bar color changes to blue. The title bar reverts to dark gray when you select an Abaqus/CAE tool or menu.



Note:

On Windows platforms you can customize the colors used by Abaqus/CAE. For more information, see *Common customizations on Windows platforms*.

All viewports are associated with a certain model and module. When you create a new model or open an existing model or output database, that model becomes associated with the current viewport. You can create different viewports and associate each one with a different model, so designating each viewport as current results in

switching between the associated models. Similarly, you can work in multiple modules simultaneously by designating a new viewport as current before entering a different module.

Additional information

• *Components of the main window*

What are arrow and text annotations?

Arrow and text annotations are arrows and text strings that you create in a viewport to enhance the appearance and clarity of displayed models or results. You can create arrow annotations and text annotations independently, and you can create arrows and text together to automatically position text at the end of an arrow. The positions of annotations in a viewport are controlled by anchor points. You define each anchor point based on viewport geometry or model coordinates; the method you choose determines how Abaqus/CAE moves the annotations. If you manipulate the viewport, Abaqus/CAE repositions any annotations anchored to viewport geometry; similarly, if you manipulate the model, Abaqus/CAE moves any annotations anchored to the model. *Figure 1* shows the use of arrows and text to describe details of a model.

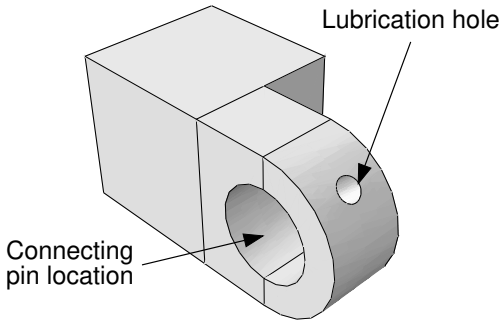


Figure 1: Arrow and text annotations.

Annotation editing operations require you to first select one or more annotations. Use the **Edit Annotations** tool from the **Viewport** toolbar to select arrow or text annotations from the current viewport. Abaqus/CAE highlights selected arrow or text annotations along with their anchor points, as shown in *Figure 2*.

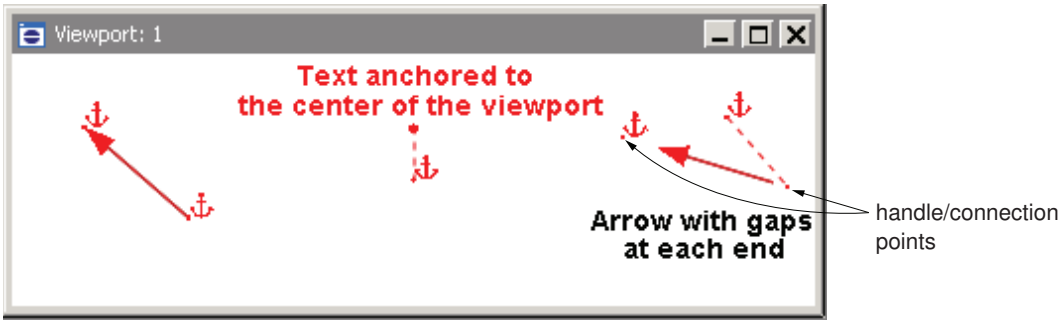


Figure 2: Selected annotations: an arrow with no offsets, text with an offset, and an arrow with gaps at both ends and an offset between the tail and its anchor point.

Anchor points are shown as a small dot with an anchor symbol placed nearby. Dashed lines indicate an offset between the anchor point and the annotation; circular “handles” are the anchor connection points—if there is no offset, the connection point and anchor point are the same.

Arrow annotations have two anchor points (you can use the same coordinates for both points). You can add a gap between the arrow ends and the connection points. Adding a gap is comparable to leaving a space between dimension lines and object lines in the Sketcher or in a CAD drawing; it can increase the clarity of your annotation. Text annotations have a single anchor point. You can change the offsets by dragging the connection points, or the entire annotation, in the viewport.

Do not confuse the viewport annotations that you can create with the viewport annotations generated by Abaqus/CAE. The generated viewport annotations include the view orientation triad; the 3D compass; and, in the Visualization module, the legend, the title block, and the state block. You can modify some display aspects of the generated annotations,

but you cannot modify their contents. For more information, see *Customizing viewport annotations*. In contrast, you have full control of all attributes related to arrow and text annotations including their colors, line styles, line thicknesses, arrowheads, fonts, anchor points, and any offsets between the anchor points and the annotations.

Abaqus/CAE saves arrow and text annotations in model and output databases; however, viewports are not saved. As a result, the arrow and text annotations in a database are not associated with a viewport. When you subsequently open a database that contains annotations, you must use the **Annotation Manager** to display a selected annotations in the current viewport. The **Annotation Manager** also allows you to copy annotations from a model database to an output database and vice versa. You cannot create annotations in a model database from the Visualization module; open the model database in a different module if you want to create annotations for it.

Additional information

- *Components of the main window*
- *Working with viewports*
- *Working with viewport arrow and text annotations*

Manipulating viewports and viewport annotations

This section explains how to manipulate viewports and viewport annotations using the options provided in the **Viewport** menu, the **Viewport** toolbar, and the **Annotation Manager**.

In this section:

- *Managing viewports and viewport annotations from the main menu bar*
- *Managing viewports and viewport annotations from the Viewport toolbar*
- *The Annotation Manager*

Managing viewports and viewport annotations from the main menu bar

Use the **Viewport** menu, located on the main menu bar, to create, delete, modify, link, or rearrange viewports and to create or edit viewport annotations—both those that you create and those generated by Abaqus/CAE. If you prefer, you can select **View->Toolbars->Viewport** from the main menu bar to display a toolbar containing most of the functionality of the items in the **Viewport** menu.

The **Viewport** menu and toolbar allow you to do the following:

- *Create* a viewport.
- *Edit* options for generated viewport annotations (triad, legend, title block, and state block).
- Create an *arrow* annotation.
- Create a *text* annotation.
- Create a *combined arrow and text* annotation.
- Edit *arrow* and *text* annotations.
- Open the **Annotation Manager** to manipulate arrow and text annotations.

  Note:

  The **Annotation Manager** provides several unique management functions; for more information, see *The Annotation Manager*.

- Open the **Viewport Annotation Options** to show or hide all annotations and to manipulate the viewport annotations generated by Abaqus/CAE.
- *Link* viewports.

In addition, the **Viewport** menu lists all the viewports in the session and allows you to delete the current viewport.

Additional information

• *Managing viewports on the canvas*

Managing viewports and viewport annotations from the Viewport toolbar

To display the **Viewport** toolbar, + select **View->Toolbars->Viewport** from the main menu bar. *Figure 1* describes the tools available from the **Viewport** toolbar.

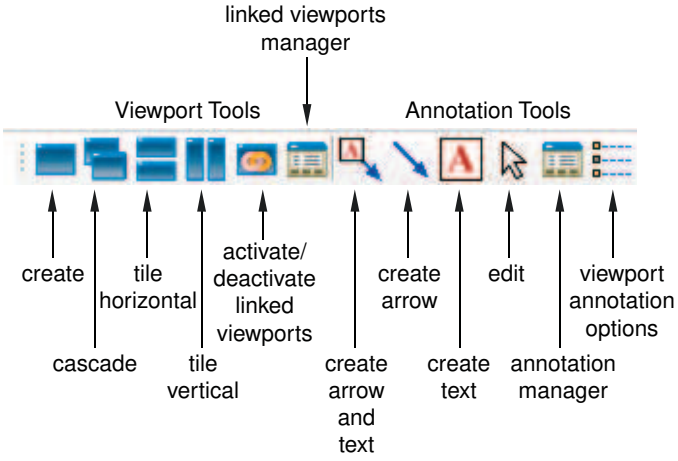


Figure 1: The Viewport toolbar.

Additional information

• *Managing viewports on the canvas*

The Annotation Manager

The **Annotation Manager** is similar to other manager dialog boxes in Abaqus/CAE. It allows you to do the following:

- Create *arrow*, *text*, or *combined arrow and text* annotations.
- Edit an *arrow* or *text* annotation.
- Copy or rename an annotation.
- Delete annotations.

In addition, the **Annotation Manager** allows you to perform the following tasks that are *not* available from the **Viewport** menu or toolbar:

- Select the source—model database (**MDB**) or output database (**ODB**)—of annotations to manage.
- Plot model database or output database annotations in the current viewport.
- Hide model database or output database annotations in the current viewport.
- Copy annotations from a model database to an output database and vice versa.
- Highlight annotations in the viewport.
- Rearrange the order of arrow and text annotations in the list.

You can display the **Annotation Manager** by selecting **Viewport->Annotation Manager** from the main menu bar or  in the **Viewport** toolbar. The **Annotation Manager** is shown in *Figure 1*.

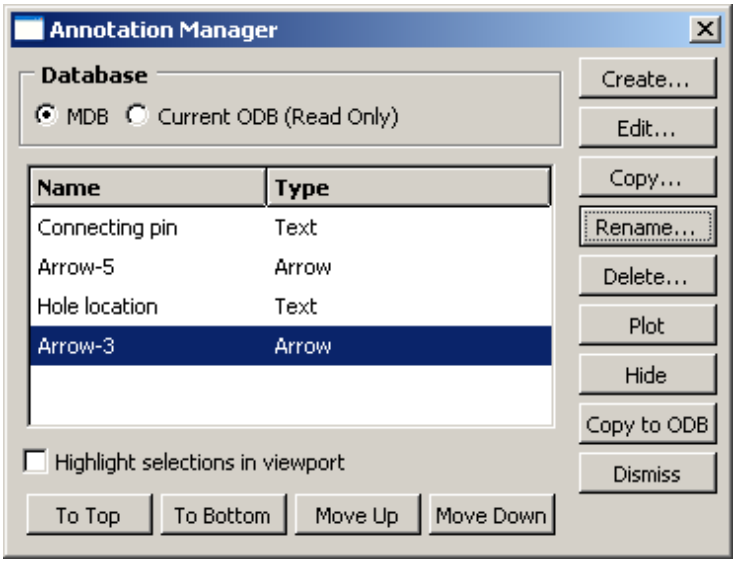


Figure 1: The Annotation Manager.

For detailed instructions on using the **Annotation Manager** to create, edit, and manipulate annotations, see the following sections :

- *Annotating viewports*
- *Editing arrow annotation attributes*
- *Editing text annotation attributes*
- *Plotting annotations in the current viewport*
- *Copying viewport annotations to another database*

- *Rearranging the annotation list order*

Additional information

- *Managing viewports on the canvas*

Displaying the drawing area in full screen mode

The drawing area is the visible portion of the canvas. You can display the drawing area in full screen mode. If you are in the Visualization module, the **Animation Controls** are available as a separate toolbar; otherwise, no menu bars or toolbars are accessible in full screen mode. To make a toolbar accessible in full screen mode, you can click and drag the toolbar's grip to “undock” the toolbar prior to switching modes.

1. From the main menu bar, select **View->Full Screen**.

   Tip: You can also press **[F11]** to toggle between full screen mode and normal mode.

   The drawing area enlarges to fill the entire screen.

2. Click the restore button in the title bar to return the drawing area to its previous size.

Additional information

• *Managing viewports on the canvas*

Working with viewports

This section explains how to create and manage viewports and how to modify their appearance.

In this section:

- *Creating new viewports*
- *Selecting viewports*
- *Moving viewports*
- *Resizing viewports*
- *Minimizing, maximizing, restoring, or deleting a viewport*
- *Cascading viewports*
- *Tiling viewports*

Creating new viewports

You can create new viewports at any time; there is no limit to the number of viewports or their position on the canvas.

From the main menu bar, select **Viewport->Create**.

Abaqus/CAE creates a new viewport in the drawing area. This viewport becomes the current viewport.

The new viewport size and position depend on the size of the current viewport and the drawing area. If the current viewport is maximized, Abaqus/CAE automatically maximizes the new viewport.

 **Tip:** You can also create a new viewport by clicking  in the **Viewport** toolbar.

Additional information

• *Managing viewports on the canvas*

Selecting viewports

Most of your interactions with the model—such as sketching a part, positioning a load, assembling part instances, generating a mesh, and customizing a plot state—take place in the current viewport. In addition, if you have multiple viewports displayed on the canvas, the current viewport indicates the model you are working on (the current model) and the module you are working in (the current module).

There are several ways that you can select a new viewport to be the current viewport:

- Click on the border or title bar of an existing viewport.
- Select an existing viewport from the list in the **Viewport** menu.
- Use **[Ctrl][Tab]**—or select **Next** or **Previous** from the **Viewport** menu—to cycle through all the viewports on the canvas.

  Note:

  On Linux platforms **[Ctrl][Tab]** is used to switch applications; **[Ctrl][F6]** is an alternative keyboard shortcut.

- Create a new viewport.

The current viewport has a dark gray title bar as shown in the following figure:



1. Move the cursor onto the border of the viewport.

   If the viewport is hidden, you may need to move other viewports to expose the one you want to select. For more information, see *Moving viewports*.

2. Click mouse button 1.

   The viewport becomes the current viewport; the viewport also becomes selected and its title bar changes to blue to indicate that it is the selected viewport. If you click on a tool or menu, the viewport remains current but the title bar reverts to dark gray.

   Note:

   On Windows platforms you can customize the colors used by Abaqus/CAE. For more information, see *Common customizations on Windows platforms*.

Additional information

- *Cascading viewports*
- *Tiling viewports*
- *Managing viewports on the canvas*

Moving viewports

You can move a selected viewport to any location on the canvas. This may be necessary to expose hidden viewports or simply to reduce clutter in the drawing area. To move a viewport, click anywhere on the viewport title bar and then drag it to the desired position.

1. Place the cursor anywhere on the viewport title bar.
2. Click mouse button 1, and + drag the cursor to the new location.

   The cursor changes to a four-headed arrow  . An outline of the viewport indicates its new position as you drag.
3. Release mouse button 1.

   The viewport moves to the new location, and it becomes the current viewport (if it was not previously).

Additional information

- *Selecting viewports*
- *Managing viewports on the canvas*

Resizing viewports

You can change the size and shape of a viewport by dragging its borders.

1. Place the cursor anywhere on the viewport border that you want to move.

   The cursor changes to a set of opposing arrows. The direction of the arrows is dependent on the relative position between the cursor and the viewport.

2. Click mouse button 1, and + drag the cursor to change the position of the borders.
3. Release mouse button 1.

   The viewport is displayed with the new size, and it becomes the current viewport (if it was not previously).

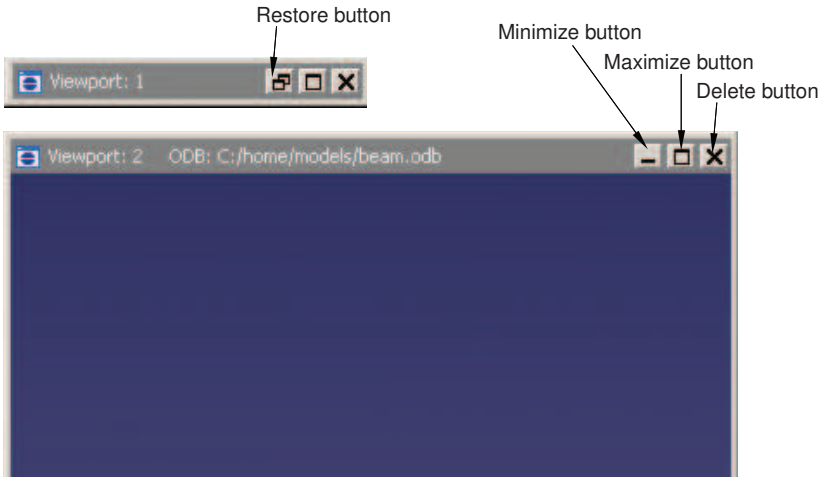
Additional information

- *Selecting viewports*
- *Managing viewports on the canvas*

Minimizing, maximizing, restoring, or deleting a viewport

Minimize, maximize, restore, and delete buttons are located in the top right corner of each viewport.

These buttons appear as shown in the following figure:



If necessary, **Tile** or **Cascade** viewports to reveal the minimize, maximize, restore, and delete buttons.

The minimize button

When you click the minimize button, the viewport reduces to an abbreviated title bar in a default location toward the lower left of the drawing area. The minimize button changes to a restore button that is retained, along with the maximize and delete buttons, in the title bar. If you move a minimized viewport, the new location will be saved as its minimized location for the duration of the Abaqus/CAE session.

The maximize button

When you click the maximize button, the viewport changes size and position to fill the canvas. The maximize button changes to a restore button and is moved, along with the minimize and delete buttons, to the right side of the main menu bar. The maximized viewport becomes the current viewport and is moved to the front of the canvas, hiding any other viewports in the drawing area. The information in the viewport title bar is displayed in the Abaqus/CAE title bar.

The delete button

When you click the delete button or select **Viewport->Delete Current** from the main menu bar, the viewport is deleted from the canvas. You cannot restore a deleted viewport. If there is only one viewport on the canvas, Abaqus/CAE does not allow you to delete it. If you have created additional viewports on the canvas, Abaqus/CAE deletes the current viewport and selects one of the other viewports to be the current viewport. You cannot control which viewport Abaqus/CAE selects to be current.

The restore button

The restore button replaces either the minimize button or maximize button, whichever you used most recently. When you click the restore button, Abaqus/CAE returns the viewport to its previous size and location. The restore

button reverts to the minimize or maximize button that it had replaced. The restored viewport becomes the current viewport and is moved to the front of the canvas, hiding any other viewports in the drawing area.

Additional information

- *Working with viewports*
- *Managing viewports on the canvas*

Cascading viewports

You can arrange the viewports on the canvas in a cascading pattern. This may be useful to expose hidden viewports or simply to reduce clutter in the drawing area. Abaqus/CAE arranges the viewports so that the title bars are visible but the contents may be hidden by another viewport.

1. Use the **Minimize** button in the upper right corner of a viewport to minimize any viewports that you do not want to cascade.
2. From the main menu bar, select **Viewport->Cascade**.

   Abaqus/CAE sets all the non-minimized viewports to the same size and arranges them from the upper left corner toward the lower right corner of the drawing area. The current viewport is always positioned at the front of the canvas; all other viewports are arranged in numerical order. Minimized viewports are positioned in front of any cascaded viewports except the current viewport.

 **Tip:** You can also cascade non-minimized viewports by clicking  in the **Viewport** toolbar.

Additional information

- *Minimizing, maximizing, restoring, or deleting a viewport*
- *Managing viewports on the canvas*

Tiling viewports

You can arrange the viewports on the canvas in a tile pattern. This may be useful to expose hidden viewports or simply to reduce clutter in the drawing area. Abaqus/CAE arranges the viewports so that the contents are visible but parts of the title bars may be hidden.

1. Use the **Minimize** button in the upper right corner of a viewport to minimize any viewports that you do not want to tile.
2. Do one of the following:

   - From the main menu bar, select**Viewport**->**Tile Horizontally** to tile the viewports while retaining the largest possible horizontal dimension.
   - From the main menu bar, select**Viewport**->**Tile Vertically** to tile the viewports while retaining the largest possible vertical dimension.

   Abaqus/CAE sets all the non-minimized viewports to the same size and arranges them to fill the drawing area. All tiled viewports are arranged in numerical order. Minimized viewports are positioned in front of any tiled viewports except the current viewport.



Tip: You can also tile non-minimized viewports by clicking  or  in the **Viewport** toolbar.

Additional information

• *Managing viewports on the canvas*

Working with viewport arrow and text annotations

This section explains how to create, modify, and manage arrow and text annotations.

In this section:

- *Annotating viewports*
- *Creating an arrow annotation*
- *Creating a text annotation*
- *Creating combined arrow and text annotations*
- *Manipulating annotations in the current viewport*
- *Editing arrow annotation attributes*
- *Editing text annotation attributes*
- *Plotting annotations in the current viewport*
- *Hiding annotations in the current viewport*
- *Copying viewport annotations to another database*
- *Rearranging the annotation list order*

Annotating viewports

Abaqus/CAE provides three types of viewport annotations that you can use to annotate models or results: text strings, arrows, and a combination arrow and text annotation. You use the **Viewport** menu to create these annotations, and you can position them anywhere on the viewport. *Figure 1* shows text strings and arrows used to annotate a model. Arrow and text annotations are anchored to points in the viewport. The anchor points define the position of the annotation with respect to either the viewport size and shape or a point in the model coordinates. When you manipulate the model or resize the viewport, the annotations follow the position of their anchor points.

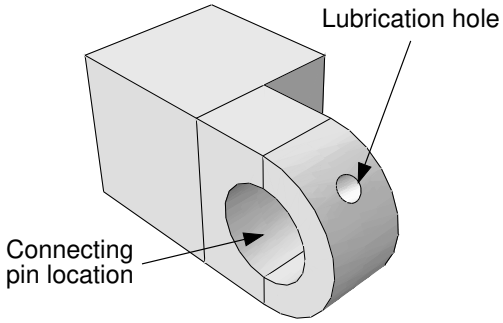


Figure 1: Arrow and text annotations.

Viewport arrow and text annotations are saved with the model database or with the output database when you exit the Abaqus/CAE session. You can also copy annotations from a model to an output database and vice versa.

Warning: You can always create or copy annotations to an output database; however, you can save them only if you opened the database with write privileges.

Text

Text annotations can consist of any characters that can be displayed using the fonts available on your workstation. Abaqus/CAE does not restrict the length or number of text lines in an annotation. However, you must take the size of the viewport into consideration when creating text annotations; the text should fit in the desired viewport location. You can place text anywhere on the viewport, and you can move a text annotation or change its anchor point after you have created it. Each text annotation can be displayed using different fonts, colors, and backgrounds, but you cannot change these attributes within a single text annotation.

Arrows

You can create arrows anywhere in the viewport; typically, an arrow will connect a text annotation to a point in model coordinates. Arrows can have one of several different line thicknesses, line styles, and endpoint styles and can be displayed in any color available on your workstation. You can use one or two anchor points to control the position of an arrow annotation. For example, you can anchor the arrowhead to a point on the model and the tail to a viewport location so that when the model is manipulated the arrow still points to the desired feature but the tail is stationary. You can modify and move an arrow after you have created it.

Arrow and text combination

You can create a combined arrow and text annotation. This method uses a reduced set of options to create an arrow annotation, anchored at the arrowhead, with a text annotation positioned at the arrow's tail. This creation method is provided to simplify the addition of information about specific parts of a model or results. You cannot edit the annotations' position during creation, except by using the **Previous** button ( ) in the prompt area.

However, once you create them, the arrow and text are saved independently; so you can edit them using all the options available for each annotation type.

In addition to the viewport annotations that you can create, Abaqus/CAE generates some annotations to provide the context of the view in a viewport. Abaqus/CAE generates the 3D compass and the view triad to show the orientation of the model in the viewport; in the Visualization module the legend, the title block, and the state block are generated to indicate various aspects of the analysis results in the viewport. The **Viewport Annotation Options** dialog box allows you to show or hide all viewport annotations—yours and those generated by Abaqus/CAE—you can also use it to edit the attributes of the generated annotations. For more information, see *Customizing viewport annotations*.

Additional information

- *What are arrow and text annotations?*
- *Creating an arrow annotation*
- *Creating a text annotation*
- *Creating combined arrow and text annotations*
- *Customizing viewport annotations*

Creating an arrow annotation

You can create an arrow to help annotate the contents of a viewport, and you can place the arrow anywhere on the viewport. Typically, you would use a *combined arrow and text annotation* to connect a text annotation to an object in a viewport, but you may want to add individual arrows if a text annotation applies to multiple locations or if you want full control over the anchor points and position without editing the arrow after you create it.

1. From the main menu bar, select **Viewport->Create Annotation->Arrow**.

   Tip: You can also create an arrow annotation by clicking  in the ***Viewport*** *toolbar*.

   An anchor symbol appears in the viewport. The default anchor point changes as you create new annotations in the session. If this is the first arrow annotation, the anchor point is in the lower left corner of the viewport.

2. If desired, click **Change Anchor** in the prompt area and do one of the following:

   - Choose a new anchor point in the viewport.
   - Select a viewport corner or center point from the list in the prompt area.
   - Click **Pick From Model** in the prompt area, and select a node or vertex or enter the coordinates of the desired anchor point.

     Note:

     If you anchor an annotation to a node or vertex, Abaqus/CAE uses the model coordinates of the selected point; the annotation will not follow model deformation in the Visualization module.

3. Choose one of the following options to position the endpoint of the arrow:

   - To position the endpoint on the current anchor point, click **On Anchor** in the prompt area.
   - To position the endpoint anywhere on the viewport, move the cursor to the desired position, and + click mouse button 1.

   By default, Abaqus/CAE creates an arrowhead at the first endpoint; the second endpoint does not have a symbol.

4. Repeat Steps 2 and 3 to anchor and position the second endpoint of the arrow annotation.

   Abaqus/CAE opens an **Arrow Editor** dialog box.

5. Complete the information in the **Arrow Editor** dialog box to finalize the arrow annotation.

   For more information, see *Editing arrow annotation attributes*.

6. Click **Preview** to view your changes in the viewport.
7. To finish creating your arrow annotation, click **OK**.

   Abaqus/CAE updates your changes in the viewport, closes the editor, and begins creating another arrow annotation starting at Step 2.

8. To stop creating arrow annotations, click mouse button 2 in the viewport or the  button in the prompt area.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*
- *Manipulating annotations in the current viewport*
- *Editing arrow annotation attributes*

Creating a text annotation

You can create one or more lines of text to annotate the contents of a viewport and place them anywhere on the viewport. Typically, you would use a *combined arrow and text annotation* to connect a text annotation to an object in a viewport, but you may also add text annotations to provide a title or other general information that applies to the entire model or results in the viewport.

1. From the main menu bar, select **Viewport->Create Annotation->Text**.

   Tip: You can also create a text annotation by clicking  in the ***Viewport*** *toolbar*.

   An anchor symbol appears in the viewport with the annotation `Text-n`. The default anchor point changes as you create new annotations in the session. If this is the first text annotation, the anchor point is in the lower left corner of the viewport.

2. If desired, click **Change Anchor** in the prompt area and do one of the following:

   - Choose a new anchor point in the viewport.
   - Select a viewport corner or center point from the list in the prompt area.
   - Click **Pick From Model** in the prompt area, and select a node or vertex or enter the coordinates of the desired anchor point.

     Note:

     If you anchor an annotation to a node or vertex, Abaqus/CAE uses the model coordinates of the selected point; the annotation will not follow model deformation in the Visualization module.

3. Choose one of the following options to position the text annotation in the viewport:

   - To position the text on the current anchor point, click **On Anchor** in the prompt area.
   - To position the text anywhere on the viewport, move the cursor to the desired position, and + click mouse button 1.

   Abaqus/CAE opens a **Text Editor** dialog box.

4. Complete the information in the **Text Editor** dialog box to finalize the text annotation.

   For more information, see *Editing text annotation attributes*.

5. Click **Preview** to view your changes in the viewport.
6. To finish creating your text annotation, click **OK**.

   Abaqus/CAE updates your changes in the viewport, closes the editor, and begins creating another text annotation starting at Step 2.

7. To stop creating text annotations, click mouse button 2 in the viewport or the  button in the prompt area.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*
- *Manipulating annotations in the current viewport*

- *Editing text annotation attributes*

Creating combined arrow and text annotations

You can create an arrow and a text annotation at the same time to relate the text to a particular feature or result in the viewport.
This method creates an arrow annotation with the anchor point at the arrowhead and a separate text annotation anchored at the same location and positioned at the arrow's tail. During creation, your annotation options are limited; for example, you cannot offset the arrowhead from the anchor point or select a second anchor point for the tail of the arrow (the text position).

Once you create them, Abaqus/CAE saves the arrow and text annotations independently so that you can edit them using all the options available for each annotation type as described in *Editing arrow annotation attributes*, and, *Editing text annotation attributes*, respectively. Since they are not linked, you can also move the arrow and text independently.

Tip: To move an arrow and text annotation while keeping the same relative positions, select both annotations and position the text in the desired location, then select only the arrow to edit the arrowhead position.

Additional information

- *What are arrow and text annotations?*
- *Manipulating annotations in the current viewport*

Create combined arrow and text annotations

1. From the main menu bar, select **Viewport->Create Annotation->Arrow & Text**.

   Tip: You can also create an arrow and text annotation by clicking  in the ***Viewport*** *toolbar*.

2. Select the arrow target by choosing a model node or vertex in the viewport or entering model coordinates in the prompt area.

   Abaqus/CAE displays the anchor symbol and arrowhead at the selected point and the default text string `Text-n` appears at the cursor position.

3. Position the text on the viewport; move the cursor to the desired text position and click mouse button 1.

   Abaqus/CAE displays the **Arrow & Text Editor** dialog box.

4. Complete the information in the **Arrow & Text Editor** dialog box to finalize both annotations.

   Click **Preview** at any time to see your changes in the viewport; click **OK** to save the arrow and text annotations and to close the **Arrow & Text Editor**.

Edit the text

1. Click the **Text** tab in the **Arrow & Text Editor**, if it is not already selected.
2. To display or suppress a box outlining the text, toggle **Show bounding box**. The bounding box visually separates the text from the underlying model or results.
3. Type the desired text.

   When typing a text annotation, you can use standard mouse and keyboard editing techniques such as backspace, copy, and paste; press **[Enter]** to start a new line.

4. To customize the background, click one of the following:

   - **Match viewport** to match the background to the viewport color.
   - **Transparent** to eliminate the background and show only the text.

- **Other color** to reveal other background color options.

If you selected **Other color**:

a. Click the color sample .

Abaqus/CAE displays the **Select Color** dialog box.

b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

c. Click **OK** to close the **Select Color** dialog box.

The color sample changes to the selected color.

5. Customize the text font.

a. Click **Set Font**.

The **Select Font** dialog box appears.

b. Use the **Select Font** dialog box to choose the font characteristics you want. For more information, see *Customizing fonts*.

c. When you are done, click **OK** to implement your changes and to close the **Select Font** dialog box.

6. Choose the text color.

a. Click the color sample .

Abaqus/CAE displays the **Select Color** dialog box.

b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.

c. Click **OK** to close the **Select Color** dialog box.

The color sample changes to the selected color.

7. Choose **Left**, **Center**, or **Right** justification to arrange the text within the bounding box area.

Edit the arrow

1. Click the **Arrow** tab in the **Arrow & Text Editor**, if it is not already selected.

2. Select the desired line style; you can choose a solid line or several styles of dashed line for your arrow.

3. Select the desired line thickness.



Note:

Changing the line thickness also changes the size of the arrowheads.

4. Select the desired arrowhead symbol.

5. Choose the arrow color.

a. Click the color sample .

Abaqus/CAE displays the **Select Color** dialog box.

b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.

c. Click **OK** to close the **Select Color** dialog box.

The color sample changes to the selected color.

Manipulating annotations in the current viewport

To manipulate viewport arrow and text annotations, you must first select them. Use **Viewport->Edit Annotations** from the main menu bar to select arrow or text annotations from the current viewport.



Note:

To manipulate the viewport annotations generated by Abaqus/CAE, use the **Viewport Annotations Options** dialog box (for more information, see *Customizing viewport annotations*").

You can move, copy, edit, hide, or delete the selected annotations. You can also rearrange the order that Abaqus/CAE uses to display annotations; the order determines which annotation will be "in front" if viewport or model manipulations cause more than one annotation to appear in the same location.

1. From the main menu bar, select **Viewport->Edit Annotations**.

   Tip: You can also edit an annotation by clicking  in the ***Viewport*** *toolbar*.

2. Click an arrow or text annotation to select it. To select additional annotations, **[Shift] + Click** instead of clicking. For more information, see *Selecting objects within the current viewport*.

   Abaqus/CAE highlights the selected annotations and their anchor points. Circular handles are the anchor connection points; they appear near each end of a selected arrow and at the reference point of a selected text annotation. Dashed lines indicate an offset between the anchors and the handles.

3. Do one of the following:

To move or copy annotations:

If you selected text or multiple annotations, drag your selection to move it around the viewport. Press **[Ctrl]** before releasing mouse button 1 to copy the annotations instead of moving them. If you selected a single arrow annotation, drag the arrow shaft to move the arrow. You can drag annotations anywhere on the viewport, even outside the viewable area. Moving or copying annotations in the viewport does not change their anchor points.

To move or copy the ends of an arrow annotation:

If you selected a single arrow annotation, notice that the circular handles are larger than when you select multiple annotations; this indicates that you can move each end of the arrow independently. Drag one of the circular handles to lengthen, reduce, or reorient the arrow. Press **[Ctrl]** before releasing mouse button 1 to create radial copies of the arrow instead of moving it. You can drag the endpoint anywhere on the viewport, even outside the viewable area. Moving or copying an arrow by dragging an endpoint does not change the arrow's anchor points.

To edit an annotation:

Double-click an arrow or text annotation to open the **Arrow Editor** or **Text Editor** dialog box, respectively. If you prefer, you can also click mouse button 3 and select **Edit** from the popup menu.

- For an arrow annotation you can change the line style, line weight, or color; and you can change the offset, arrowhead style, and anchor point for each end of the arrow, as well as add a gap between the handle positions and the arrow endpoints. For more information, see *Editing arrow annotation attributes*.

- For a text annotation you can edit the text; show or hide a box around the text; change the background color, font, font color, justification, and rotation; or change the reference point, anchor point, and offset. For more information, see *Editing text annotation attributes*.

To hide selected annotations:

Click mouse button 3, and select **Hide** from the menu that appears.

Abaqus/CAE removes the selected annotations from the current viewport. To show the annotations again, you can plot them from the ***Annotation Manager***. You can also hide or plot annotations by selecting them in the Model Tree, clicking mouse button 3, and selecting the desired option from the menu that appears.

To delete selected annotations:

Press `Backspace` or `Delete`. Abaqus/CAE displays a warning dialog box; click **Yes** to delete the annotations.

Abaqus/CAE deletes the selected annotations from both the current viewport and the associated database.

To rearrange the display order of annotations:

By default, Abaqus/CAE displays viewport annotations in the order that they are created or plotted in the viewport; the last annotation created or plotted will appear in front of any preexisting annotations. Changes in the display order will not affect the current view unless two or more annotations appear in the same location. All viewport annotations are displayed in front of the model.

Click mouse button 3, and select **Bring to Front**, **Send to Back**, **Bring Forward**, or **Send Backward** from the popup menu to change the display order. Alternatively, to move a single annotation to the front, plot it again from the ***Annotation Manager***.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Editing arrow annotation attributes

You can change the following attributes of arrow annotations:

- Name (use the **Annotation Manager** to rename existing annotations).
- Line thickness and line style.
- Color.
- Anchor point and offset for each end.
- Endpoint style.
- Gap between the endpoint and the arrowhead or tail position.

If you are creating a new annotation, Abaqus/CAE applies many of your customizations not only to the current annotation but also to any new annotations you subsequently create. If you are editing an existing annotation, Abaqus/CAE applies your changes only to the selected annotation.

You can also copy an arrow annotation and its attributes and edit them to create a new arrow annotation. For more information, see *The Annotation Manager*.

1. Use one of the following methods to access the **Arrow Editor** dialog box:
   - Select an arrow annotation from the **Annotation Manager**, and click **Edit**.
   - Select **Viewport->Edit Annotations** or  in the ***Viewport** toolbar*, and double-click an arrow annotation in the viewport.
   - Create a new arrow annotation in the viewport; Abaqus/CAE opens the editor after you select the endpoints in the viewport.
2. If you are creating a new annotation, you can edit the name of the arrow.
3. Select the desired line style; you can choose a solid line or several styles of dashed line for your arrow.
4. Select the desired line thickness.

    **Note:**

   Changing the line thickness also changes the size of the arrowheads.

5. Choose the arrow color.
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.
6. Click the **Start Point** tab or the **End Point** tab.

   Abaqus/CAE highlights the selected point and its anchor point in the viewport.
7. Select a new anchor point; the following methods are available:

    **Tip:** To avoid changing the position of the annotation, use the **Pick Anchor** button to change the anchor point and recalculate the **Offset** between the anchor point and the annotation.

Predefined

Select this method to anchor the annotation to a predefined point of viewport geometry. The available points are the viewport corners, the center, and the midpoint of each edge. The anchor point definition changes if you modify the viewport shape.

% Viewport X, Y

Select this method to anchor your annotation to a point based on the position of the lower left viewport corner and the viewport size. Enter the anchor point as a percentage of the total width and height of the current viewport. As with the **Predefined** method, the anchor point definition changes if you modify the viewport shape.

Model point X, Y, Z

Select this method to anchor your annotation to a point on the model. Enter the model coordinates of the new anchor point in the text field. The anchor point definition, and any annotation linked to it, changes if you manipulate the model. For example, if you rotate the model in the viewport, the anchor point and the arrow endpoint will rotate with it.

Pick Anchor button

Select this method to change the anchor point without changing the current position of the annotation.

Abaqus/CAE hides the **Arrow Editor** dialog box and prompts you to select a point from the viewport. Alternatively, you can pick one of the predefined viewport points from the list in the prompt area or click **Pick From Model** to pick a model node or vertex from the viewport or to enter the coordinates of a model point in the prompt area.

Each selection type corresponds to one of the preceding anchor point selection methods. After you make your selection, the **Arrow Editor** dialog box reappears with your selection indicated in the appropriate anchor selection method and the offset value recalculated for the new anchor point.

8. Enter new *X*- and *Y*-values to change the **Offset**, in millimeters, between the endpoint and its anchor point.

   Note:

   After you close the **Arrow Editor**, you can also change the **Offset** by dragging each endpoint, or the entire arrow, in the viewport. For more information, see *Manipulating annotations in the current viewport*.

9. Select the desired arrowhead symbol.

   The defaults are a plain line for the **Start Point** and a filled arrow for the **End Point**.

10. If desired, add a **Gap** between the endpoint position and the start or end of the arrow annotation. Adding a gap is comparable to leaving a space between dimension lines and object lines in the Sketcher or in a CAD drawing; it can increase the clarity of your annotation. For example, you can indicate a specific point on a model and add a gap so that the arrowhead does not obscure the point.
11. Repeat Steps 7–10 to edit the remaining endpoint of the arrow.
12. Click **Apply** to see your changes in the viewport.



Note:

Once you have applied your changes in the viewport, you cannot recover the original settings except by recreating the annotation or editing the options again.

13. Click **OK** to close the **Arrow Editor**.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Editing text annotation attributes

You can change the following attributes of text annotations:

- Name (use the **Annotation Manager** to rename existing annotations).
- Bounding box display.
- Background color.
- Font style and color.
- Justification.
- Rotation angle.
- Reference point location and offset.
- Anchor point location.

If you are creating a new annotation, Abaqus/CAE applies many of your customizations not only to the current annotation but also to any new annotations you subsequently create. If you are editing an existing annotation, Abaqus/CAE applies your changes only to the selected annotation.

You can also copy a text annotation and its attributes and edit them to create a new text annotation. For more information, see *The Annotation Manager*.

1. Use one of the following methods to access the **Text Editor** dialog box:
   - Select a text annotation from the **Annotation Manager**, and click **Edit**.
   - Select **Viewport->Edit Annotations** or  in the ***Viewport** toolbar*, and double-click a text annotation in the viewport.
   - Create a new text annotation in the viewport; Abaqus/CAE opens the editor after you select the text position in the viewport.
2. If you are creating a new annotation, you can edit the name of the text annotation.
3. To display or suppress a box outlining the text, toggle **Show bounding box**. The bounding box visually separates the text from the underlying model or results.
4. Type the desired text.

   When typing a text annotation, you can use standard mouse and keyboard editing techniques such as backspace, copy, and paste; press **[Enter]** to start a new line.
5. To customize the background, click one of the following:
   - **Match viewport** to match the background to the viewport color.
   - **Transparent** to eliminate the background and show only the text.
   - **Other color** to reveal other background color options.

   If you selected **Other color**:

   a. Click the color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

   The color sample changes to the selected color.

6. Customize the text font.
   a. Click **Set Font**.

      The **Select Font** dialog box appears.

   b. Use the **Select Font** dialog box to choose the font characteristics you want. For more information, see *Customizing fonts*.
   c. When you are done, click **OK** to implement your changes and to close the **Select Font** dialog box.
7. Choose the text color.
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.
8. Choose **Left**, **Center**, or **Right** justification to arrange the text within the bounding box area.
9. Enter a rotation angle (in degrees) to orient the text; 0° is horizontal.

   Abaqus/CAE rotates the text counterclockwise about the **Reference Point**.
10. Click the **Location** tab.
11. Select a new **Reference Point**.

    The reference point is the point where a text annotation is attached to its anchor point; Abaqus/CAE also uses the reference point as the center of rotation if you rotate the text. The following methods are available:

Predefined

Select this method to use a predefined point of the text bounding box geometry as the reference point. The available points are the bounding box corners, the center, and the midpoint of each edge. The definition of the reference point changes if you modify the annotation text.

% Text X, Y

Select this method to define the reference point based on the position of the lower left bounding box corner and the bounding box size. Enter the reference point as a percentage of the total width and height of the text bounding box. As with the **Predefined** method, the reference point definition changes if you modify the text.



Note:

Abaqus/CAE determines the lower left corner position based on a 0° rotation angle. If you rotate the text, the corner position is also rotated accordingly.



Tip: You can select a reference point outside of the bounding box by entering values less than 0% or greater than 100%.

12. Enter new *X*- and *Y*-values to change the **Offset**, in millimeters, between the reference point and the anchor point.



Note:

After you close the **Text Editor**, you can also change the offset by dragging the text annotation in the viewport. For more information, see *Manipulating annotations in the current viewport*.

13. Select a new anchor point; the following methods are available:



Tip: To avoid changing the position of the annotation, use the **Pick Anchor** button to change the anchor point and recalculate the **Offset** between the anchor point and the annotation.

Predefined

Select this method to anchor the annotation to a predefined point of viewport geometry. The available points are the viewport corners, the midpoint of each edge, and the center of the viewport. The anchor point definition changes if you modify the viewport shape.

% Viewport X, Y

Select this method to anchor your annotation to a point based on the position of the lower left viewport corner and the viewport size. Enter the anchor point as a percentage of the total width and height of the current viewport. As with the **Predefined** method, the anchor point definition changes if you modify the viewport shape.

Model point X, Y, Z

Select this method to anchor your annotation to a point on the model. Enter the model coordinates of the new anchor point in the text field. The anchor point definition, and any annotation linked to it, changes if you manipulate the model. For example, if you rotate the model in the viewport, the anchor point and the arrow endpoint will rotate with it.

Pick Anchor button

Select this method to change the anchor point without changing the current position of the annotation.

Abaqus/CAE hides the **Arrow Editor** dialog box and prompts you to select a point from the viewport. Alternatively, you can pick one of the predefined viewport points from the list in the prompt area or click **Pick From Model** to pick a model node or vertex from the viewport or to enter the coordinates of a model point in the prompt area.

Each selection type corresponds to one of the preceding anchor point selection methods. After you make your selection, the **Text Editor** dialog box reappears with your selection indicated in the appropriate anchor selection method and the offset value recalculated for the new anchor point.

14. Click **Apply** to see your changes in the viewport.



Note:

Once you have applied your changes in the viewport, you cannot recover the original settings except by recreating the annotation or editing the options again.

15. Click **OK** to close the **Text Editor**.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Plotting annotations in the current viewport

When you open a database that contains saved viewport annotations or create a new viewport for an open database, Abaqus/CAE does not automatically display your viewport arrow or text annotations. You can use the **Annotation Manager** to plot arrow and text annotations in the current viewport; you can also select the annotations in the Model Tree, click mouse button 3, and select **Plot** from the menu that appears.

Annotations generated by Abaqus/CAE are displayed according to the settings in the **Viewport Annotation Options** dialog box (for more information, see *Customizing viewport annotations*").

1. From the main menu bar, select **Viewport->Annotation Manager**.

   Tip: You can also display the **Annotation Manager** dialog box by clicking  from the ***Viewport*** *toolbar*.

2. Click the **MDB** or **ODB** radio button to select the source of the annotations to be plotted.
   a. If you have more than one output database open or are otherwise uncertain of the database files involved, position the cursor over the **MDB** or **ODB** radio button to display a tooltip containing the database path and file name.
   b. If necessary, change the active output database by displaying the desired database in the current viewport. Alternatively, you can open a new viewport to display the database or close all the other output databases.

   Abaqus/CAE lists the annotations available in the selected database.

3. Select annotations from the list in the dialog box (for more information, see *Selecting multiple items from lists and tables*).

   Tip: Toggle on **Highlight selections in viewport** to preview the selected annotations.

4. Click **Plot**.

   Abaqus/CAE plots the selected annotations in the current viewport.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Hiding annotations in the current viewport

If there are arrow and text annotations obscuring the model view in the viewport, you can use the **Annotation Manager** to hide them.

Annotations generated by Abaqus/CAE are displayed according to the settings in the **Viewport Annotation Options** dialog box (for more information, see *Customizing viewport annotations*").

1. From the main menu bar, select **Viewport->Annotation Manager**.

   Tip: You can also display the **Annotation Manager** dialog box by clicking  from the ***Viewport*** *toolbar*.

2. Click the **MDB** or **ODB** radio button to select the source of the annotations to be hidden.
   a. If you have more than one output database open or are otherwise uncertain of the database files involved, position the cursor over the **MDB** or **ODB** radio button to display a tooltip containing the database path and file name.
   b. If necessary, change the active output database by displaying the desired database in the current viewport. Alternatively, you can open a new viewport to display the database or close all the other output databases.

   Abaqus/CAE lists the annotations available in the selected database.

3. Select annotations from the list in the dialog box (for more information, see *Selecting multiple items from lists and tables*).

   Tip: Toggle on **Highlight selections in viewport** to preview the selected annotations.

4. Click **Hide**.

   Abaqus/CAE hides the selected annotations in the current viewport.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Copying viewport annotations to another database

You can copy viewport annotations that you created from a model database (**MDB**) to an output database (**ODB**), and vice versa. You cannot copy annotations from one output database to another. Annotations generated by Abaqus/CAE contain model- and results-specific information; this information is automatically available in every database and cannot be copied (for more information, see *Customizing viewport annotations*").

1. From the main menu bar, select **File->Open** to open the desired model database and output database (`.cae` and `.odb`) file.

   Note:

   You do not need to have write privileges for the database files to copy annotations; however, if you do not have write privileges, you cannot save the copied annotations.

2. From the main menu bar, select **Viewport->Annotation Manager**.

   Tip: You can also display the **Annotation Manager** by clicking from the ***Viewport** toolbar*.

3. Click the **MDB** or **ODB** radio button to select the source of the annotations to be copied.
   a. If you have more than one output database open or are otherwise uncertain of the database files involved, position the cursor over the **MDB** or **ODB** radio button to display a tooltip containing the database path and file name.
   b. If necessary, change the active output database by displaying the desired database in the current viewport. Alternatively, you can open a new viewport to display the database or close all the other output databases.

   Abaqus/CAE lists the annotations available in the selected database.

4. Select annotations from the list in the dialog box (for more information, see *Selecting multiple items from lists and tables*).

   Tip: Toggle on **Highlight selections in viewport** to preview the selected annotations.

5. Click the **Copy to MDB** or **Copy to ODB** button.

   Abaqus/CAE copies the selected annotations.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*

Rearranging the annotation list order

You can change the order of viewport annotations in the **Annotation Manager** list. Changing the order allows you to keep related annotations—such as text and an arrow that annotate the same area—together in the list. If you plot the multiple selections from the **Annotation Manager** list to a viewport, the display order is driven by the list order; that is, the annotation at the top of the list will also be in front of all the other annotations in the viewport. However, you can change the display order of annotations in each viewport independently, either by plotting individual annotations from the **Annotation Manager** or by selecting an annotation, clicking mouse button 3, and using the options in the popup menu. (For more information, see *Manipulating annotations in the current viewport*.)

1. From the main menu bar, select **Viewport->Annotation Manager**.

   Tip: You can also display the **Annotation Manager** by clicking  from the ***Viewport*** *toolbar*.

2. Select annotations from the list in the dialog box (for more information, see *Selecting multiple items from lists and tables*).

   Tip: Toggle on **Highlight selections in viewport** to preview the selected annotations.

3. Use the buttons along the bottom of the **Annotation Manager** to rearrange the list.

   Note:

   The **Move Up** and **Move Down** buttons are not available if you select multiple annotations from the list.

   Abaqus/CAE moves the selected annotations within the list.

Additional information

- *What are arrow and text annotations?*
- *The Annotation Manager*
- *Manipulating annotations in the current viewport*

Linking viewports for view manipulation

This section explains how to link multiple Abaqus/CAE viewports for synchronized view manipulation.

In this section:

- *Using linked viewports*
- *Linking viewports*

Using linked viewports

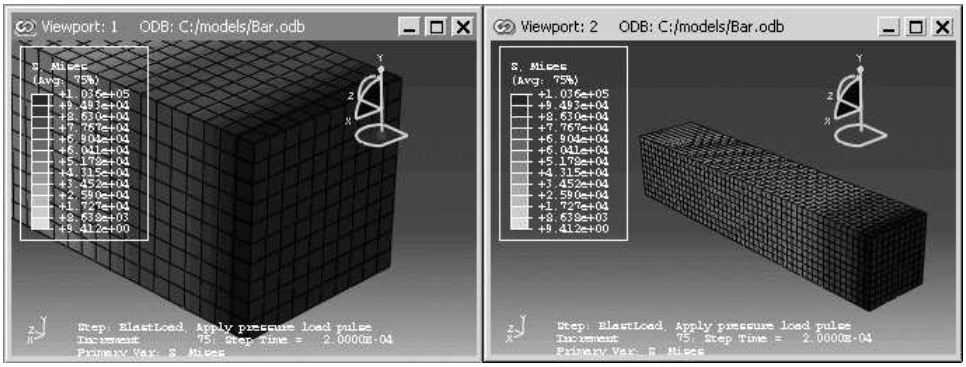
Linked viewports enable you to manipulate your view of objects in different viewports simultaneously. When you manipulate an object in one linked viewport using any of the view manipulation tools described in *Manipulating the view and controlling perspective*, Abaqus/CAE performs the same action in all linked viewports in your session.

Abaqus/CAE allows only one set of linked viewports, so every viewport in your session is either independent or a part of the group of linked viewports. When you change the view in an independent viewport, Abaqus/CAE changes the view in that viewport only.

When using the standard view manipulation tools (see *Understanding the view manipulation tools*), the manipulations in the linked viewports are not dependent on the view orientation in each viewport. For example, panning the view to the left will pan the view to the left in all linked viewports, regardless of different view orientations in each viewport. When manipulating the view using the 3D compass (see *The 3D compass*), the manipulations in the linked viewports are dependent on the view orientation in each viewport. For example, panning the view along the *X*-axis will pan the view along the *X*-axis in all linked viewports; the view orientation in each viewport determines the direction of motion.

You can activate or deactivate viewport linking for your session from the **Viewport** menu, the **Viewport** toolbar, or the **Linked Viewports Manager**. If this option is deactivated, all viewports in your session are independent. By default, if this option is activated, all viewports in your session are linked. In the bottom portion of the **Linked Viewports Manager**, you can toggle off a viewport's checkbox to exclude it from the group of linked viewports. From the **Linked Viewports Manager**, you can control which characteristics the linked viewports share. By default, all of the characteristics are shared. Several of the linked viewport options are available in all modules; some of the options, such as displaying the same plot state in all linked viewports, are applicable only in the Visualization module.

Abaqus/CAE indicates linked viewports with a red chain link icon that appears on the left side of the viewport title bar, as shown in the two viewports below.



Additional information

- *Using the view manipulation tools*
- *Understanding plot states and plot customization*

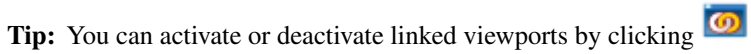
Linking viewports

You can control which viewports in your Abaqus/CAE session will be linked and the characteristics that the linked viewports will share.

Several of the linked viewport options are available in all modules; some of the options, such as displaying the same plot state in all linked viewports, are applicable only in the Visualization module.

1. Link viewports using one of the following methods:

   - From the main menu bar, select**Viewport->Link Viewports**.

     Tip: You can activate or deactivate linked viewports by clicking  in the *Viewport toolbar*.

   - From the main menu bar, select**Viewport->Linked Viewports Manager** and toggle on **Link viewports**.

     Tip: You can also display the **Linked Viewports Manager** by clicking  in the *Viewport toolbar*.

2. By default, all of the characteristics of the viewports are shared in linked viewports. From the **Options** portion of the **Linked Viewports Manager**, you can control which characteristics are shared.

   - Toggle on **Animation** to use the same animation settings in the linked viewports. After linking, when animation is started or stopped, the animation state of the current viewport will be applied to all viewports that are linked to the current one. This option applies only in the Visualization module.
   - Toggle on **Display groups** to perform selected display group operations simultaneously in all linked viewports. If you select specific items for the display group by names or labels, the same display group operation is performed in all linked viewports that share the items specified. This option applies only in the Visualization module.
   - Toggle on **Field output** to display results for the same field output variable in the linked viewports. When you change the field output variable for one of the linked viewports, Abaqus/CAE displays the newly selected field output variable in all viewports with output databases that contain that field output variable. This option applies only in the Visualization module.
   - Toggle on **Frames** to set the step and frame in the linked viewports. This option applies only in the Visualization module.
     - Select **Frame navigation** to set the step and frame in the linked viewports using the **ODB Frame** buttons (**First**, **Previous**, **Next**, and **Last**). For example, **Next** increments the frame number currently displayed in each of the linked viewports.
     - Select **Frame selection** to set the linked viewports to the same step and frame number by making a selection in the **Step/Frame** dialog box or the **Frame Selector**.
   - Toggle on **Plot options** to display the same plot options in the linked viewports. This option applies only in the Visualization module.
   - Toggle on **Plot states** to display the same plot state or states in the linked viewports. This option applies only in the Visualization module.
   - Toggle on **Rotation centers** to use a common center of rotation across all linked viewports.

- Toggle on **Section points** to share section point selection across viewports. Linking section point selection by category occurs only if all the section categories in the current viewport are present in the other linked viewports; for envelope plots, this restriction does not exist. This option applies only in the Visualization module.
- Toggle on **View cuts** to display the same view cuts in all linked viewports. This option applies only in the Visualization module.
- Toggle on **View manipulations** to enable simultaneous view manipulations in the linked viewports.
- Toggle on **Viewport annotation options** to display viewport annotations in the same way across all linked viewports. When you display, hide, or customize the display of a viewport annotation such as the legend in the **Viewport Annotation Options** dialog box, Abaqus/CAE applies that change across all linked viewports.
- Toggle on **XY data from history output** to share *X–Y* data created from history output across viewports. This option applies only in the Visualization module.

Use **Select All** to select all the options and **Unselect All** to clear all the selected options.

3. By default, all viewports in your session are linked when you link viewports. From the **Linked Viewports** portion of the **Linked Viewports Manager**, deselect the viewports that you want to remain independent.
4. If you made changes in the **Linked Viewports Manager**, click **OK** to apply your changes and to close the manager.

   The linked viewports exhibit the shared behavior you specified in the **Options** selections.

Additional information

- *Using the view manipulation tools*
- *Understanding plot states and plot customization*
- *Customizing viewport annotations*
- *Creating or editing a display group*
- *Reading X–Y data from output database history output*
- *Saving an X–Y data object*
- *Producing an X–Y plot*
- *Selecting section point data*
- *Selecting the results step and frame*

Working with background images and movies in viewports

This section explains how to display images and movies in the background of Abaqus/CAE viewports.

In this section:

- *Using background images and movies*
- *Displaying a background image*
- *Displaying and customizing a background movie*
- *Customizing the appearance of background images and movies*

Using background images and movies

You can customize the viewports in your session by displaying an image or movie in the viewport background. Both background images and movies are viewport-specific, so you can display a different image or movie in each viewport in your session. Background images persist in a viewport as you change modules, while background movies appear in the Visualization module only. Abaqus/CAE displays images and movies on top of the existing viewport background, so if you have customized the color of the viewport background, the image or movie might obscure some or all of the custom background color.

You can use background images to help you while you create your model; for example, an image of a completed prototype can help you to align part instances in an assembly. Alternatively, a background image can serve as a watermark or display a logo when you generate images of your model.



Note: In the Sketch module you can display a second background image that can help you sketch parts more effectively. Abaqus/CAE displays the Sketcher background image on top of the module-wide background image when the Sketch module is selected and hides the Sketcher background image in all other modules. See *Managing images in the Sketcher background*.

Background movies can help you to compare the results of an Abaqus analysis with experimental results. For example, if you display a background movie that shows deformation in a prototype, you can animate the results from a similar Abaqus analysis and compare the animations in a single viewport.

Before you can display an image or movie in the viewport background, you must add the file to your Abaqus/CAE session. To add an image from the **Image/Movie Options** dialog box, click



; then enter a name for the image or movie and provide its location. Images in the session are available in all modules, while movies are available in the Visualization module only. Both images and movies persist for your session only; they are not saved to the model database or output database.

Abaqus/CAE supports background images in the following formats: Bitmap (`.bmp`), PNG (`.png`), GIF (`.gif`), JPEG (`.jpg`, `.jpeg`), TIFF (`.tif`), XPM (`.xpm`), PCX (`.pcx`), ICO (`.ico`), TGA(`.tga`), and RGB (`.rgb`).

Abaqus/CAE supports background movies that satisfy the following two criteria:

- The movie file's format must be supported on your Abaqus/CAE platform. For Linux systems Abaqus/CAE supports MP4 format (`.mp4`), Audio Video Interleave format (`.avi`), QuickTme format (`.mov`), and animated GIF format (`.gif`). For Windows systems Abaqus/CAE supports MP4, AVI, QuickTime, and animated GIF formats, as well as Mpeg movie format (`.mpeg`, `.mpg`, `.mlv`, `.wm`) and Windows Media Format (`.asf`, `.wmv`, `.wm`).
- The codec used to create the background movie file must be one of the codecs available for creating movie files in Abaqus/CAE. For example, you can display a QuickTime movie file in the viewport background only if it was created using one of the three codecs available for QuickTime movie creation in Abaqus/CAE: Raw 8, Raw 24 or RLE 24. The available codecs for creating each movie file format in Abaqus/CAE are described in *Choosing the animation file format*.

Additional considerations for selecting the codec used to create the background movie file:

- You need to create the movie file with a codec whose 64-bit version is available and installed on your computer so it can be read by Abaqus/CAE.
- Codecs on one computer might differ from the ones available on other computers. In particular, a movie file created on a 32-bit system using a 32-bit codec might not open in Abaqus/CAE if the 64-bit version of the codec is unavailable on that system.

- If the codec to read an AVI file is not installed on your computer, Abaqus/CAE reports information about the unsupported compression format and the color depth (number of color bits in a pixel).
- Third-party software is available to convert to RLE or RAW AVI format, which can be read by Abaqus/CAE.

Additional information

• *Choosing background colors*

Displaying a background image

The viewport background image that you select is displayed in the current viewport only, and it remains visible in that viewport as you change modules. Because background images are viewport-specific, you can display different images in each viewport that either help you with different tasks or enable you to distinguish viewports more quickly.

While the module-wide background image appears in all modules, its appearance may be obscured by the Sketcher background image (in the Sketch module) or the background movie (in the Visualization module). When active, each of these viewport decorations are displayed on top of the module-wide background image in the selected module. You can also include the background image in a printout of a viewport; see *Selecting which part of the image to print*, for more information.

1. From the main menu bar, select **View->Image/Movie Options**.

   The **Image/Movie Options** dialog box opens with the **Image** page selected.
2. Toggle **Show image** to display or hide the background image in the viewport.
3. Select an image file to display:

   - To display an image that has been defined for your session, expand the **Image name** list and select the image name.
   - To add a new image, click ; then enter a name and specify a file location in the dialog box that appears. You can either enter the file location directly in the **File Name** field or click  to navigate to it in the **Select Image File** dialog box.
4. Click **OK** to apply your changes and to close the **Image/Movie Options** dialog box.

   If desired, you can also customize the scale, positioning, or translucency of the background image; see *Customizing the appearance of background images and movies*.

Additional information

- *Displaying and customizing a background movie*
- *Customizing the appearance of background images and movies*

Displaying and customizing a background movie

You can display some or all of a movie file in the viewport background. Background movies appear in the Visualization module only and, when active, they are displayed on top of the viewport background image. If you print a viewport when the background movie is active, Abaqus/CAE displays the current frame of the movie in the background of the printed image.

This section describes how to display an image using the **Image/Movie Options** dialog box. Alternatively, you can display and customize background movies from the **Movies** container in the Results Tree or from the **Movie Manager** dialog box. To display the movie manager, select **Tools->Movie->Manager**.

1. From the main menu bar, select **View->Image/Movie Options**. In the **Image/Movie Options** dialog box that appears, click the **Movie** tab.
2. Toggle **Show movie during animation** to display or hide the background movie while animations are playing.
3. Select a movie file to display:
   - To display a movie that has been defined for your session, expand the **Movie name** list and select the movie name.
   - To add a new movie, click ; then enter a name and specify a file location in the dialog box that appears. You can either enter the file location directly in the **File Name** field or click  to navigate to it in the **Select Movie File** dialog box.

     For newly added movies, the **Edit Movie** dialog box appears.
4. If desired, customize the movie's active frames from the **Edit Movie** dialog box:
   - Activate or deactivate frames at the beginning and end of the movie. You can either drag the **Active Frames** sliders or enter frame numbers in the **Start** and **End** fields to change these values.
   - Toggle on **Preview changes in viewport** to change the background movie's length dynamically as you adjust its active frames. When this option is not selected, the background movie's active frames do not change until you click **Apply** or **OK**.
   - Toggle on **Show movie only** to hide model data in the viewport. In some cases the model data can obscure the background movie, making movie editing difficult.
5. If desired, customize the movie's timeline from the **Edit Movie** dialog box. If you select a frame-based animation as the background movie, you can change the movie's starting frame and ending frame; if you select a time-based animation, you can change the movie's starting time and ending time. To change the ending value, expand the appropriate field under the **End** column and select **Specify**. Abaqus/CAE then allows you to enter a value in this field.
6. Click **OK** to apply your changes and to close the **Image/Movie Options** dialog box.

   If desired, you can also customize the scale, positioning, or translucency of the background movie; see *Customizing the appearance of background images and movies*.

Additional information

- *Displaying a background image*
- *Customizing the appearance of background images and movies*

Customizing the appearance of background images and movies

You can customize the appearance of a background image or movie by repositioning it in the viewport, by stretching or compressing it in the *X*- or *Y*-direction, and by adjusting its level of translucency. These customization options are available for both images and movies, but their settings are separate; for example, if you change the *X*-axis scale setting for background images, the same setting for background movies in the Visualization module remains unchanged.

1. From the main menu bar, select **View->Image/Movie Options**.

   The **Image/Movie Options** dialog box appears with the **Image** tab selected by default. To customize background movies, click the **Movie** tab.

2. Toggle **Show image** on the **Image** page or **Show movie during animation** on the **Movie** page to activate the customization options.
3. Select one of the **Position and scale** options for your background image or movie:

Fit to viewport

This option positions the background image or movie in the center of the viewport and scales it to the **Width** or **Height** of the current viewport. In addition, you can select the **Best** option, which scales either the background movie in the direction that causes the lesser amount of deformation to the image.

Auto-align

This option enables you to align the background image or movie in one of nine positions in the viewport: its center, any of the four corners, or centered along any of the four edges. Expand the **Alignment** list, then select one of the options from the graphical depictions of alignment in the list.

Manual

This option enables you to reposition the background image at a location other than the center of the viewport. Click **Manual**, then enter in the **Origin** field the location at which you want to position the lower left corner of the background image.

4. For **Auto-align** or **Manual** scaling, you can specify the amount of scaling along either axis. Specify the **X scale** or **Y scale** for the background image or movie. The default scale values are 1; increase the scale to stretch the image or movie along the selected axis, or decrease the value to compress it along the selected axis.
5. Drag the **Translucency** slider to the percentage of translucency you want. This setting changes the translucency of the background image or movie and blends it with the displayed model. A value of 0.00 is transparent, and a value of 1.00 is opaque. By default, background images or movies and models are displayed with opaque translucency.
6. Click **OK** to apply your changes and to close the **Image/Movie Options** dialog box.

Additional information

- *Displaying a background image*
- *Displaying and customizing a background movie*

Manipulating the view and controlling perspective

This chapter describes the view options, the view manipulation tools, the 3D compass, and the perspective tools, all of which control a camera that creates the view in a viewport.
The view options allow you to switch between two camera modes and numerically control some of their properties. The view manipulation tools and 3D compass control the camera to position, orient, and magnify objects in the view; you can also select custom views such as front and back, as well as define your own views. The perspective tools control whether Abaqus/CAE displays your model with or without perspective; using perspective gives a more realistic appearance for three-dimensional models.

In this section:

- *Understanding camera modes and view options*
- *Understanding the view manipulation tools*
- *The 3D compass*
- *Customizing the view triad*
- *Controlling perspective*
- *Using the view manipulation tools*
- *Using the 3Dconnexion motion controllers with Abaqus/CAE*

Understanding camera modes and view options

This section describes the camera modes used to create views in Abaqus/CAE.

In this section:

- *Camera modes and view terminology*
- *Using view options to control the camera*

Camera modes and view terminology

A view is a two-dimensional representation—a camera image—of your model or analysis results, displayed in a viewport. Abaqus/CAE uses a single camera to create the view in each viewport. You can choose from two camera modes to create the desired view of your model or results. The default mode allows you to position the camera anywhere outside the model. The movie mode allows you to position the same camera inside the model as well as outside it. In addition, the movie mode provides you with two clipping planes that you can use to eliminate objects from the view when they are too close to, or too far from, the camera. The view depth is not limited in default mode; objects in the view may be directly in front of the camera or at a distance that makes them too small for you to see in the viewport.

View manipulation tools are available so that you can fully utilize both camera modes. All view manipulation tools can be used in either camera mode, but an "alternate mode" of some view manipulation tools is intended primarily for use in movie mode. For example, the magnify tool allows you to magnify the current view without moving the camera; the alternate mode of this tool moves the camera closer to the model. The effects of both view manipulations appear identical; but when used in the default camera mode, the alternate mode stops working if the camera "hits" the outer edge of the model. In movie mode you can use the alternate mode of the magnify tool to move the camera into and through the model. The view manipulation tools are described in *Understanding the view manipulation tools*.

Figure 1 shows the two camera modes. The shaded areas in the figure represent the visible space—the view—in each camera mode.

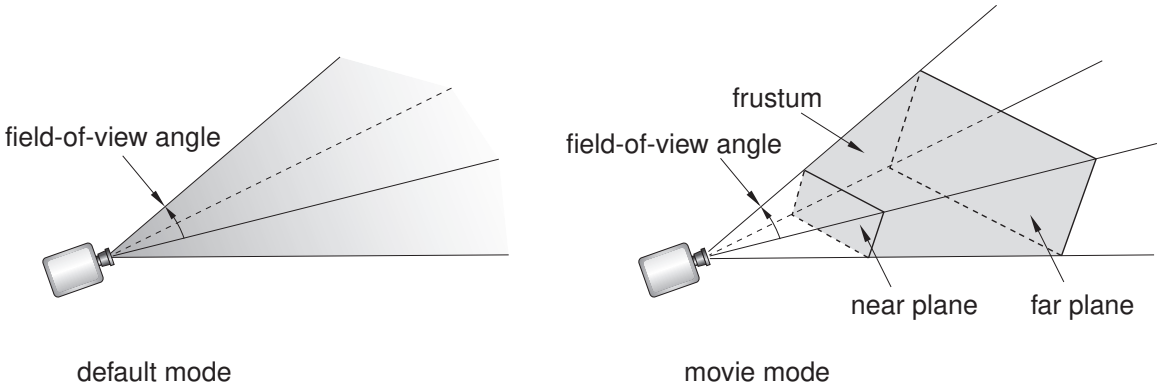


Figure 1: The default mode and movie mode camera views.

The camera terms that follow are used to describe the view that you see in a viewport:

Camera target

The camera target is a point in space that controls how the camera moves during most view manipulations. For all default views the camera target coincides with the center point of all objects in the view. The camera target moves away from the center of all objects when you use the alternate mode of the pan, rotate, and magnify view manipulation tools.

Frustum

The frustum is the three-dimensional space visible with the movie camera mode. The camera position forms the apex of a pyramid created by a left, right, top, and bottom plane (the same as it does in the default mode). To create the frustum, two additional planes are added to the default view, the near plane and the far plane. Only those objects (or portions of objects) that are within the frustum are visible in movie camera mode.

Field-of-view angle

The field-of-view angle is the larger of the angles between the left and right or the top and bottom planes that form the sides of the view. The angle that is used depends on the shape of the frustum (effectively the shape of the viewport); in both images of *Figure 1* the angle between the left and right planes is larger; therefore, this angle is indicated as the field-of-view angle. The field-of-view angle applies to both the default camera mode and the movie camera mode; changing the angle is comparable to adjusting the zoom on a stationary camera to expand or shrink the camera image.

The magnify, box zoom, and auto-fit view manipulation tools all change the field-of-view angle to resize the view in the viewport. See *Understanding the view manipulation tools*, for more information about these tools.

Near plane

The near plane lies perpendicular to the camera direction and is effective only in the movie camera mode. The distance from the camera to the near plane is the closest distance that an object can be to the camera and still remain in the view. The view from the default camera includes objects at any distance, as if the near plane were positioned directly in front of the camera lens.

The near plane is a clipping plane; it removes model surfaces and edges from view without cutting through the model. In the Visualization module you can cut the model such that interior surfaces are visible; for more information, see *Cutting through a model*.

Far plane

Like the near plane, the far plane lies perpendicular to the camera direction and is effective only in the movie camera mode. The distance from the camera to the far plane is the farthest distance that an object can be from the camera and still remain in the view. The view from the default camera includes objects at any distance, as if the far plane were positioned an infinite distance from the camera lens.

The far plane is a clipping plane; it removes model surfaces and edges from view without cutting through the model. In the Visualization module you can cut the model such that interior surfaces are visible; for more information, see *Cutting through a model*.

You use the view options, view manipulation tools, and perspective tools to change the camera mode settings or to change the relationship between the camera, the camera target, and the object that you are viewing. The current settings of these tools and options define the current view.

Using view options to control the camera

Use the view options to control the current camera mode and other options that you cannot set using the view manipulation tools.

Select **View->View Options** from the main menu bar to access the view options for the current viewport, as shown in *Figure 1*.

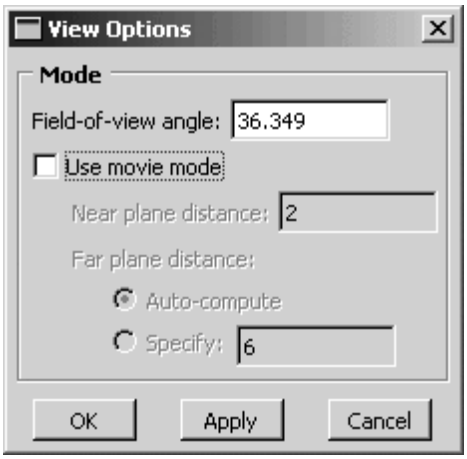


Figure 1: The View Options dialog box.

You can use the **View Options** dialog box to control:

- The field-of-view angle (effectively, the magnification of the current view).
- The camera mode (movie mode allows you to move the camera into and through the model).
- The distance from the movie mode camera to the near plane (the closest distance an object can be to the camera while remaining in the view).
- The distance from the movie mode camera to the far plane (the farthest distance an object can be from the camera while remaining in the view).

Note:

Specifying the near plane and far plane distance can improve the display performance for large models by excluding from view any portion of the model that lies beyond the specified range.

In the Visualization module the **View Options** dialog box also includes **Camera Movement** options. You can make the camera follow the motion of a local coordinate system and choose whether the camera also follows the rotation of the selected coordinate system. If movie mode is on, you can position the camera on the origin of the selected coordinate system. For more information on the view options in the Visualization module, see *Customizing camera movement*.

1. Select **View->View Options** from the main menu bar.

   The **View Options** dialog box appears.

2. If desired, change the **Field-of-view angle** to resize the view in the viewport:

   - Decrease the angle to magnify the view
   - Increase the angle to reduce the view

3. Toggle **Use movie mode** on or off to switch between the default and movie camera mode.

Activating the movie camera mode does not change the current view. However, if you apply the default camera mode (toggle off **Use movie mode**), you will see one of the following effects:

- If the model lies entirely within the frustum, the current view does not change.
- If a portion of the model is cut off by the near or far plane, that portion reappears in the view and Abaqus/CAE resets the near plane and far plane distances to include the entire object in the view the next time you activate movie mode.
- If the movie mode camera is positioned inside the model, the camera moves “back” along the current view direction so that the entire model is in front of the camera. Abaqus/CAE adjusts the field-of-view angle to display the model in a size similar to that in the movie mode view and resets the near plane and far plane distances to include the entire object in the view the next time you activate movie mode.

 **Tip:** Use the *cycle* view manipulation tool to return to the previous view.

4. If desired, adjust the **Near plane distance**.

   Abaqus/CAE removes from view any objects, or portions of objects, that lie between the movie mode camera and the near plane.

5. If desired, click **Specify** and adjust the **Far plane distance**.

   By default, Abaqus/CAE automatically computes the far plane distance, setting it to a value beyond the farthest point in the model and adjusting it when you manipulate the view so that no part of the model is excluded from view. If you specify a distance, Abaqus/CAE removes from view any objects, or portions of objects, that lie beyond that distance from the camera.

6. + Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Camera modes and view terminology*
- *Understanding the view manipulation tools*

Understanding the view manipulation tools

This section describes basic concepts you should understand before using the view manipulation tools.

In this section:

- *The view manipulation tools*
- *The pan view tool*
- *The rotate view tool*
- *The magnify tool*
- *The box zoom tool*
- *The auto-fit tool*
- *The cycle tool*
- *Custom views*
- *Numerically specifying a view*

The view manipulation tools

The camera position, orientation, and zoom factor combine to define the view of an object in the viewport. Your view of the assembly, as well as each of your parts, is positioned relative to a default Cartesian coordinate system, and the orientation of this default coordinate system within a viewport is indicated by the view triad. By default, an isometric view is used when a module first displays a three-dimensional part or assembly.

You can manipulate the view using the pan, rotate, magnify, box zoom, and auto-fit tools on the View Manipulation toolbar to control the relative positions of the camera, the camera target, and the model or results that you are viewing. For example, you might want to pan and zoom a contour plot to view an area of stress concentration. The view manipulation tools allow you to perform the following operations:

-  Move the view horizontally and vertically; that is, *pan* the view.
-  *Rotate* the view.
-  *Magnify* or reduce the view.
-  *Zoom* in to a selected area of the view.
-  *Rescale* the view to fill the viewport; that is, auto-fit the view.
-  *Cycle* through previous views.

Other types of view manipulations can be performed using the 3D compass; for more information, see *The 3D compass*.

You can click mouse button 3 to access the following view manipulation tools:

- **Set As Rotation Center**: Set the center of rotation at the position of the mouse click, which can be at any location in the viewport.
- **Use Default Rotation Center**: Clear a previously set center of rotation.
- **Center View**: Center the view at the position of the mouse click.

When an *X*–*Y* plot is displayed in the viewport, you can use the view manipulation tools to change the view of the *X*–*Y* curves. Because *X*–*Y* plots are two-dimensional, the rotate tool is disabled when the current viewport displays an *X*–*Y* plot.

Clicking a view manipulation tool puts you into the corresponding view manipulation mode. You then manipulate the view in a particular viewport by moving the cursor to that viewport and dragging or clicking as necessary. In addition, the pan, rotate, and magnify tools have alternate modes that you can access by holding the **[Shift]** key in conjunction with the normal use of these tools. The alternate modes of these tools are intended for use in the movie camera mode, but they can also be used in the default mode. For more information about camera terminology and the view modes, see *Understanding camera modes and view options*. To exit a view manipulation mode, do one of the following:

- Click mouse button 2.
- Click the cancel button  in the prompt area.
- Click the view manipulation tool again.
- Click any other view manipulation tool.

You can use the view manipulation tools as many times as necessary to reach the desired view, and you can perform the view manipulation in any viewport, regardless of what is being displayed. Abaqus/CAE stores the eight most recent

views from each viewport, and you can use the cycle view manipulation tool to cycle backward and forward through these views.

When you use the move, rotate, magnify, zoom, or rescale tools in a viewport that is linked to other viewports, Abaqus/CAE manipulates the view of objects in the linked viewports as well. For more information, see *Linking viewports for view manipulation*.

By default, Abaqus/CAE displays the image using the current render style (wireframe, filled, hidden line, or shaded) while you manipulate the view of an object. Alternatively, you can change the **Drag mode** in the **Graphics Options** dialog box to display the image as a simple wireframe while you manipulate the view; this mode allows faster manipulation of very large models in the shaded render style. The view reverts to the original render style when you complete a manipulation.

If you prefer to use menus rather than the tools on the View Manipulation toolbar, you can access all of the view manipulation tools through the **View** menu on the main menu bar. In addition, you can apply predefined and user-defined views using the Views toolbar, and you can numerically specify a precise view using the dialog box that appears when you select **View->Specify** from the main menu bar. For more information on custom and numerically specified views, see *Custom views*, and *Numerically specifying a view*, respectively.

Alternatively, you can enter three of the view manipulation modes by using a combination of keyboard and mouse actions.

- To rotate the view, press **[Ctrl][Alt]**, and hold down mouse button 1.
- To pan the view, press **[Ctrl][Alt]**, and hold down mouse button 2.
- To magnify or reduce the view, press **[Ctrl][Alt]**, and hold down mouse button 3.

Add **[Shift]** to any of these combinations to access the alternate modes of these tools. For example, press **[Shift][Ctrl][Alt]** and hold mouse button 3 to access the alternate mode of the magnify tool and move the camera closer to or farther from the objects in the view. The **[Shift]** key has no effect on the view manipulation tools when you are not using alternate modes. To exit a view manipulation mode after using one of the preceding actions, simply release the mouse button.

You can reconfigure these keyboard and mouse combinations to mimic the view manipulation interfaces used by five other common CAD applications by selecting **Tools->Options** from the main menu bar. See *Using view manipulation shortcuts*, for more information.

Additional information

- *Using the view manipulation tools*
- *Understanding camera modes and view options*

The pan view tool

When a model is displayed in the viewport and the pan mode is set, the position of your view of the model changes as you click and then drag the cursor, and a rubberband line indicates the amount of translation.

When you select the pan tool and the viewport in which to work, Abaqus/CAE enters pan mode, as indicated by the cursor.

Panning the view is comparable to moving the camera over a snapshot of the model, as shown in *Figure 1*; the snapshot moves in the viewport but any faces of the model that were hidden in the original camera position remain hidden as you pan.

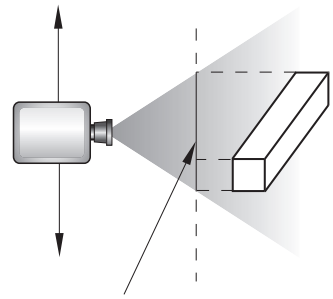


Figure 1: Panning the view.

The alternate mode of the pan tool, accessed by holding **[Shift]** while performing the manipulation, creates a more realistic camera view. Instead of a snapshot of the model, you pan the camera over the real model. Faces of the model that were hidden at the original camera position are exposed as you move the camera over the model, as shown in *Figure 2*.

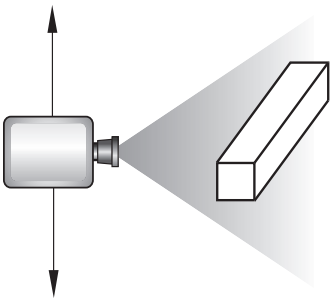


Figure 2: Panning the view in alternate mode.

Note:

If perspective is not on, the alternate mode of pan works identically to the standard pan tool.

For both modes of the pan tool, the initial location of the cursor is not important, as long as you place it within the viewport. Cursor motion is limited only by the physical bounds of your monitor, and panning will continue even if you move the cursor outside the viewport or window.

When an *X–Y* plot is displayed in the viewport, you can change your view of the *X–Y* curves in the plot by clicking and dragging the cursor in the grid. Abaqus/CAE updates the values in the axes as you manipulate your view of the *X–Y* data.

Additional information

- *The view manipulation tools*
- *Panning the view*
- *Understanding camera modes and view options*

The rotate view tool

In the rotate mode the cursor changes to two curved arrows, and a large circle appears in the viewport. To define the center of rotation, you can enter its coordinates directly or select a point from the viewport. Otherwise, Abaqus/CAE will rotate the view about the center of the viewport.

When you select the rotate tool and the viewport in which to work, Abaqus/CAE enters rotate mode. If you select a center of rotation by selecting a position from the viewport or entering coordinates, that rotation center position overrides the view center and remains selected until you select a new rotation center, display a different object, or choose the default rotation center. Your view of the model rotates as you drag the cursor, and a rubberband line indicates the amount and the direction of rotation. As you rotate your view of the model, the view triad indicates the orientation of the global coordinate system.

Note:

Abaqus/CAE disables the rotate tool when an *X*–*Y* plot is displayed in the current viewport.

The circle that is drawn when you enter rotate mode represents the silhouette of an imaginary sphere that surrounds the object. When you drag the mouse inside the circle, you might imagine that you are actually rotating the sphere, as you would a trackball. Your model is attached to the center of the sphere, so that rotating the sphere causes your view of the model to rotate as well.

You determine the axis of rotation as you move the cursor over the surface of the imaginary sphere. The rubberband line represents the intersection of a cutting plane with the sphere's surface, and the rotation axis is normal to this cutting plane. The angle of rotation is equal to the angle made by the rubberband line on the sphere's surface, so that dragging all the way across the circle produces a 180° rotation. *Figure 1* illustrates the imaginary sphere and a rubberband line being dragged across its surface.

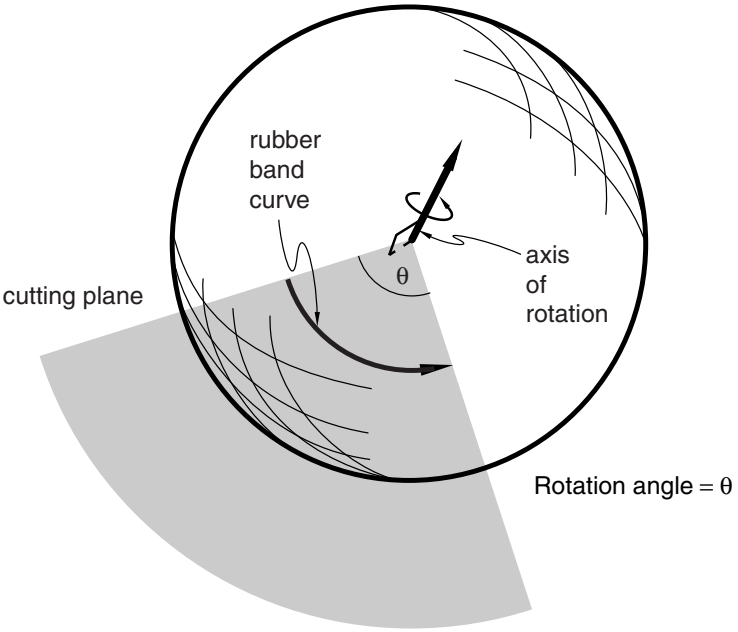


Figure 1: The rotate tool.

When you drag outside the circle, the rubberband line is superimposed on the edge of the circle, and your view of the object simply rotates about an axis normal to the screen and passing through the center of the circle. In the same way as it does for dragging inside the circle, the rubberband line represents the angle through which the object has rotated.

Using the default mode of view rotation is comparable to rotating the camera around the view center or selected center of rotation, as shown in *Figure 2*.

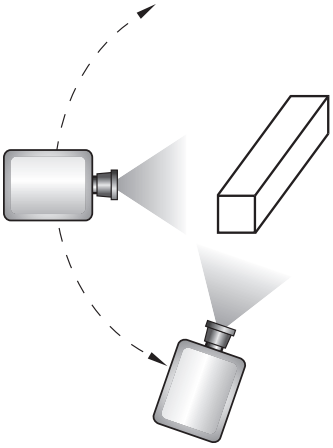


Figure 2: Rotating the camera about the target or selected center of rotation.

The alternate mode of the rotate tool, accessed by holding **[Shift]** while performing the manipulation, rotates the camera about itself, as shown in *Figure 3*. This moves the camera target and frustum without regard to the position of objects in the original view. Rotating the camera about itself is most useful when you are in movie mode and the camera is positioned inside the model. In this position, moving the camera target and frustum brings different portions of the interior of the model into view.

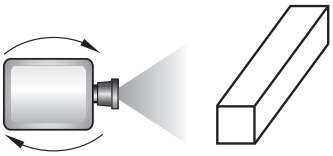


Figure 3: Rotating the camera about itself.

Note:

If you have selected a point as the center of rotation, your selection overrides the alternate mode of rotation.

In either mode, it is usually easier to obtain a desired rotation by performing a sequence of smaller rotations rather than one large one. If you need to abandon the rotation and return to a known orientation, use either the predefined views in the Views toolbar or the cycle view tool . Using any of the predefined views will also reset the camera target to the center of the model.

Because *X*–*Y* plots are two-dimensional, Abaqus/CAE disables the rotate tool when an *X*–*Y* plot is displayed in the current viewport.

Additional information

- *The view manipulation tools*
- *Rotating the view*
- *Understanding camera modes and view options*

The magnify tool

When you drag the cursor along the positive direction while in magnify mode, your view of the model or plot expands within the viewport, and a rubberband line indicates the relative magnification. Similarly, when you drag the cursor along the negative direction, your view of the model or plot contracts, and a rubberband line indicates the relative reduction.

When you select the magnify tool and the viewport in which to work, Abaqus/CAE enters magnify mode, as indicated by the magnify cursor . The positive and negative directions depend on your settings in the view manipulation options (see *Using view manipulation shortcuts*). If you are using the default Abaqus/CAE configuration for view manipulations, the positive direction is to the right and the negative direction is to the left. If you are using a nondefault configuration for view manipulations, the positive direction is upward and the negative direction is downward. To reflect the configuration settings, the rubberband line is horizontal for the default configuration and vertical for nondefault configurations.

The dragging action must start in the viewport, but you can continue to drag within the limits of your monitor. You can also drag repeatedly to achieve the desired view. The magnify tool recognizes only the horizontal (for the default configuration) or vertical (for nondefault configurations) component of your dragging motion, as indicated by the rubberband line. Consequently, you can achieve finer control by dragging diagonally across the screen, since this results in a smaller component of the cursor's motion in the effective direction than dragging the same distance along the effective direction.

Using the default mode of the magnify tool, as its name suggests, magnifies the view; as shown in *Figure 1*, the camera does not move with respect to the objects in the view. The magnification is caused by changing the field-of-view angle, the same method you use when changing the zoom on a stationary camera.

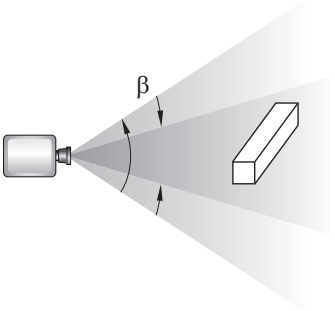


Figure 1: Magnifying the view.

The alternate mode of the magnify tool, accessed by holding **[Shift]** while performing the manipulation, keeps the field of view constant and moves the camera towards or away from the objects in the view, as shown in *Figure 2*.

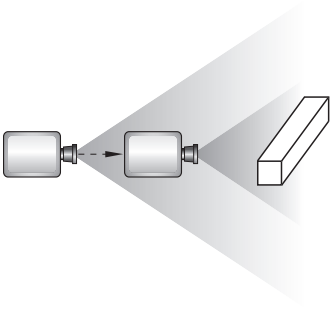


Figure 2: Moving the camera closer to the model.

Moving the camera in this manner is most useful when movie mode is on. Then your view is not limited; you can move the camera through the model such that any parts that you do not want to see are removed by the near or far planes or are actually behind the camera. If you are not using movie mode, the camera can move forward only until it reaches the outer limits of the model.

When an *X–Y* plot is displayed in the viewport, you can magnify your view of the data to focus in on a particular component of an *X–Y* curve. Abaqus/CAE updates the values in the axes as you change the magnification of the *X–Y* plot.

If you lose track of your position, you can use the *auto-fit* tool  to rescale the view to fit the viewport. Using the auto-fit tool also resets the camera target to the center of the model.

Additional information

- *The view manipulation tools*
- *Magnifying or reducing the view*
- *Understanding camera modes and view options*

The box zoom tool

You use the box zoom tool to select a rectangular area of your model or plot; Abaqus/CAE enlarges your view of the selected portion of your model or plot to fill the viewport.

When you select the box zoom tool  and the viewport in which to work, Abaqus/CAE enters box zoom mode, as indicated by a crosshair-shaped cursor. For *X*–*Y* plots, Abaqus/CAE enlarges your view of the selected *X*–*Y* curves and updates the axis values to match the data that you select.

Additional information

- *The view manipulation tools*
- *Zooming in to a selected area of the view*

The auto-fit tool

The View Manipulation toolbar contains an auto-fit tool.

Use the auto-fit tool  from the View Manipulation toolbar to quickly adjust your view of the model so that the model or model plot fills the viewport and is centered within it. When you fit a view of a model, the orientation does not change, as indicated by the view triad.

When you auto-fit an *X–Y* plot, the auto-fit tool resets the values in the axes to their specified minimum and maximum values; see *Customizing X–Y plot axes*. The auto-fit tool does not necessarily fill the viewport with the *X–Y* plot, because the chart options may dictate that the plot occupy only part of the viewport; see *Customizing X–Y plot appearance*, for more information about the chart sizing and positioning options.

Auto-fitting occurs in the current viewport as soon as you click the auto-fit tool. If you have more than one viewport, select the viewport that you want to rescale to make it the current viewport before selecting the auto-fit tool.

A separate option, **Auto-fit after rotations**, is available when you select **View->Graphics Options** from the main menu bar. You use this option to control whether or not Abaqus/CAE automatically rescales the view to fit the viewport as you rotate. For more information on using this option, see *Rotating the view*.

Additional information

- *The view manipulation tools*
- *Rescaling the view to fit the viewport*

The cycle tool

You can cycle through the eight most recent views in each viewport.

When you select the cycle tool  and the viewport in which to work, Abaqus/CAE enters cycle mode, as indicated by a cursor in the form of a two-way arrow.

To cycle through previous views, click in the viewport whose view you want to change. To control the direction of cycling, click **Backward** or **Forward** in the prompt area. The default is to cycle backward. After you cycle backward to the oldest available view, continued clicking has no effect. Similarly, after you cycle forward to the most recent view, continued clicking has no effect.

Additional information

- *The view manipulation tools*
- *Cycling through views*

Custom views

The Views toolbar allows you to apply a custom view to the model in the selected viewport. (A view is the combination of the position, orientation, and zoom factor of the model in the viewport.)

Note:

The Views toolbar is not visible in the Abaqus/CAE main window by default. To display the Views toolbar, select **View->Toolbars->Views** from the main menu.

Custom views include seven predefined views (such as front and back) and up to four user-defined views.

Predefined views

Predefined views are based on the six faces of an imaginary cube and an isometric view. The view triad indicates the orientation of this imaginary cube within a viewport. *Figure 1* illustrates the six predefined cube face views.

Note:

Predefined views have no effect when an *X*–*Y* plot is displayed in the current viewport.

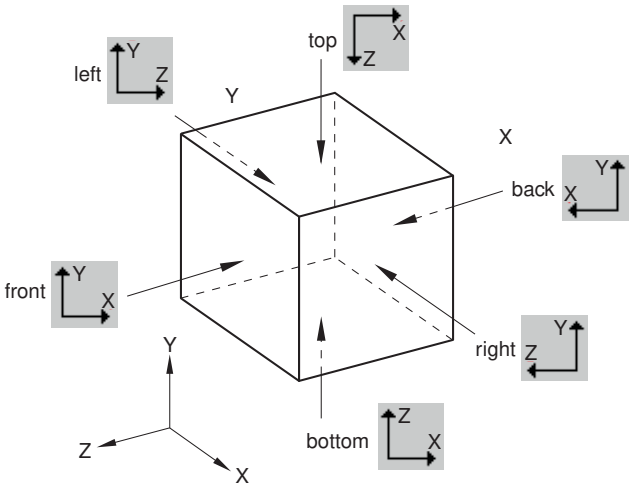


Figure 1: Predefined views.

User-defined views

You can use the view manipulation tools to position your view of a model in a viewport and then click in the Views toolbar to save the view as one of four user-defined views. You can use this saved view to restore the object in the viewport to a known orientation, and you can apply a saved view to other viewports. By default, saved views are not stored between sessions. If you want to retain a saved view for subsequent sessions, save it to an XML file, to the model database, or to an output database. For more information, see *Managing session objects and session options*.

The view consists of three components: orientation, zoom factor, and position. You can choose whether or not all three of these components are saved using the **Scale & Position** options, as follows:

Auto-fit

When you save a view after choosing this option, only the orientation is saved. When you apply a view saved with this option, the saved orientation is applied, but the zoom factor and position are adjusted to make the view fit the viewport.

Save current

When you save a view after choosing this option, the orientation, the zoom factor, and the position are all saved. When you apply a view saved with this option, the saved orientation, zoom factor, and position are all applied to the object in the viewport. To compare different objects in different viewports by placing the viewports side-by-side and applying a known orientation, zoom factor, and position to each, choose the **Save current** option.

Additional information

- *The view manipulation tools*
- *Applying custom views*
- *Saving a user-defined view*

Numerically specifying a view

You can bypass the view manipulation tools and specify a particular view numerically. Specifying a view is useful if you want to reproduce a particular view between Abaqus/CAE sessions or if numerically specifying a view is simpler and more convenient than applying a series of view manipulations.

Select **View->Specify** from the main menu bar to specify a view.

Tip: You can also specify a view by double-clicking the 3D compass.

You can use the following methods to specify your view:

Rotation Angles

Enter three angles (θ_1, θ_2, θ_3) representing the angles through which your view of the model rotates about the screen or model 1-, 2-, and 3-axes, respectively. Rotations are interpreted in the order (θ_1, θ_2, θ_3), and a positive angle represents a right-handed rotation about the axis. If you previously specified a nondefault center of rotation while using the rotate view tool (see *The rotate view tool*), the specified rotations will also be about this point. You must choose one of the following modes to apply the rotation:

- **Increment About Model Axes**. When you choose **Increment About Model Axes**, Abaqus/CAE simply applies the rotation to the current view. *Figure 1* shows the result of applying an incremental model axes rotation of 90, 0, 0 from the isometric view.

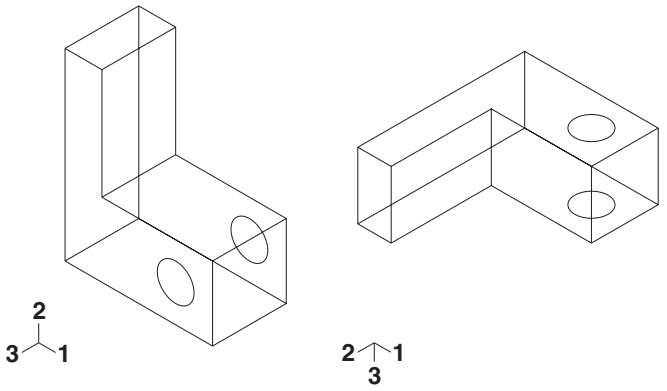


Figure 1: Specifying an incremental model axes rotation angle.

- **Increment About Screen Axes**. The screen *X*-axis is horizontal, the *Y*-axis is vertical, and the *Z*-axis is out of the screen. The origin of the screen axes is the camera target. In most cases the camera target coincides with the center of the viewport, but some view manipulation methods can move the camera target (for more information, see *Camera modes and view terminology*). When you choose **Increment About Screen Axes**, Abaqus/CAE simply applies the rotation to the current view. *Figure 2* shows the result of applying an incremental screen axes rotation of 90, 0, 0 from the isometric view.

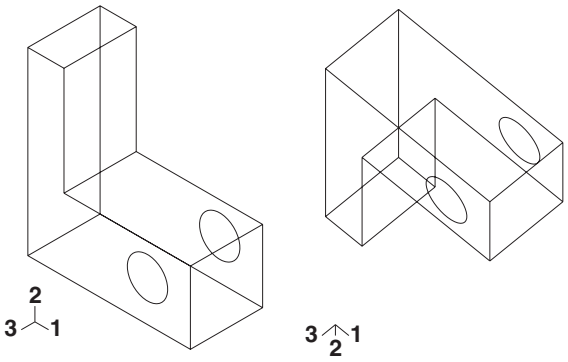


Figure 2: Specifying an incremental screen axes rotation angle.

- **Total Rotation From (0,0,1)**. When you choose **Total Rotation From (0,0,1)**, Abaqus/CAE first rotates the view to the default position (a view looking down the 3-axis with the 1- and 2-axes in the plane of the screen) and then applies the desired rotation. *Figure 3* shows the result of applying a total rotation of 90, 0, 0 from the isometric view.

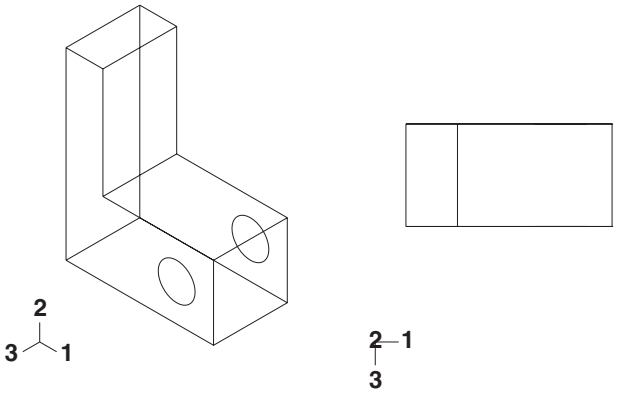


Figure 3: Specifying a total rotation angle.

Viewpoint

When you choose **Viewpoint**, you enter three values representing the 1-, 2-, and 3-position of an observer. Abaqus/CAE constructs a vector from the origin of the model to the position that you specify and rotates your view of the model so that this vector points out of the screen. *Figure 4* shows the result of applying a viewpoint of 1, 1, 1 (an isometric view) and a viewpoint of 1, 0, 0.

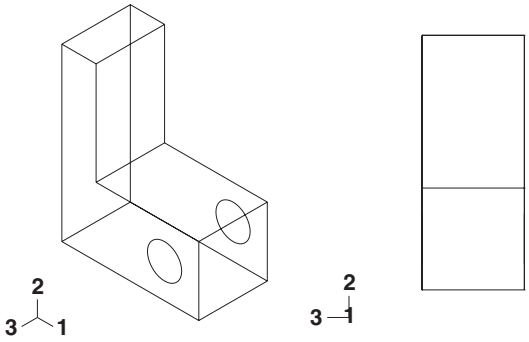


Figure 4: Specifying a viewpoint.

When you use the **Viewpoint** method to specify a view, you can also specify the **Up vector**. Abaqus/CAE positions your view of the model so that this vector points upward. *Figure 5* shows the result of applying an up vector of 0, 1, 0 and an up vector of 0, −1, 0 to an isometric view. The **Up vector** must not equal the **Viewpoint** vector.

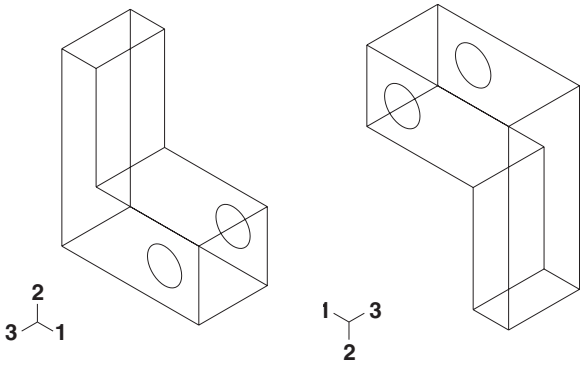


Figure 5: Specifying an Up vector.

Zoom

Enter a value representing a magnification factor. A value greater than 1 expands your view of the model in the viewport; for example, a **Zoom factor** of 2 doubles the size of your view of the model. A value between 0 and 1 contracts your view of the model in the viewport; for example, a value of 0.25 contracts your view of the model to a quarter of its original size. The value must be greater than zero.

You must choose one of the following methods to apply the zoom:

- **Absolute**. When you choose **Absolute**, Abaqus/CAE first fits the view to the viewport and then applies the desired **Zoom factor**.
- **Relative**. When you choose **Relative**, Abaqus/CAE applies the **Zoom factor** to the current view.

Pan

Enter values that Abaqus/CAE uses to **Pan** your view of the model by a specified horizontal and vertical distance. Abaqus/CAE moves the view relative to its current postion in the viewport. The values that you enter correspond to fractions of the viewport dimensions; the first value represents horizontal motion and the second value represents vertical motion. A positive first value moves your view of the model toward the right edge of the viewport, and

a positive second value moves your view of the model toward the top of the viewport. For example, if the viewport is 200 mm wide and 100 mm tall and you enter values of 0.5, −0.1 in the **Fraction of viewport to pan (X,Y)** field, Abaqus/CAE positions your view of the model 100 mm toward the right and 10 mm down from its current position.

Additional information

- *The view manipulation tools*
- *Applying a specified view*

The 3D compass

This section describes the basic functions and features of the 3D compass.

In this section:

- *About the 3D compass*
- *Rotating the view using the 3D compass*
- *Panning the view using the 3D compass*
- *Predefined views for the 3D compass*
- *Customizing the 3D compass*

About the 3D compass

The 3D compass is a viewport annotation that appears in the upper right-hand corner of a viewport.

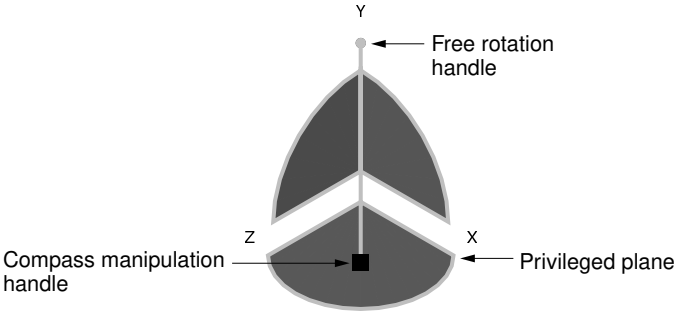


Figure 1: The 3D compass.

The 3D compass in Abaqus/CAE is based on the 3D compass used in CATIA V5. The 3D compass indicates the orientation of the model in the viewport, similar to the view triad. Unlike the view triad, you can manipulate the orientation of the 3D compass by clicking and dragging on it. When you manipulate the 3D compass, the viewport camera pans or rotates to change the viewport orientation accordingly. The behavior of the compass view manipulations is identical to the compass view manipulation behavior in CATIA V5.

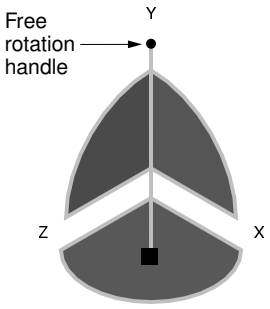
The 3D compass is a helpful shortcut for certain view manipulation options since it is available in all modules and during all procedures; you do not need to enter a view manipulation mode to change the viewport orientation using the 3D compass.

Rotating the view using the 3D compass

The 3D compass allows you to rotate the view of a model using two different methods: you can rotate freely in all directions, or you can constrain the manipulation to rotation about a specific axis. In both cases the model view rotates about the current center of rotation for the viewport, as defined by the rotate tool (see *The rotate view tool*).

Free rotation

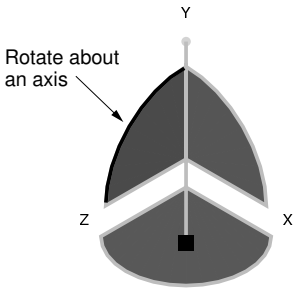
To rotate a model in any direction, click and drag the free rotation handle on the 3D compass:



As you drag the mouse, the compass rotates about its pivot in the direction of the mouse motion (the pivot point coincides with the compass manipulation handle). The rotation is dependent on the direction of the mouse motion, not on the location of the pointer; in other words, the compass continues to rotate as long as you continue to drag. As the orientation of the compass changes, the view of the model changes accordingly.

Rotation about an axis

You can also rotate a model about a specified axis, thereby maintaining a constant orientation in a particular direction during the manipulation. To rotate about an axis, click and drag one of the three arcs along the perimeter of the 3D compass:



As you drag the mouse, the compass rotates about the axis that is perpendicular to the plane subtended by the selected arc (the *X*-axis in the above example). The rotation is dependent on the location of the pointer. As you drag the mouse, the path of the pointer in the viewport is projected onto the selected compass arc. The compass rotates according to this projected path. As the orientation of the compass changes, the view of the model changes accordingly.

Additional information

- *What is a viewport?*
- *Camera modes and view terminology*
- *The rotate view tool*

Panning the view using the 3D compass

The 3D compass allows you to pan the view of a model using two different methods: you can pan along a specified axis, or you can pan within a specified plane.

The pan manipulations performed with the 3D compass are different than the manipulation performed with the standard pan view tool (see *The pan view tool*). The standard pan tool moves the camera laterally in front of the model; the camera is constrained to move only in a plane parallel to the viewport. The panning planes for the 3D compass are not necessarily parallel to the viewport. As a result, panning with the compass behaves like a combination of the standard pan and zoom tools in their alternate modes: the camera moves both laterally across the model and perpendicularly toward or away from the model. The orientation of the compass and model view does not change during pan manipulations.

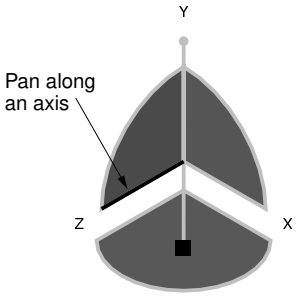


Note:

As with the alternate mode of the standard pan view tool, panning the view with the 3D compass also translates the viewport's center of rotation along with the camera. See *The rotate view tool*, for information about the center of rotation.

Panning along an axis

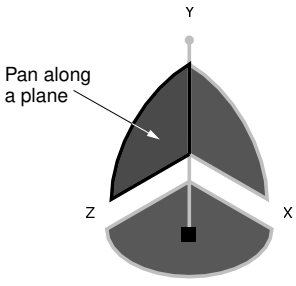
To pan the view along an axis, click and drag any of the straight axes on the 3D compass:



As you drag the mouse, the path of the pointer in the viewport plane is projected onto the selected compass axis (the *Z*-axis in the above example). The camera moves according to this projected linear path.

Panning along a plane

To pan the view along a plane, click and drag any of the quarter-circular faces on the 3D compass:



As you drag the mouse, the path of the pointer in the viewport plane is projected onto the selected compass plane (the *Y*–*Z* plane in the above example). The camera moves according to this projected path.

Additional information

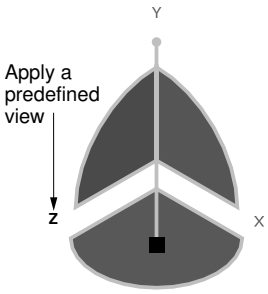
- *What is a viewport?*
- *Camera modes and view terminology*
- *The pan view tool*

Predefined views for the 3D compass

You can use the 3D compass to quickly set the viewpoint to one of six predefined views. You can also access the **Specify View** dialog box through the compass.

Predefined views

The predefined views correspond to the three planes of the 3D compass. To apply a predefined view, click the label for any of the axes on the 3D compass:



The view is adjusted so that the selected axis (the *Z*-axis in the above example) is perpendicular to the plane of the viewport. Clicking the same axis label again flips the view orientation to the opposite side of the viewport plane. In other words, clicking the same axis label repeatedly oscillates the view between the front and back side of the displayed model.

The six predefined views associated with the 3D compass are identical to the predefined views in the Views toolbar.

Numerically specifying a view

Double-clicking anywhere on the 3D compass opens the **Specify View** dialog box. You can use this method to numerically specify a viewpoint or camera location. See *Numerically specifying a view*, for more information.

Additional information

- *What is a viewport?*
- *Camera modes and view terminology*
- *Custom views*
- *Numerically specifying a view*

Customizing the 3D compass

To customize the look and orientation of the 3D compass, click mouse button 3 on the compass and select an option from the menu that appears. You can perform the following customizations:

Edit

Select **Edit** to display the **Specify View** dialog box and numerically specify a custom view orientation. See *Numerically specifying a view*, for details on specifying custom view orientations.

Tip: You can also double click the 3D compass to display the **Specify View** dialog box.

Change the privileged plane

The base of the compass (which contains the compass manipulation handle) is called the privileged plane. By default, the *X*–*Z* plane is the privileged plane in Abaqus/CAE. The privileged plane can be helpful in determining the "correct" orientation of a model in the viewport. In the default isometric orientation of the compass, the privileged plane appears at the bottom and the free rotation handle appears at the top; the axis from the privileged plane to the free rotation handle in effect indicates the "up" direction for the model.

If the *Y*-axis does not correspond to the "up" direction in a model, you can change the privileged plane to any of the three major planes in the compass. For example, select **Make XY the Privileged Plane** to set the *X*–*Y* plane as the privileged plane. Changing the privileged plane only reconfigures the shape of the 3D compass, as indicated in *Figure 1*; the view orientation and the predefined views in the Views toolbar are not changed.

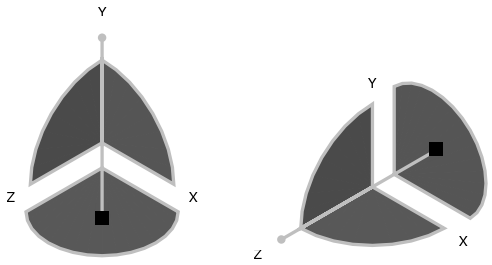


Figure 1: Changing the privileged plane from the *X*–*Z* plane (left) to the *X*–*Y* plane (right).

Hide

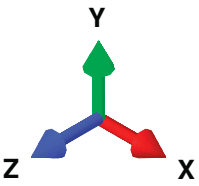
Select **Hide** to remove the 3D compass from the viewport display. To resume the display of the 3D compass, you must use the viewport annotation options (select **Viewport->Viewport Annotation Options** from the main menu). For information on controlling the visibility of viewport annotations, see *Using viewport annotation options*.

Help

Select **Help** to display a help window with documentation on using the 3D compass.

Customizing the view triad

The view triad, shown below, is a set of three perpendicular axes that indicate the orientation of your view of the model currently being displayed. The *X*, *Y*, and *Z* labels correspond to the 1-, 2-, and 3-directions, respectively. As you rotate your view of the model, the triad changes to indicate the new orientation. For more information on using the rotate tool, see *Rotating the view*.



The view triad and the 3D compass both indicate the view orientation, and they are always aligned with each other. You can directly manipulate the 3D compass orientation in the viewport, thereby changing the view of the model (see *The 3D compass*). The view triad acts only as a reference; it rotates with the 3D compass, but it cannot be directly manipulated.

You can use the **Viewport->Viewport Annotation Options** menu item to request or suppress the display of the triad and to control the triad's size, position, and appearance. You can also control the triad's labels, including their color and font.

1. From the main menu bar, select **Viewport->Viewport Annotation Options**.

   The **Viewport Annotation Options** dialog box appears.

2. Toggle **Show triad** to display or suppress the triad in the current viewport.

   When **Show triad** is toggled on, triad options become available.

3. Click the **Triad** tab.

   Abaqus/CAE displays the triad view options.

4. Enter the **Triad size** as a percentage of the viewport size. When you resize the viewport, the size of the triad changes accordingly. The minimum allowable triad size is 1% of the viewport, and the maximum allowable triad size is 50% of the viewport.

5. Enter percentage values for the triad X and Y positions in the **% Viewport X** and **% Viewport Y** boxes, respectively.

   A value of 0 for **% Viewport X** moves the triad origin to the extreme left of the viewport while a value of 100 moves it to the extreme right. A value of 0 for **% Viewport Y** moves the triad origin to the extreme bottom of the viewport, while a value of 100 moves it to the extreme top.

6. Choose the color of the triad labels:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

7. Click the arrow next to the **Labels** field and select either numerical or alphabetical labeling for the triad.

   The specified style appears in the **Labels** field.

8. Click **Set Label Font** to set the font type, size, and style using the dialog box that appears.
9. + Click **Apply** to implement your changes.
   Your changes are saved for the duration of the session.

Additional information

- *Using graphics display options*
- *Customizing viewport annotations*

Controlling perspective

Perspective representation accurately depicts the spatial relationship of three-dimensional objects in a two-dimensional plane. In other words, a three-dimensional model on your screen appears more realistic when perspective is turned on. Alternatively, parallel lines in the model appear parallel when perspective is turned off. Perspective affects all plots except *X–Y* plots, applies in all modules, and is turned on by default.

- To turn perspective on, select the  icon located in the View Options toolbar or select **View->Perspective** from the main menu bar.
- To turn perspective off, select the  icon located in the View Options toolbar or select **View->Parallel** from the main menu bar.

Your changes apply only to the current viewport and are saved for the duration of the session.

Additional information

• *Manipulating the view and controlling perspective*

Using the view manipulation tools

This section provides details of using the tools in the View Manipulation toolbar that allow you to manipulate the position, orientation, and scaling of the model or *X*–*Y* plot within a viewport.
When the viewport you manipulate is linked to other viewports, Abaqus/CAE also changes the view within every linked viewport in your session.

In this section:

- *Centering the view*
- *Panning the view*
- *Rotating the view*
- *Magnifying or reducing the view*
- *Zooming in to a selected area of the view*
- *Rescaling the view to fit the viewport*
- *Cycling through views*
- *Applying custom views*
- *Saving a user-defined view*
- *Applying a specified view*

Centering the view

Click mouse button 3 in the viewport to access the option to center the view.

You can also use the auto-fit tool  from the View Manipulation toolbar to quickly pan and magnify or reduce a view so that the view fills the viewport and is centered within it.

1. Position the cursor in the viewport at the location to be used to center the view, and click mouse button 3.
2. From the menu that appears, select **Center View**.

   Abaqus/CAE shifts the position that you selected to the center of the viewport.

Additional information

- *Linking viewports for view manipulation*
- *The view manipulation tools*
- *Rescaling the view to fit the viewport*

Panning the view

You can use the View Manipulation toolbar to move the view horizontally and vertically within the viewport.

Use the pan tool  from the View Manipulation toolbar to move the view horizontally and vertically within the viewport. Hold **[Shift]** while using the pan tool to access the alternate pan mode. The alternate mode shifts the view perspective as you pan the view, creating a more realistic image (for more information, see *The pan view tool*).

If the current viewport is linked to other viewports, Abaqus/CAE also moves the view of objects within all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

1. From the View Manipulation toolbar, click the pan tool  to enter pan mode.

    **Tip:** You can also select **View->Pan** from the main menu or press **[F2]**.

2. Position the cursor in the viewport whose view you want to change.

   The cursor changes to a four-headed arrow: 

3. Drag the cursor in any direction until you obtain the desired view.

   The position of your view of the model or *X*–*Y* plot in the viewport changes as you drag the cursor, and a rubberband line indicates the amount of translation.

    **Note:**

   The initial location of the cursor is not important, as long as you place it within the viewport. Cursor motion is limited only by the physical bounds of your monitor, and panning will continue even if you move the cursor outside the viewport or window.

4. Repeat Steps 2 and 3 until you achieve the desired view.
5. To exit pan mode, do one of the following:
   - Click mouse button 2.
   - Click the cancel button  in the prompt area.
   - Click the pan tool.
   - Click any other view manipulation tool.

    **Tip:** Use the *cycle* view manipulation tool to return to the previous view.

Additional information

- *Linking viewports for view manipulation*
- *The pan view tool*
- *The view manipulation tools*
- *Using the view manipulation tools*

Rotating the view

You can use the rotate tools to access several modes for manipulating views within the viewport.

Use the rotate tool  from the View Manipulation toolbar to rotate the view within the viewport. Hold **[Shift]** while using the rotate tool to access the alternate rotate mode. The alternate mode rotates the camera about itself instead of rotating it about the view center or the selected rotation center. The alternate mode is most useful when you use movie camera mode and position the camera inside of the model so that the model is surrounding the camera (for more information, see *The rotate view tool*). You can also specify a point in each viewport or select a location in each viewport to use as the center of rotation. Using the **Auto-fit after rotations** option in the **Graphics Options** dialog box, you can control whether or not Abaqus/CAE rescales your model to fit the viewport as you rotate.

 **Note:**

You cannot rotate *X*–*Y* plots.

If the current viewport is linked to other viewports, Abaqus/CAE also rotates objects within all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

Additional information

- *Linking viewports for view manipulation*
- *The rotate view tool*
- *The view manipulation tools*
- *Rescaling the view to fit the viewport*

Rotate the view

1. From the View Manipulation toolbar, click the rotate tool  to enter rotate mode.

    **Tip:** You can also select **View->Rotate** from the main menu or press **[F3]**.

2. By default, Abaqus/CAE rotates the view about the center of the viewport. You can change the center of rotation using one of the following methods:

   - Position the cursor at any location on the model or in the viewport, click mouse button 3, and select **Set As Rotation Center**.
   - Click **Select** in the prompt area. Select the center of rotation from the highlighted vertices in the viewport or enter coordinates to specify a point. In the Visualization module you can select a node and the center of rotation will remain on that node in both undeformed and deformed model states.

     When using this method, you can select a center of rotation only when there is existing geometry in the viewport. If you are working in the Sketcher, sketched points will not be available for selection.
   - Click mouse button 3 and select **Use Default Rotation Center**, or select **Use Default** in the prompt area to return to the default (center of viewport) rotation method.

   Your selected center of rotation persists in the viewport until you display another object in the viewport, select a new center of rotation, or return to the default rotation method.

3. Position the cursor in the viewport whose view you want to change.

   A large circle appears in the viewport and the cursor changes to a right facing arrow. If you selected a center of rotation in the viewport, it is highlighted.

4. Drag the cursor in any direction.

   The view rotates as you drag the cursor, and a rubberband line indicates the amount and direction of rotation.

   Tip: It is usually easier to achieve the desired orientation by performing a sequence of small rotations rather than a single large rotation.

   To rotate the view about the normal to the screen, move the cursor outside the circle and drag it clockwise or counterclockwise.

5. Repeat Steps 2, 3, and 4 until you achieve the desired views.
6. To exit rotate mode, do one of the following:
   - Click mouse button 2.
   - Click the cancel button  in the prompt area.
   - Click the rotate tool.
   - Click any other view manipulation tool.

   Tip: Use the *cycle* view manipulation tool to return to the previous view.

Rescale the view to fit the viewport as you rotate

1. From the main menu bar, select **View->Graphics Options**.

   The **Graphics Options** dialog box appears.
2. Toggle **Auto-fit after rotations** on to automatically rescale the view to fit the viewport as you rotate; toggle it off to disable automatic rescaling during rotation.
3. Click **OK** to implement your changes and close the dialog box.

   Your changes are saved for the duration of the session.

Magnifying or reducing the view

You can use the View Manipulation toolbar to change the scale of the view in the viewport.

Use the magnify tool  from the View Manipulation toolbar to change the scale of the view in the viewport. Hold **[Shift]** while using the magnify tool to access the alternate magnify mode. The alternate mode moves the camera closer to or farther from the objects in the view instead of changing the magnification. The alternate mode is most useful when you use movie camera mode (for more information, see *The magnify tool*).

Your configuration settings for the view manipulation options affect the behavior of the magnify tool. If you are using the default Abaqus/CAE view manipulation configuration, the magnify tool operates by dragging the cursor horizontally: dragging to the right zooms in and dragging to the left zooms out. In all other view manipulation configurations, the magnify tool operates by dragging the cursor vertically: dragging upward zooms in and dragging downward zooms out. See *Using view manipulation shortcuts*, for more information.

If your mouse has a wheel as mouse button 2, you can also change the scale of the view by scrolling up or down when the cursor is in the viewport. For *X*–*Y* plots the effect of this scrolling depends on the location of the cursor:

- When the cursor is within the *X*–*Y* plot grid, the *X*–*Y* plot will retain its aspect ratio as you magnify or reduce the view of the *X*–*Y* curves in the plot.
- When the cursor is placed on one of the axes, Abaqus/CAE stretches or compresses the *X*–*Y* plot along the axis you select. This functionality enables you to adjust your view of *X*–*Y* plots that exhibit much more change along one axis than the other.

If the current viewport is linked to other viewports, Abaqus/CAE also changes the scale of the view within all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

1. From the View Manipulation toolbar, click the magnify tool  to enter magnify mode.

    **Tip:** You can also select **View->Magnify** from the main menu or press **[F4]**.

2. Position the cursor in the viewport whose view you want to change.

   The cursor changes to a magnifying glass: 

3. Drag the cursor to change the view. The direction in which you drag depends on your view manipulation configuration settings. You can change these settings by selecting **Tools->Options** from the main menu bar and selecting a setting from the **View Manipulation Options** tabbed page.

   - If you are using the default view manipulation configuration, drag the cursor to the right of the starting point to magnify the view (zoom in). Drag the cursor to the left of the starting point to reduce the view (zoom out).
   - If you are using a nondefault view manipulation configuration, drag the cursor above the starting point to magnify the view (zoom in). Drag the cursor below the starting point to reduce the view (zoom out).

   Abaqus/CAE draws a rubberband line from the starting point as you drag the cursor across the screen. The rubberband line indicates the amount of zooming that has been applied, which is proportional to the component of your dragging motion in the effective direction.

4. Repeat Steps 2 and 3 until you achieve the desired view.
5. To exit magnify mode, do one of the following:

   - Click mouse button 2.

- Click the cancel button  in the prompt area.
- Click the magnify tool.
- Click any other view manipulation tool.

 **Tip:** Use the *cycle* view manipulation tool to return to the previous view.

Additional information

- *Linking viewports for view manipulation*
- *The magnify tool*
- *The view manipulation tools*
- *Using the view manipulation tools*

Zooming in to a selected area of the view

You can use the View Manipulation toolbar to enlarge the view so that a selected area fills the viewport.

Use the box zoom tool  from the View Manipulation toolbar to enlarge the view so that a selected area fills the viewport. If the current viewport is linked to other viewports, Abaqus/CAE enlarges the view in the linked viewports by the same factor. For more information, see *Linking viewports for view manipulation*.

1. From the View Manipulation toolbar, click the box zoom tool  to enter zoom mode.

    **Tip:** You can also select **View->Box Zoom** from the main menu or press **[F5]**.

2. Position the cursor in the viewport whose view you want to change.
   The cursor shape changes to crosshairs.
3. Position the cursor at one corner of the area to be enlarged.
4. Drag the cursor to the opposite corner.
   A rectangle indicates the area to be enlarged.
5. Release mouse button 1.
   The area defined by the rectangle enlarges to fill the viewport.
6. Repeat Steps 2 through 5 as many times as necessary to achieve the desired view.
7. To exit box zoom mode, do one of the following:
   - Click mouse button 2.
   - Click the cancel button  in the prompt area.
   - Click the box zoom tool.
   - Click any other view manipulation tool.

    **Tip:** Use the *cycle* view manipulation tool to return to the previous view.

Additional information

- *Linking viewports for view manipulation*
- *The box zoom tool*
- *The view manipulation tools*
- *Using the view manipulation tools*

Rescaling the view to fit the viewport

The View Manipulation toolbar contains a tool that rescales the view to fit the viewport.

Use the auto-fit tool  from the View Manipulation toolbar to quickly pan and magnify or reduce a view so that the view fills the viewport and is centered within it. When you fit a view, the orientation remains fixed, as indicated by the view triad. If the rescaled viewport is linked to other viewports, Abaqus/CAE magnifies or reduces the view in those viewports by the same factor. For more information, see *Linking viewports for view manipulation*.

From the View Manipulation toolbar, click the auto-fit tool  to enter auto-fit mode.

 **Tip:** You can also select **View->Auto-fit** from the main menu or press **[F6]**.

If you have only one viewport, Abaqus/CAE immediately scales the view to fit the viewport without changing the orientation, centers the view within the viewport, and exits fit mode. If you have more than one viewport, select the auto-fit tool and then place the cursor over the viewport you want to rescale. Click in the viewport to auto-fit; Abaqus/CAE rescales the view and exits fit mode.

 **Tip:** Use the *cycle* view manipulation tool to return to the previous view.

For information on how to automatically rescale the view to fit the viewport during rotation, see *Rotating the view*.

Additional information

- *Linking viewports for view manipulation*
- *The auto-fit tool*
- *The view manipulation tools*
- *Using the view manipulation tools*

Cycling through views

You can use the View Manipulation toolbar to cycle through previous views.

Use the cycle tool  from the View Manipulation toolbar to cycle through previous views; Abaqus/CAE saves the eight most recent views for each viewport.

If the current viewport is linked to other viewports, Abaqus/CAE also cycles through the views in all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

1. From the View Manipulation toolbar, click the cycle tool  to enter cycle mode.

    **Tip:** You can also select **View->Previous Views** from the main menu or press **[F7]**.

2. Position the cursor in the viewport whose view you want to change (the cursor changes to a two-way arrow); then click.
3. To control the direction of cycling, click **Backward** or **Forward** in the prompt area. The default is to cycle backward.
4. Repeat Steps 2 and 3 as many times as necessary to achieve the desired views.

   After you cycle backward to the oldest available view, continued clicking has no effect. Similarly, after you cycle forward to the most recent view, continued clicking has no effect.
5. To exit cycle mode, do one of the following:
   - Click mouse button 2.
   - Click the cancel  or **Done** button in the prompt area.
   - Click the cycle view tool.
   - Click any other view manipulation tool.

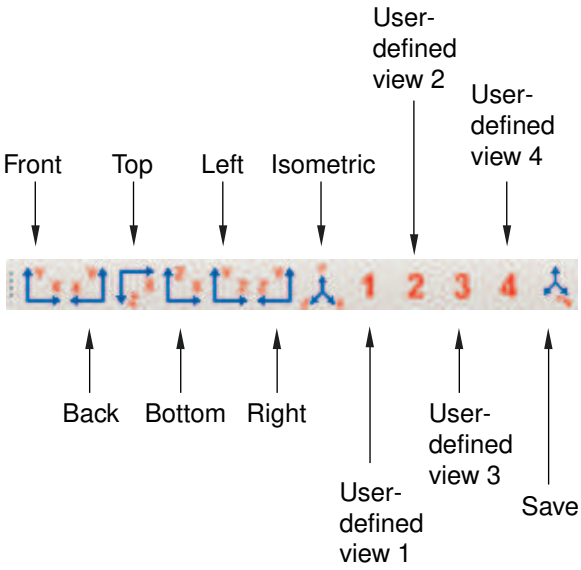
Additional information

- *Linking viewports for view manipulation*
- *The cycle tool*
- *The view manipulation tools*
- *Using the view manipulation tools*

Applying custom views

Use the Views toolbar to orient, scale, and position a view to one of seven predefined or four user-defined settings.

To display the Views toolbar, select **View->Toolbars->Views** from the main menu; the Views toolbar is illustrated in the following figure:



The following custom views are available:

- **Front**, **Back**, **Top**, **Bottom**, **Left**, and **Right**: equivalent to observing the model from the six sides of a cube.
- **Iso**: an isometric view. This is the default orientation for three-dimensional models.
- **User1**, **User2**, **User3**, and **User4**: four user-defined views. See *Saving a user-defined view*, for a description of how to save a user-defined view.

If the current viewport is linked to other viewports, Abaqus/CAE also applies the same custom view to all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

1. If it is not visible already, display the Views toolbar by selecting **View->Toolbars->Views** from the main menu.

   Abaqus/CAE displays the Views toolbar.

2. From the Views toolbar, click the desired tool.

   If you have only one viewport, Abaqus/CAE immediately applies the selected view and unselects it from the Views toolbar. If you have more than one viewport, place the cursor over the viewport whose view you want to change. The cursor changes to a triad; click, and Abaqus/CAE applies the selected view to that viewport.

   Note:

   When you apply a view that was saved with the **Auto-fit** option selected, the view adopts the orientation of the saved view and immediately rescales it to fill the viewport. When you apply a view that was saved with the **Save current** option selected, the view adopts the orientation, zoom factor, and position of the saved view.

3. Repeat Step 2 as many times as necessary to achieve the desired view.



Tip: Use the *cycle* view manipulation tool to return to the previous view.

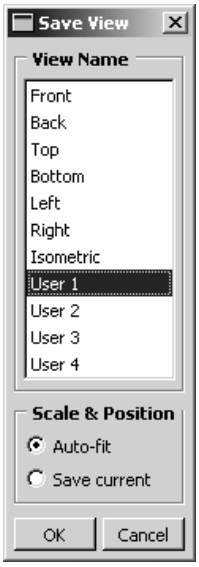
Additional information

- *Linking viewports for view manipulation*
- *Custom views*
- *Saving a user-defined view*
- *The view manipulation tools*
- *Using the view manipulation tools*

Saving a user-defined view

You can use the Views toolbar to open the **Save Views** dialog box and save a user-defined view.

Use the save tool  in the Views toolbar to open the **Save Views** dialog box and save a user-defined view. The **Save Views** dialog box is illustrated in the following figure:



Use the **Scale & Position** options to determine whether the saved view contains zoom factor and position information.

1. From the main menu bar, select **View->Save**.

    **Tip:** You can also save a view by clicking the  tool in the Views toolbar.

   If you have only one viewport, Abaqus/CAE immediately opens the **Save View** dialog box. If you have more than one viewport, click in the viewport whose view you want to save; Abaqus/CAE then opens the **Save View** dialog box.

2. From the **Save View** dialog box, choose the desired **Scale & Position** behavior:

   - Choose **Auto-fit** to save only the orientation of the view. When you apply a view saved with this option, the saved orientation is applied, but the scaling factor and position are adjusted to make the view fill the viewport.
   - Choose **Save current** to save the orientation, the zoom factor, and the position of the view. When you apply a view saved with this option, the saved orientation, scaling factor, and position are all applied.

3. In the **View Name** list in the **Save Views** dialog box, click the name of the tool you will use to recall this view.

   If you overwrite one of the six custom views—front, back, top, bottom, left, right—the other five views still retain their original definitions; that is, they do *not* become rotated to positions orthogonal to your saved view.

4. From the **Save View** dialog box, click **OK**.

Abaqus/CAE saves the definition of the view you selected. The view is saved only for the duration of the current session; the saved view will not be available the next time you run Abaqus/CAE.

Additional information

- *Linking viewports for view manipulation*
- *Custom views*
- *Applying custom views*
- *The view manipulation tools*
- *Using the view manipulation tools*

Applying a specified view

Select **View->Specify** from the main menu bar to specify a view. You can choose from the following methods to specify the view:

Rotation Angles

You can specify the angles through which Abaqus/CAE will rotate your view of the model about the model or screen 1-, 2-, and 3-axes. You can also choose to rotate your view of the model from an absolute position (a "Front" view) or from the current position.

Viewpoint

You can specify the coordinates of a vector along which an observer views your model. You can also orient the global 1-, 2-, and 3-axes within the viewport by specifying a vector representing the "up" direction.

Zoom

You can specify a zoom factor that expands or contracts the view. You can also choose to zoom the view relative to an absolute size of the objects in the viewport (the default size with a zoom factor of one applied) or relative to the current size of the objects in the viewport.

Pan

You can specify movement of the view in the 1- and 2-directions. The values correspond to fractions of the viewport horizontal and vertical dimensions and are relative to the current view.

For a more detailed explanation, see *Numerically specifying a view*.

If the current viewport is linked to other viewports, Abaqus/CAE also applies the specified view to all linked viewports in your session. For more information, see *Linking viewports for view manipulation*.

1. From the main menu bar, select **View->Specify**.

   Tip: You can also specify a view by double-clicking the 3D compass.

   Abaqus/CAE displays the **Specify View** dialog box.

2. From the **Specify View** dialog box, select the desired **Method** and do one of the following:

   - If you selected the **Rotation Angles** method, enter the rotation angles about the *X*-, *Y*-, and *Z*-axes (θ_x, θ_y, θ_z); a positive number corresponds to a counterclockwise rotation about each axis.

     Use the **Mode** button to specify how Abaqus/CAE is to apply your rotation:

     - Choose **Increment About Model Axes** to apply the rotation to the model axes of the current view.
     - Choose **Increment About Screen Axes** to apply the rotation to the screen axes of the current view. The screen *X*–axis is horizontal, the *Y*–axis is vertical, and the *Z*–axis is out of the screen. The origin of the screen axes is the center of the viewport.
     - Choose **Total Rotation From (0,0,1)** to first rotate the view to the default position (a view looking down the 3-axis with the 1- and 2-axes in the plane of the screen) and then apply the rotation.

- If you selected the **Viewpoint** method, enter the *X*-, *Y*-, and *Z*-coordinates of the viewpoint vector and the coordinates of the up vector.
- If you selected the **Zoom** method, enter the zoom factor and choose either **Absolute** or **Relative** magnification. A zoom factor greater than one expands your view of the model, and a zoom factor between zero and one contracts your view of the model.
- If you selected the **Pan** method, enter the values indicating how you want to position your view of the model relative to the current view. The values are fractions of the viewport dimensions; the first value represents horizontal motion, and the second value represents vertical motion.

3. Click **OK** to apply your specified view and to close the **Specify View** dialog box.



Tip: Use the *cycle* view manipulation tool to return to the original view.

Additional information

- *Linking viewports for view manipulation*
- *Numerically specifying a view*
- *The view manipulation tools*
- *Using the view manipulation tools*

Using the 3Dconnexion motion controllers with Abaqus/CAE

http://www.3dconnexion.com/ manufactures a variety of view manipulation devices that are popular with users of CAE and CAD systems. One example of their devices is the SpaceBall that is illustrated in *Figure 1*.

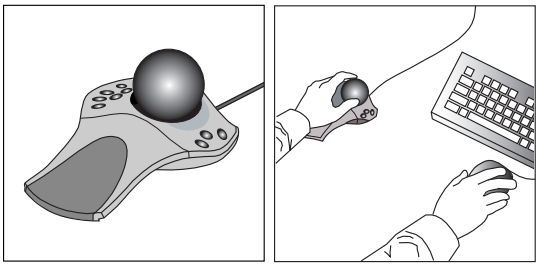


Figure 1: The 3Dconnexion SpaceBall.

You can use a 3Dconnexion motion controller together with the mouse to interact more efficiently with Abaqus/CAE. You can use the motion controllers to manipulate the view of your model with one hand while using the mouse to select from the model with the other hand. The motion controllers operate on whatever is under the cursor; for example, a part or a deformed plot in a viewport or a scrollbar in a dialog box. If neither a viewport nor a scrollbar is under the cursor, the motion controller operates on the current viewport.

You can change the center of rotation of an object if you use the motion controller in conjunction with the rotate view manipulation tool. By default, both the rotate tool and the motion controller rotate an object about the center of the viewport. However, you can select the center of rotation for both the rotate tool and the motion controller by positioning the cursor in the viewport at the location to be used as the center of rotation, clicking mouse button 3, and selecting **Set As Rotation Center**. Alternatively, if you click the rotate tool in the View Manipulation toolbar, a **Select** button appears in the prompt area. If you click on this button, you can select the center of rotation from a vertex or node in the viewport. The motion controller continues to use the specified rotation center even after you exit the rotate mode. For more information, see *The rotate view tool*.

In addition, Abaqus provides a set of "Application Functions" that are available through the standard 3Dconnexion user interface. These functions provide shortcuts to the view manipulation, view, and display tools. You can map these functions to the programmable buttons that are built into the 3Dconnexion motion controllers.

On Windows platforms Abaqus provides the following functions:

Movie Mode

Toggle between the default and alternate rotation modes. The default mode is comparable to rotating the camera around the camera target or selected center of rotation; the alternate mode rotates the camera about itself instead of about the camera target. You can also toggle on the alternate rotation mode if you hold down the **[Shift]** key while using the rotate view manipulation tool. For more information, see *Camera modes and view terminology*; *Using view options to control the camera*; and *The rotate view tool*.

Decrease Abaqus Sensitivity

Decrease the sensitivity of the 3Dconnexion motion controller. This setting applies only when manipulating the view in Abaqus/CAE.

Increase Abaqus Sensitivity

Increase the sensitivity of the 3Dconnexion motion controller. This setting applies only when manipulating the view in Abaqus/CAE.

Reset Abaqus Sensitivity

Restore the sensitivity of the 3Dconnexion motion controller to its default setting.

Auto-fit

Fit the model into the viewport. This is the same functionality as the auto-fit  view manipulation tool. For more information, see *The auto-fit tool*.

Keep in View

Keep the model in view during rotate and pan. When you are manipulating the view with the motion controller, toggling on this function prevents you from moving the target of the camera out of the viewport. Similarly, if you have changed the center of rotation, this option prevents you from moving the center out of the viewport.

Zoom to Cursor

Replace rotate and pan with zoom. If you press mouse button 2 and manipulate the motion controller, Abaqus/CAE disables the normal pan and rotate modes and replaces them with a mode that only zooms in or out from the area below the mouse cursor. If you release mouse button 2, the normal pan and rotate modes are restored. This behavior is enabled by default. You can use this function to toggle between the two modes.

Wireframe/Shaded

Toggle between wireframe and shaded render style. This is the same functionality as the wireframe  and shaded  icons located in the Render Style toolbar. For more information, see *Choosing a render style*.

Perspective

Toggle between perspective and parallel views. This is the same functionality as the perspective  and parallel  icons located in the View Options toolbar. For more information, see *Controlling perspective*.

Manipulate Layers

Manipulate all layers or the current layer. If you have created an overlay plot of results in the Visualization module, you can use this function to toggle between applying view manipulations to all the layers or to only the current layer. For more information, see *Manipulating the view for an overlay plot*.

Page Up

Page up in a scrollable dialog box. If the dialog box allows only horizontal scrolling, this button moves the page to the right.

Page Down

Page down in a scrollable dialog box. If the dialog box allows only horizontal scrolling, this button moves the page to the left.

Set Rotation Center

Set the center of rotation at the position of the mouse.

Clear Rotation Center

Clear a previously set center of rotation.

Set View Center

Center the view at the position of the mouse.

On Linux platforms Abaqus also provides a set of “Application Functions” that are available through the standard 3Dconnexion user interface. You should select Abaqus from the list of **Applications**. If Abaqus is not listed, you should select **XWindow Driver Version 2.0/3.0**.

The default mapping of the programmable buttons on Linux platforms is shown in the following table:

Button	Meaning
4	Movie mode
5	Decrease sensitivity
6	Increase sensitivity
7	Reset sensitivity
8	Auto-fit
9	Keep in view
A	Zoom to cursor ON/OFF
B	Wireframe/Shaded

The button mapping is fixed for the Abaqus application. However, 3Dconnexion allows you to reassign the buttons to use the standard functionality provided by their driver. For more information, see the Linux version of the 3Dconnexion documentation.

Selecting objects within the viewport

This chapter explains how to select objects that appear within a viewport, such as nodes, elements, vertices, edges, faces, and cells.

Selecting dialog box options is discussed in *Interacting with dialog boxes*. Selecting viewports is discussed in *Selecting viewports*.

In this section:

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*
- *Using the selection options*

Understanding selection within viewports

This section describes the objects that you can select in a viewport and explains what these objects represent.

In this section:

- *What objects can you select from the viewport?*
- *What is a selection group?*
- *Understanding the correspondence between geometric and physical objects*

What objects can you select from the viewport?

Selecting an object within the current viewport is one of the most common tasks you have to perform during the modeling process. In the course of various procedures you may need to select geometric objects (vertices, edges, faces, cells, and datums) or discrete objects (nodes and elements) by picking them directly from the viewport. *Figure 1* shows these different object types.

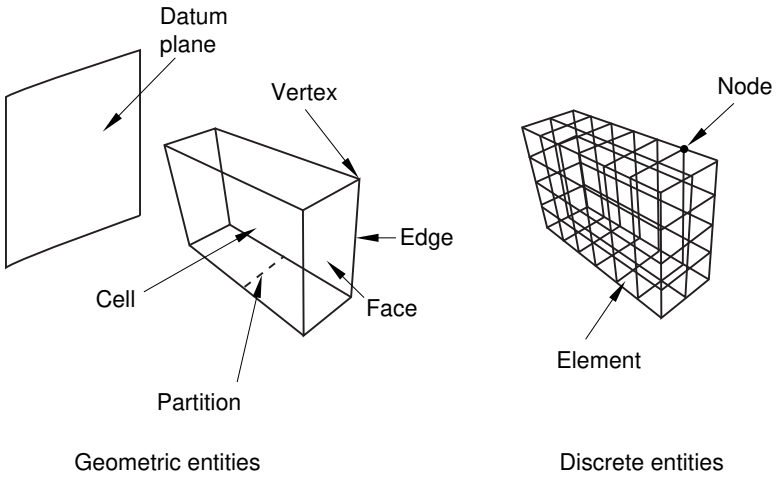


Figure 1: Object types that you can select.

You can select objects in the viewport during certain procedures, such as those listed below:

- Creating sets and surfaces
- Partitioning a part instance
- Editing a feature
- Seeding a part instance for meshing
- Creating or editing a display group composed of elements or nodes
- Color coding elements in your model
- Creating a node list path through your model
- Creating a load

You can also select objects in the viewport in advance of selecting a procedure. If you make selections prior to selecting a procedure, Abaqus/CAE does not limit your selections. When you select a procedure, Abaqus/CAE filters any selections that you made and keeps only those selections that are appropriate for the procedure. For more information, see *Selecting objects before choosing a procedure*.

If you select objects as part of a procedure, in most circumstances Abaqus/CAE only allows you to select objects that are appropriate for the current procedure. For example, the first step in partitioning an edge is selecting the edge of interest. Therefore, at this point in the procedure you can select only an edge; you cannot select a cell, a face, or a vertex. Messages in the prompt area guide you through the steps of a procedure and indicate which types of objects are available for selection. You can select only objects that are part of the current display group.

In some circumstances Abaqus/CAE cannot determine which objects are appropriate for selection and does not limit your selection. For example, when you are creating a set, you can select from cells, faces, edges, and vertices to include in the set and Abaqus/CAE allows you to select any of these objects. If you make an ambiguous selection from the viewport during a procedure, Abaqus/CAE allows you to cycle through the available objects until the desired object is selected. This ambiguity is described in *Cycling through valid selections*. You may find it easier to use the selection filters to limit the type of object you can select. For more information, see *Using the selection options*.

Many procedures to define attributes (interactions, constraints, loads, boundary conditions, predefined fields, and engineering features) allow you to select objects from the viewport to identify the region on which to apply the attribute. The default behavior for these procedures is to create a set or surface that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.

Additional information

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*

What is a selection group?

You can copy entities (vertices, edges, faces, or cells) that are highlighted in the viewport into a temporary storage area called a selection group. Abaqus/CAE saves the selection group for the duration of a session. Rather than manually reselecting the same entities during a subsequent selection procedure, you can paste the selection group into your selection. For example, you can copy all of the small faces highlighted by the **Geometry Diagnostics** query tool into a selection group. You can then paste the same selection group into your selection when you are using the Geometry Edit toolset to repair small faces.

When you paste a group into your selection, you can choose from selection groups and from display groups. Selection groups are designed to be a temporary convenience for the user, and they do not appear with display groups in the Display Groups toolset. You can create any number of display groups. In contrast, Abaqus/CAE saves a maximum of five selection groups. Abaqus/CAE overwrites the existing selection groups if you create more than five selection groups.

After you have selected the desired entities, you create a selection group by clicking mouse button 3 in the viewport and selecting **Copy** from the menu that appears. Abaqus/CAE copies all of the highlighted entities into a selection group. You paste a selection group to your current selection by clicking mouse button 3 in the viewport and selecting **Paste** from the menu that appears. Abaqus/CAE displays the **Paste to Selection** dialog box, and you can select one or more of the existing selection groups to paste to your current selection. When you paste a group, Abaqus/CAE appends the entities in the group to any other entities that you have already selected.



Note:

You cannot create or use selection groups if you are selecting objects in advance of selecting a procedure.

Understanding the correspondence between geometric and physical objects

When you select geometric objects in a viewport, it is important to understand what physical structure each object represents. The geometric objects that make up a model—cells, faces, edges, and vertices—can represent different physical structures depending on the space in which they are embedded.

For example, beams and other wire parts are represented by edges in the geometric model (see *Figure 1*).

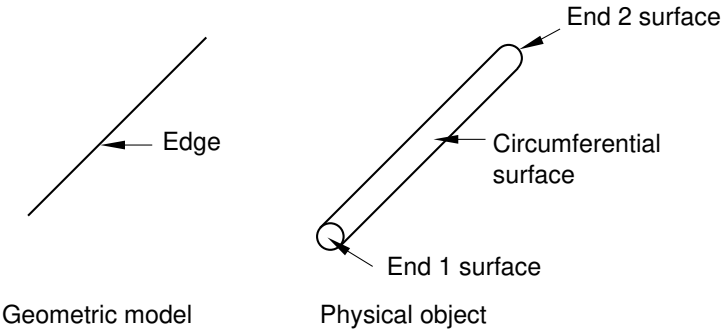


Figure 1: Selecting wire parts.

The end surfaces of these parts are represented by the vertices on either side of the edge, and the circumferential surface is represented by the line joining the vertices. To select a wire part, you can click the edge, and, if necessary, Abaqus/CAE prompts you to specify the surface of interest.

Likewise, axisymmetric shells are also represented by edges in the geometric model (see *Figure 2*).

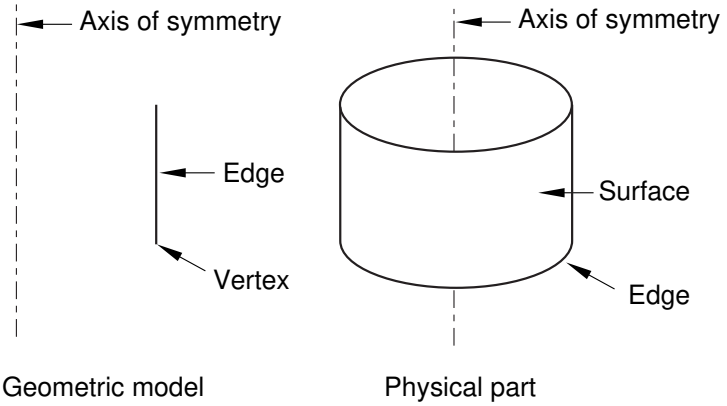


Figure 2: Selecting axisymmetric shells.

You can select the axisymmetric shell by clicking the edge in the viewport, and, if necessary, Abaqus/CAE prompts you to specify either the inside surface or the outside surface of the shell. You must select either the inside or the outside surface if you are applying a prescribed condition or contact definition to the surface. For example, if you want to apply a pressure load to a shell, you must specify which side of the shell should receive the load.

For more information on selecting surfaces, see *Specifying a particular side or end of a region*. For more information on modeling space, see *The relationship between parts and features*, and *Part modeling space*.

Additional information

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*

Selecting objects within the current viewport

This section describes techniques that you can use for selecting one or more objects in the current viewport.

In this section:

- *Selecting and unselecting individual objects*
- *Drag-selecting multiple objects*
- *Using the angle and feature edge method to select multiple objects*
- *Using the face curvature method to select multiple faces*
- *Using the topology method to select multiple elements*
- *Using the limiting angle, layer, and analytic methods to select multiple element faces*
- *Adding adjacent objects to a selection*
- *Combining selection techniques*
- *Excluding objects from your selection*
- *Cycling through valid selections*
- *Using groups while selecting entities*
- *Selecting interior surfaces*

Selecting and unselecting individual objects

Selecting and unselecting objects in the current viewport are straightforward operations that use standard methods. For more information on selecting viewports, see *Selecting viewports*.

Preselection highlighting allows you to preview which object Abaqus/CAE will select if you click at the current cursor location. In addition, preselection in the Sketcher uses a secondary cursor to indicate the exact position and type of entity that will be selected. (For more information on Sketcher preselection, see *The Sketcher cursors and preselection*.)

You will use the following three selection operations most frequently:

Click to select an object

To select a single object from the current viewport, move the cursor to the object and click mouse button 1.

- To select a point, click the corresponding point marker. The point marker changes color when selected. Vertices that you can select are marked by small, filled circles; and datum points are marked by small, unfilled circles. (See *Understanding the role of datum geometry*, for information on datum points.) Edge midpoints and arc centers that you can select are marked by small diamonds.

  Note:

  Some of the selection markers that appear when you are using the Sketch module are different from those described here. For information on selecting objects while using the Sketch module, see *The Sketcher cursors and preselection*.

- To select an edge, click the edge while positioning the cursor away from any vertex. Selected edges are highlighted.
- To select a face, click the face while positioning the cursor away from any edge or vertex. Selected faces are highlighted with a grid pattern. (The grid pattern is unrelated to mesh element location.)
- To select a cell, click any of its faces. All edges of selected cells are highlighted.

If you are unable to select the desired objects, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

Once you select an object, any objects previously selected in the current viewport are unselected automatically.

If your current procedure, options, and cursor position do not clearly specify one object for preselection, Abaqus/CAE highlights all of the potential selections and adds ellipsis marks (...) next to the cursor arrow to indicate an ambiguous preselection. If you accept an ambiguous preselection or otherwise make an ambiguous selection, use the buttons in the prompt area to make your final selection. For more information, see *Cycling through valid selections*.

[Shift] + Click to select additional objects

To select an additional object, move the cursor to the object and **[Shift] + Click**. Your original selection remains highlighted, and the newly selected object becomes highlighted.

An alternative method for selecting multiple objects is to drag a rectangle around the objects. For more information, see *Drag-selecting multiple objects*.

[Ctrl] + Click to unselect objects

To unselect an object, move the cursor to the object and **[Ctrl] + Click**. To unselect all objects, click an unused region of the current viewport.

When you have finished selecting and unselecting items in the viewport, click mouse button 2 to confirm your selection. You can use the selection option tools to adjust the shape of the drag-select region. You can also choose which objects are selected by the drag-select region. The selection option tools are located in the **Selection** toolbar. For more information, see *Modifying the shape of the drag-select region*, and *Choosing which objects are selected by the drag-select region*.

Additional information

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*
- *The Sketcher cursors and preselection*
- *Using the chain method to select edges in the Sketcher*

Drag-selecting multiple objects

Most prompts ask you to select just one object from the current viewport. However, some tasks allow you to select one or more objects; for example, the Set toolset allows you to select several objects of the same type and group them into sets. You can select multiple objects using the **[Shift] + Click** method described in *Selecting and unselecting individual objects*. An additional method for selecting multiple objects is to drag a rectangle around those objects. You can use the selection option tools to adjust the shape of the drag-select region. You can also choose which objects are selected by the drag-select region. The selection option tools are located in the **Selection** toolbar. For more information, see *Modifying the shape of the drag-select region*, and *Choosing which objects are selected by the drag-select region*.

1. Imagine a rectangle that encloses only the objects you want to select.
2. Click at one corner of the rectangle and, while continuing to press the mouse button, drag until you have enclosed all the objects.
3. Release the mouse button.

   All the valid objects inside or crossing the rectangle are highlighted.
4. Click mouse button 2 to indicate that you have finished selecting objects.

Sometimes it is convenient to use a combination of the **[Shift] + Click** and drag-select selection techniques. For more information, see *Combining selection techniques*.



Tip: If you select multiple objects and then want to unselect one or more of them, **[Ctrl] + Click** the objects you want to unselect. To unselect all the objects, click in an unused area of the viewport.

Additional information

- *Selecting objects within the current viewport*
- *Selecting and unselecting individual objects*
- *Cycling through valid selections*
- *Using the chain method to select edges in the Sketcher*

Using the angle and feature edge method to select multiple objects

In complicated models selecting individual faces or edges from geometry or selecting element faces or nodes from a mesh can be time consuming and prone to error. For example, when creating a surface from a mesh, you must select the individual element faces that make up the surface and append them to your selection. To speed up the selection process, Abaqus/CAE provides the angle and feature edge methods for selecting multiple faces, edges, elements, element faces, or nodes.

When you are performing a task in which you must pick more than one face or edge from geometry or more than one element, element face, or node from a mesh, Abaqus/CAE displays a field in the prompt area. The field allows you to choose between three selection methods—**individually**, **by angle**, and **by feature edge**, as shown in *Figure 1*.

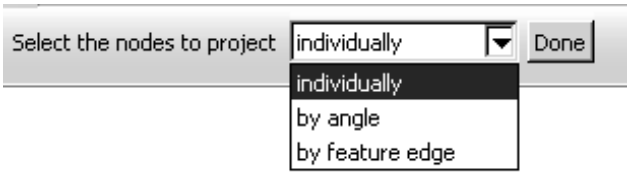


Figure 1: Choose the selection method from the field in the prompt area.

Individually

Selecting individual objects is described in *Selecting and unselecting individual objects*.

By angle

Selecting objects using the angle method is a two-step process:

1. In the prompt area, you enter an angle (from 0° to 90°).
2. From the part or assembly, you select a face, edge, element face, or node.

The angle must be greater than the angle through which adjacent edges or faces must rotate to create the geometry as if it was being formed by bending a straight wire or folding a series of faces. Abaqus/CAE starts from the selected geometry and selects all adjacent geometry until the angle you entered is met or exceeded.

For example, to select the edges of a regular hexagon, enter an angle greater than 60° (since each adjacent edge must be rotated 60° to form the shape from a straight wire), and select one of the edges. Abaqus/CAE then selects every adjacent edge since none of the angles is equal to or exceeds the angle that you entered.

Figure 2 illustrates how the angle method allows you to select all the elements around the flange of an exhaust manifold mesh.

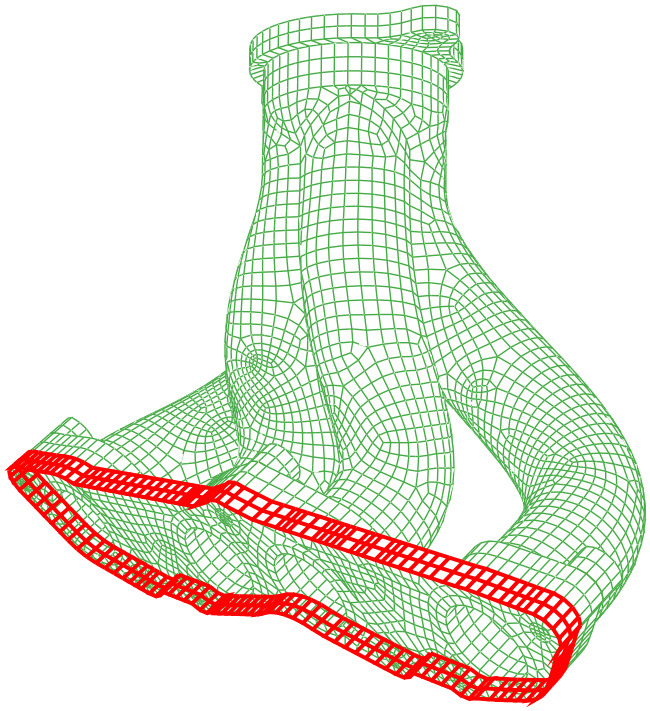


Figure 2: Enter an angle and select an element to select an entire face.

In the Sketch module, the angle method is available only when you are selecting objects from the underlying part or assembly. When you are selecting edges in the sketch, the chain method replaces the angle method. Use the chain method to select a group of edges that are connected end-to-end, like the links of a chain. For more information on the chain method, see *Using the chain method to select edges in the Sketcher*.

By feature edge

The feature edge method is also a multistep process:

1. In the prompt area, you enter an angle (from 0° to 90°).
2. Abaqus/CAE identifies all the feature edges in your model by finding all the element edges where the angle between two adjacent element faces is greater than the angle specified.
3. From the mesh, you select an element edge or node.
4. Abaqus/CAE follows the feature edge that passes through the selected element edge or node. The feature edge is truncated if another feature edge intersects it at an angle greater than the angle specified in Step 1.
5. Abaqus/CAE selects all the elements or nodes along the feature edge.

Figure 3 illustrates how the feature edge method allows you to select all the nodes along the edges of a flange of an exhaust manifold mesh.

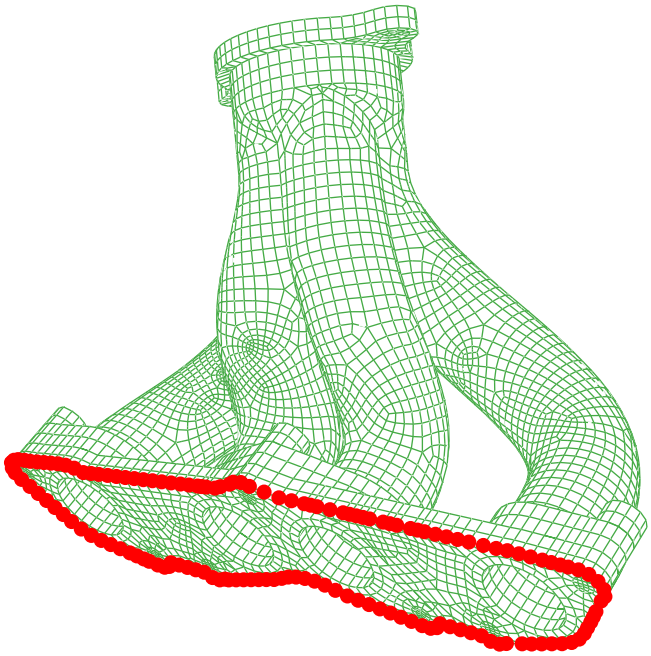


Figure 3: Enter an angle and select a segment of an edge to select adjacent nodes.

After you use the angle or feature edge methods, you can click the **individually** method in the prompt area and **[Shift] + Click** on individual faces, edges, elements, element faces, or nodes to append them to your selection. You can also **[Ctrl] + Click** on items to unselect them. In addition, you can continue to use the angle and feature edge methods and use **[Shift] + Click** to append faces, edges, elements, element faces, or nodes to your selection. You can keep the same angle, or you can change the angle while you continue to append items. For more information, see *Combining selection techniques*.

Additional information

- *Selecting objects within the current viewport*
- *Understanding selection within viewports*

Using the face curvature method to select multiple faces

In addition to selecting objects by the angle between them, you can select multiple faces from a part based on the curvature of the faces. When you are performing a task that allows you to pick more than one geometric face, Abaqus/CAE displays a field in the prompt area. The field allows you to choose between the three selection methods—**individually**, **by face angle**, and **by face curvature**, as shown in *Figure 1*.

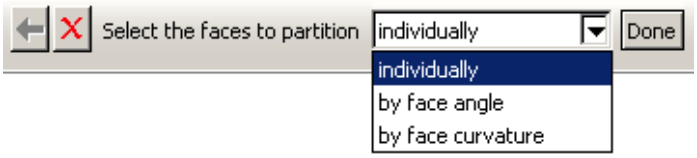


Figure 1: Choose the selection method from the field in the prompt area.

The angle selection method is described in *Using the angle and feature edge method to select multiple objects*.

The face curvature method is available during procedures that select faces. If a procedure accepts object types other than faces, you can change the object type in the **Selection** toolbar to **Faces** to access the face curvature method.

Select a face from the part or assembly. Abaqus/CAE selects all connected faces that have similar curvature along both principal directions and are joined at an angle of less than 20°. If you select a flat face, Abaqus/CAE adds any adjoining flat faces that lie in the same plane. Disconnected faces that share similar curvature are not selected, nor are faces that share similar curvature but have significantly different face normals at the edge where they meet. *Figure 2* shows two rounded faces selected using the face curvature method.

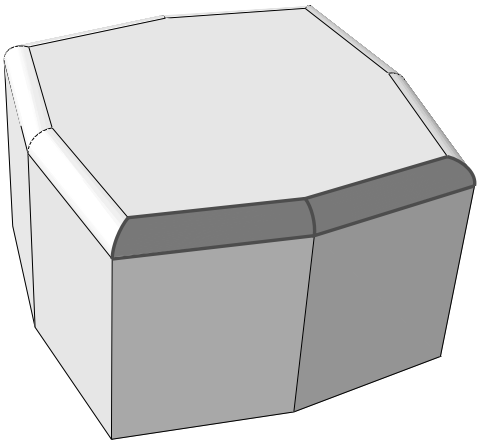


Figure 2: Select a single curved face to select adjoining faces with similar curvature.

After you use the face curvature method, you can click the **individually** method in the prompt area and **[Shift] + Click** on individual faces to append them to your selection. You can also **[Ctrl] + Click** on items to unselect them. In addition, you can continue to use the face curvature method and use **[Shift] + Click** to append faces to your selection. For more information, see *Combining selection techniques*.

Additional information

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*

Using the topology method to select multiple elements

You can select multiple elements based on the connection of a row or layer of elements. When you are performing a task that allows you to pick more than one element, Abaqus/CAE displays a field in the prompt area. The field allows you to choose between the four selection methods—**individually**, **by angle**, **by feature edge**, and **by topology**, as shown in *Figure 1*.

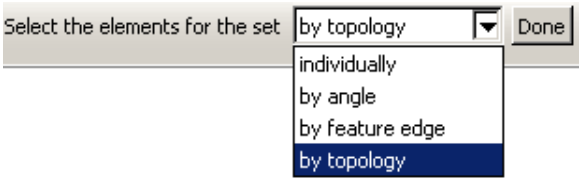


Figure 1: Choose the selection method from the field in the prompt area.

The angle and feature edge selection methods are described in *Using the angle and feature edge method to select multiple objects*.

The topology method is available during most procedures that select elements. If a procedure accepts object types other than elements, you can change the object type in the **Selection** toolbar to **Elements** to access the topology method.

The topology method is designed for use with two- and three-dimensional structured meshes. Select an element face from the mesh, and Abaqus/CAE selects all the elements connected to it in a row through the mesh. Select an element edge from the mesh, and Abaqus/CAE selects all the elements in a layer starting with the element faces that share the selected edge. *Figure 2* shows selection of an interior row on the left and an interior layer on the right.

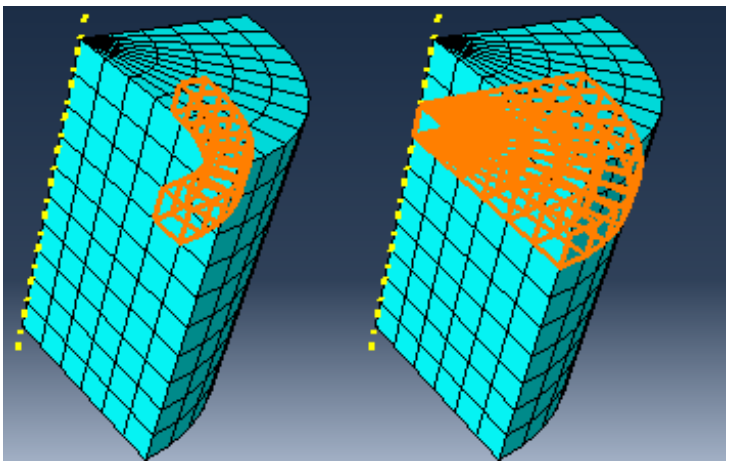


Figure 2: Using the topology method to select a row or a layer of elements.

You can use the topology method to select elements from other mesh types, but without the clearly defined rows or layers of a structured mesh, the selections may be unpredictable. In some cases, such as with a tetrahedral mesh, topology selection may be limited to only the elements that share the face or edge you select.

After you use the topology method, you can select other methods in the prompt area and **[Shift] + Click** to append more elements to your selection. You can also **[Ctrl] + Click** on items to unselect them. In addition, you can continue to use the topology method and use **[Shift] + Click** to append elements to your selection. For more information, see *Combining selection techniques*.

Additional information

- *Understanding selection within viewports*

- *Selecting objects within the current viewport*

Using the limiting angle, layer, and analytic methods to select multiple element faces

When you are selecting orphan element faces to create geometry (for more information, see *Create face from element faces*), Abaqus/CAE displays a field in the prompt area. The field allows you to choose between five selection methods—**individually**, **by angle**, **by limiting angle**, **by layer**, and **by analytic**, as shown in *Figure 1*.

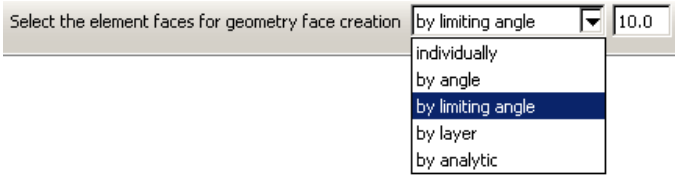


Figure 1: Choose the selection method from the field in the prompt area.

The angle selection method is described in *Using the angle and feature edge method to select multiple objects*. The limiting angle, layer, and analytic methods are available only while selecting orphan element faces to create new geometric faces.

By limiting angle

Selecting objects using a limiting angle is a two-step process:

1. In the prompt area, you enter an angle (from 0° to 90°).
2. From the part or assembly, you select an orphan element face.

The angle must be greater than the total angle between the selected element face and the element faces connected to it. Abaqus/CAE starts from the selected geometry and selects all adjacent geometry until the angle between the selected face and the last face in the series of adjacent faces meets or exceeds the angle you entered. *Figure 2* shows selection of element faces with a limiting angle of 45°, and one of the vertical element faces below the rounded area is picked.

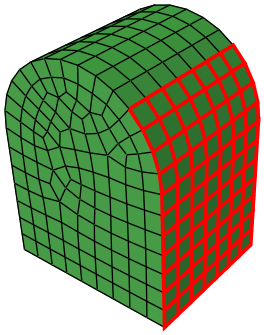


Figure 2: A limiting angle of 45° with a selected vertical face.

Increasing the limiting angle to its maximum of 90° would select the faces up to the top of the rounded section. In contrast, using the angle method with an angle of 13° or more would continue the selection around the rounded portion and down the far side since the angle between each adjacent face is less than 13°.

By layer

Selecting objects using the layer method is a two-step process:

1. In the prompt area, you enter a number of layers.
2. From the part or assembly, you select an orphan element face.

Abaqus/CAE starts from the selected face and selects layers of adjacent element faces around it in all directions. Selection continues around corners and other features until the number of layers is reached or until there are no more adjacent orphan element faces.

Figure 3 illustrates the selection of three layers of orphan shell element faces around a starting face and the resulting geometric face.

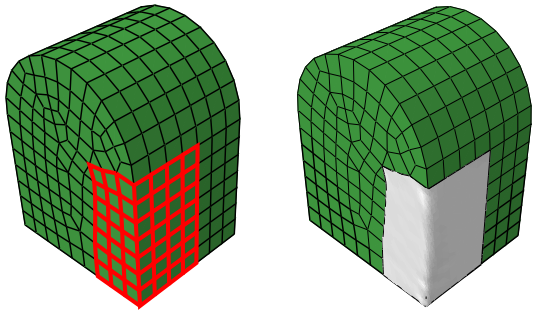


Figure 3: Face layer selection and creation of a geometric face.

As shown in *Figure 3*, layer selection can traverse sharp corners and other model features that would normally signify the end of a geometric face. In most cases you should preserve logical model edges and other features by creating separate faces. Otherwise, the resulting geometry may be difficult to repair and mesh.

Note:

When you are working with solid orphan elements, selections that include multiple faces from the same orphan element are not acceptable for the creation of a single geometric face.

By analytic

The analytic selection method for orphan element faces is based on the recognition of basic shapes in analytic geometry (such as planes, cylinders, cones, spheres, and tori), or portions of these shapes. Analytic selection attempts to recognize the logical boundaries of a set of orphan element faces that would make a recognizable geometric face.

Figure 4 illustrates analytic selection of orphan element faces. A spherical section of element faces is highlighted; this selection could not be made using any of the other selection options for multiple objects.

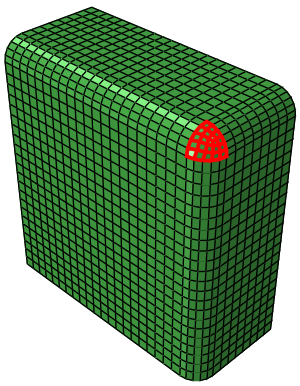


Figure 4: Analytic geometry selection.

After you use any of the above methods, you can select other methods in the prompt area and **[Shift] + Click** to append more elements to your selection. You can also **[Ctrl] + Click** on items to unselect them. In addition, you can continue to use the current method and use **[Shift] + Click** to append elements to your selection. For more information, see *Combining selection techniques*.

Additional information

- *Understanding selection within viewports*
- *Selecting objects within the current viewport*
- *Create face from element faces*

Adding adjacent objects to a selection

If you have already selected one or more objects, you can expand your selection to include all adjacent objects of the same type. Adding adjacent objects is an alternative to using drag-select or the angle method (see *Drag-selecting multiple objects*, and *Using the angle and feature edge method to select multiple objects*, respectively) to quickly select multiple objects. Selecting adjacent objects allows you to expand your selection in all directions, regardless of the shape of surrounding features or the angle at which objects are joined. It also allows you to pick multiple areas of interest in a model and expand the selection set in each area at the same time.

To add adjacent objects to the current selection, click mouse button 3 over an existing selected object and select **Add Adjacent Entities**. Abaqus/CAE expands your selections to all adjacent objects of the same type, including objects that are not included in the current display. If necessary, Abaqus/CAE adds the newly selected objects to the current display group to make them visible. Adjacent objects are defined in terms of the currently selected entities as follows:

- Edges that share a common vertex with one or more selected edges
- Vertices that share a common edge with one or more selected vertices
- Faces that share a common vertex or edge with one or more selected faces
- Nodes that share a common edge with one or more selected nodes
- Elements that share a common element edge or node with a selected element

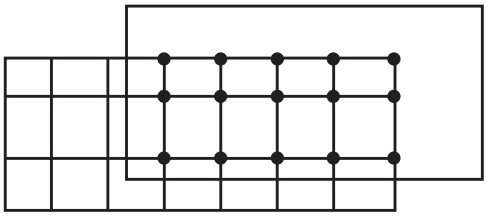
Additional information

- *Selecting objects within the current viewport*
- *Using the chain method to select edges in the Sketcher*

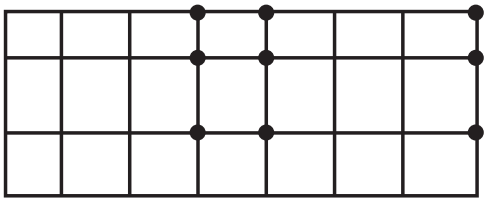
Combining selection techniques

There are times when it is convenient to use a combination of the methods for selecting and unselecting objects. For example, you can drag-select a group of nodes while creating a node set using the Set toolset. You can then **[Ctrl] + Click** individual nodes to unselect them and **[Shift] + Click** additional nodes to add them to your selection. A combination of the three techniques is illustrated below:

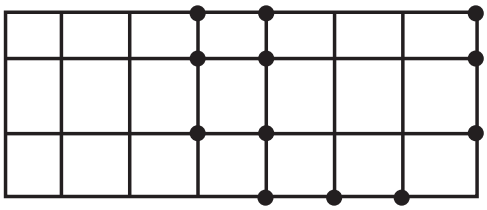
1. First, you use drag-select to select a group of nodes.



2. Then, you use **[Ctrl] + Click** to unselect individual nodes.



3. Finally, you use **[Shift] + Click** to add nodes to your set and then click mouse button 2 to indicate you have finished selecting.



You may find it useful to adjust the view orientation to make particular items in the viewport more accessible. You can adjust the view orientation at any point during the selection process. For information on the view manipulation tools, see *Manipulating the view and controlling perspective*.

Tip: To unselect all the objects, click an unused part of the current viewport.

Additional information

- *Selecting objects within the current viewport*
- *Selecting and unselecting individual objects*
- *Drag-selecting multiple objects*
- *Using the chain method to select edges in the Sketcher*

Excluding objects from your selection

When you select an object from the viewport, your selection includes all the entities of lower dimensionality that are associated with the object. For example, if you select a cell, your selection includes all the faces, edges, and vertices associated with the cell. Similarly, if you select an edge, your selection includes all the vertices associated with the edge. In some circumstances you may want to exclude the entities of lower dimensionality from your selection. For example, if you select an edge to include in a set, you may not want the set to contain the vertices at each end of the edge. Excluding entities of lower dimensionality from your selection may solve any problems that you encounter with overconstraints.

1. Select all the objects using a combination of select, drag-select, **[Ctrl] + Click**, and **[Shift] + Click**.
   Abaqus/CAE highlights the selected objects in red.
2. **[Ctrl] + Click** an object to exclude it from your selection.
   Abaqus/CAE highlights the excluded objects in purple.

Additional information

- *Selecting objects within the current viewport*
- *Selecting and unselecting individual objects*
- *Cycling through valid selections*
- *Using the chain method to select edges in the Sketcher*

Cycling through valid selections

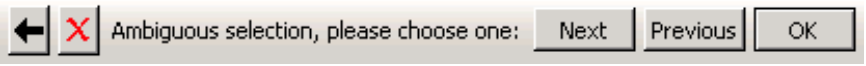
In some cases Abaqus/CAE is unable to differentiate between the object you have selected and other nearby or related objects. This ambiguity can arise as follows:

- Imagine a small square surrounding the cursor. When you click an object, any other valid objects of the same type that fall inside this square are also considered to be possible selections. For example, if you select an edge that is positioned very close to another edge, Abaqus/CAE may consider both edges to be possible selections.

  The size of the square is independent of the monitor size, the viewport size, and the dimensions of the model. It also remains constant when you zoom in and out on your model. Therefore, you can select a specific object in the viewport more precisely by zooming in on your model to increase the distance between objects.
- If your model is three-dimensional, imagine a line that is perpendicular to the screen and that passes through the cursor and into the model. When you select an object, any valid objects of the same type that intersect this line are considered to be possible selections. (Rotating your model may remove some of the ambiguity.)

Abaqus/CAE reduces the potential for ambiguity by filtering your selection against the current procedure whenever possible. For example, if you are partitioning a cell, Abaqus/CAE prompts you to select the cell to partition. When you make a selection, Abaqus/CAE considers only cells to be a valid selection. Conversely, if you are creating a geometry set, Abaqus/CAE considers cells, faces, edges, and vertices to be a valid selection and the potential for ambiguity is increased. In addition, preselection highlighting allows you to see exactly which object would be selected before you make the selection. Moving the cursor around the viewport may remove the ambiguity in the selection and result in Abaqus/CAE highlighting the object of your choice. If the ambiguity remains, Abaqus/CAE changes the cursor, adding ellipsis marks (...) to the right of the arrow, and highlights all the possible selections.

When your selection is ambiguous, Abaqus/CAE displays buttons in the prompt area that allow you to cycle through all of the possible selections, as shown here:



Use the **Next** and **Previous** buttons to cycle forward and backward through all of the objects in the viewport that are possible selections; each object becomes highlighted in turn. When the object of your choice is highlighted, click **OK** or click mouse button 2 to confirm your selection. (You can also click mouse button 3 in the current viewport to reveal a menu of the options in the prompt area.)

Additional information

• *Selecting objects within the current viewport*

Using groups while selecting entities

You can append groups of entities to your selection to speed up the process of selecting many entities from the viewport. The group can be a display group, or it can be a temporary selection group. You can click mouse button 3 on the viewport and do the following:

- Create a selection group by copying entities (vertices, edges, faces, or cells) that are highlighted in the viewport into a selection group.
- Append to your selected entities by pasting the entities stored in a selection group or a display group into your current selection.

Selecting interior surfaces

You can use the selection tools to select an interior surface of a model; for example, when you create a surface or when you select a region using the solid offset mesh tool.

1. In the **Selection** toolbar, toggle on the **Select From Interior Entities** tool .

   Note:

   The **Select From Interior Entities** tool is hidden by default. For more information, see *Using toolboxes and toolbars that contain hidden icons*.

2. Select the interior surface from the viewport.

Additional information

- *Understanding and using toolboxes and toolbars*
- *Understanding selection within viewports*
- *Selecting objects within the current viewport*
- *Filtering your selection based on the position of the object*
- *What is a surface?*
- *Specifying a particular side or end of a region*

Using the selection options

Abaqus/CAE provides a set of tools that can make it easier and more efficient for you to select entities from the viewport.
The selection tools are located in the **Selection** toolbar. The available options depend on the current selection procedure; some options can be used to preselect entities outside of a procedure.

This section describes the selection options.

In this section:

- *Available selection options*
- *Filtering your selection based on the type of object*
- *Filtering your selection based on the position of the object*
- *Highlighting objects prior to selection*
- *Modifying the shape of the drag-select region*
- *Choosing which objects are selected by the drag-select region*
- *Selecting objects before choosing a procedure*

Available selection options

When you are prompted to select an object from the viewport, Abaqus/CAE provides selection tools that can make it easier and more efficient for you to make the desired selection.

Use the **Selection** toolbar to configure the selection options. *Figure 1* shows the layout of the selection tools. Selection tools appear dimmed if they are not valid for the current procedure.



Figure 1: The Selection toolbar.

Additional information

- *Components of the toolbars*
- *Using dimmed dialog box and toolbox components*
- *Selecting objects within the current viewport*
- *Using the selection options*

Filtering your selection based on the type of object

To help you select the desired entities (such as vertices, edges, faces, nodes, and elements) from the current viewport, Abaqus/CAE provides a set of filters that you can use to limit your selection based on the type of object.
For example, if you are creating a set that contains only surfaces, you can limit your selection to only faces—other objects, such as vertices and edges, will not be selected.

The object filters are listed in the **Selection** toolbar. Abaqus/CAE configures the filter list based on the current procedure. If you have not started a selection procedure, Abaqus/CAE lists some commonly used filters (for more information on selecting objects outside of a procedure, see *Selecting objects before choosing a procedure*).

You can press **[Ctrl] + A** to select all objects in the current viewport based on the active selection filter. The view is rescaled to fit the viewport to clearly display the selected objects, which are highlighted in the viewport. For example, if edge objects are selected in the filter and you press **[Ctrl] + A**, all edges visible in the viewport are selected. You cannot customize the **[Ctrl] + A** keyboard shortcut.

If the current viewport contains an Abaqus/CAE part or part instance, you can select one of the following filters:

All

All objects except skins and stringers.

Cells

All volumes, such as cells and elements.

Faces

All planar objects, such as faces, datum planes, and element faces.

Edges

All edge objects, such as edges, datum axes, and element edges.

Vertices

All point objects, such as vertices, datum points, and nodes.

Elements

All elements.

Nodes

All nodes.

Datums

All datum objects, such as datum planes, datum axes, and datum points.

Features

All feature objects.

Ref points

All reference points.

The list of available filters is updated as you change modules or selection procedures, and additional filters (such as **Instances**, **Skins**, and **Stringers**) are displayed where applicable. By default, Abaqus/CAE selects from all objects but excludes skins and stringers. You can select a skin or stringer from the viewport only after you select the appropriate filter.

Similarly, if you are selecting orphan elements in the current viewport (to assign an element type, for example), you can select one of the following filters:

- All
- Zero-dimensional elements
- One-dimensional elements
- Two-dimensional elements
- Three-dimensional elements
- Skins
- Stringers

By default, Abaqus/CAE selects from all elements except skins and stringers.

• *Using the selection options*

Filtering your selection based on the position of the object

The selection tools allow you to choose from which objects to select, based on their positions in the viewport. The **Selection** toolbar contains all of the selection tools.

The following position-based selection tools are available only after you start a procedure that requires object selection (position-based selection is not available outside of a procedure):

Objects closest to the screen

 Toggle on this tool to select only the objects closest to the front of the screen. This tool is toggled on by default.

If you toggle off this tool, Abaqus/CAE allows you to cycle through all of the possible selections. Use the **Next** and **Previous** buttons in the prompt area to cycle forward and backward through all of the objects in the viewport that are possible selections; each object becomes highlighted in turn. For more information, see *Cycling through valid selections*.

This filter applies to native vertices, edges, faces, and cells and to orphan nodes and orphan elements.

Interior and exterior objects

Choose one of the following filters:

-  Select objects located both outside and inside a part.
-  Select only objects located on the outside of a part. In most cases this tool is selected by default.
-  Select only objects located on the inside of a part.

• *Using the selection options*

Highlighting objects prior to selection

When you are selecting objects from the viewport and you stop moving the cursor, Abaqus/CAE highlights the object that would be selected at the cursor position.
This behavior, called “preselection,” allows you to see exactly which object would be selected before you make the selection. If the current selection options make more than one object available, Abaqus/CAE changes the cursor, adding ellipsis marks (...) to the right of the arrow, and highlights all the possible selections. The position and type filtering that you choose for selection also applies to preselection.

Toggle off the **Allow Preselection During Picking** tool  in the **Selection** toolbar to turn off preselection highlighting for procedures in the current session. This tool is toggled on by default.



Note:

Preselection highlighting may be delayed for large models; toggling off preselection may improve the display speed.

• *Using the selection options*

Modifying the shape of the drag-select region

The selection tools allow you to change the shape of the drag-select region.

From the **Selection** toolbar, choose one of the following:

Rectangle

 Click to indicate one corner of the rectangle, and drag the cursor to the second corner. This tool is selected by default.

Circle

 Click to indicate the center of the circle, and drag the cursor to a point on the circumference.

Polygon

 Click to indicate one vertex of the polygon, and drag the cursor to the second vertex. You then continue to click on each vertex of the polygon. Click mouse button 2 to indicate you have finished entering vertices. There is no limit to the number of vertices in the polygon.

• *Using the selection options*

Choosing which objects are selected by the drag-select region

The selection tools allow you to choose which objects are selected by the drag-select region.

From the **Selection** toolbar, choose one of the following:

Inside

 Select only the objects that fall inside the drag-select region.

Inside and crossing

 Select only the objects that fall inside or cross the drag-select region. This tool is selected by default.

Crossing

 Select only the objects that cross the drag-select region.

Outside and crossing

 Select only the objects that fall outside or cross the drag-select region.

Outside

 Select only the objects that fall outside the drag-select region.

• *Using the selection options*

Selecting objects before choosing a procedure

You can select objects from the current viewport before choosing a procedure to work with them.

You can use the **Selection** toolbar 

to toggle *selection* and to limit viewport selections based on the type of object. The **Selection** tool is the first tool in the toolbar; it is available only when there are no active procedures running in a viewport. By default, selection is active and viewport object types are not limited. You can select multiple objects using any of the methods described in *Combining selection techniques*. Toggle off object selection to prevent selecting objects when you are not in a procedure.

Note:

Preselection highlighting may be delayed for large models; toggling off selection may improve the display speed by preventing preselection and selection unless you first choose a procedure that requires object selection.

When you select a procedure after selecting objects from the viewport, Abaqus/CAE applies the selection filters for the procedure. For example, if you selected a vertex, a face, and an edge, then started a procedure that can accept only vertices, Abaqus/CAE accepts the selected vertex, cancels the selection of the face and edge, and begins the procedure with the second step. Similarly, if you select a procedure that requires a single object selection and you have already selected multiple valid objects, Abaqus/CAE accepts the first valid selection and cancels the remaining selections. If Abaqus/CAE cannot determine which object you selected first, it will cancel all selections and begin the procedure at the first step.

If a procedure includes multiple selection steps, objects that you select before the procedure can be used only to complete the first selection step. Any subsequent selection steps require you to select new objects interactively from the viewport or, if applicable, use saved selection groups. You cannot save selections made prior to the start of a procedure.

- *Using the selection options*

Configuring graphics display options

This chapter explains how you can configure the graphics display options in Abaqus/CAE.

In this section:

- *Using graphics display options*
- *Using display lists*
- *Using antialiasing*
- *Choosing a highlight method*
- *Choosing a translucency mode*
- *Controlling drag mode*
- *Choosing background colors*

Using graphics display options

When you start a session, Abaqus detects the graphics hardware installed on your system and sets the graphics options accordingly. If your graphics hardware is not supported by Abaqus/CAE or if you wish to override the default graphics options, you can use the **Graphics Options** dialog box to tune display performance.

Abaqus/CAE applies the settings to all viewports and saves the settings for the duration of the session. To use the customized settings each time you start an Abaqus/CAE session, modify the environment file (`abaqus_v6.env`). For additional information on the environment file, see the *Abaqus Configuration Guide*.



Note: Recommended settings for recently introduced graphics adapters are available from the **Support** page at *www.3ds.com/simulia*.

You can also use the **Graphics Options** dialog box to do the following:

- Choose the appearance of your model during rotation, pan, or zoom view manipulations. The appearance is related to the render style and can be set to **Fast (wireframe)** or **As is**.
- Choose whether Abaqus/CAE will auto-fit the image to the current viewport after you rotate the view. Automatically fitting the image to the viewport is equivalent to clicking the auto-fit tool  in the View Manipulation toolbar. Auto-fit adjusts your view of the model so that the model fills the viewport and is centered within it. The orientation remains fixed, as indicated by the view triad.
- Choose whether Abaqus/CAE will optimize the display of translucent objects for performance, for accuracy, or for a level in between.
- Choose the viewport background color. Your selected color will be applied to all viewports in the current session of Abaqus/CAE.

1. From the main menu bar, select **View->Graphics Options**.

   The **Graphics Options** dialog box appears.

2. Select one of the following options:

   - Tune performance using options for display lists, highlight method, and translucency mode.
   - Choose the display mode while you drag objects in the viewport.
   - Enable or disable the automatic fitting of your view to the viewport after rotations.
   - Choose the background color of the viewports.

Additional information

- *Using display lists*
- *Choosing a highlight method*
- *Choosing a translucency mode*
- *Controlling drag mode*
- *Choosing background colors*

Using display lists

Display lists help you display repeated images faster. When display lists are enabled, every drawing operation is recorded in a list that can be quickly replayed if necessary. This results in faster image refreshes on most systems but requires graphics memory to record each drawing operation. Display lists are not used at all in the Visualization module (Abaqus/Viewer). In Abaqus/CAE display lists are usually enabled; you should disable them if you experience memory problems or a degradation of graphics performance while displaying exceptionally large models.

1. Locate the display lists options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.

2. From the **Hardware** options, toggle **Use display lists** on or off to control the use of display lists for images. This option has no effect in the Visualization module.

   When **Use display lists** is on, you may notice a brief delay the first time an image is drawn; this occurs because Abaqus/CAE must construct the display lists. Subsequent drawing of the image is faster.

3. Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Choosing a highlight method*
- *Choosing a translucency mode*
- *Controlling drag mode*
- *Choosing background colors*

Using antialiasing

Abaqus/CAE uses antialiasing to improve the display of curved and diagonal lines on models. Antialiasing is enabled on compliant systems, and you can toggle off this option to improve performance for some systems.

1. Locate the graphics options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.
2. From the **Hardware** options, toggle on **Anti-alias lines** to enable antialiasing or toggle it off to disable antialiasing.
3. Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Using display lists*
- *Controlling drag mode*
- *Choosing background colors*

Choosing a highlight method

The highlight method controls how Abaqus/CAE displays highlighting in the viewport while you interact with the model. The effect of changing the highlight method is most apparent when you use the view manipulation tools to rotate the model or while sketching a profile in the Sketch module. **Hardware Overlay** provides the best graphics performance; however, this option is not supported by all graphics adapters.

In most circumstances you should not have to change the default highlight method setting. When you start a session, Abaqus/CAE detects the graphics hardware installed on your system and selects the appropriate highlight method. However, if you are using a graphics adapter that is not supported, the default setting selected by Abaqus/CAE may not be optimal.

1. Locate the highlight method options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.

2. From the **Highlight method** menu, select one of the following:

   - **Hardware Overlay**. This option appears only if it is supported by the graphics adapter on your system. It uses graphics hardware to display the highlighting layer. If your workstation supports this option, choosing **Hardware Overlay** provides the optimum quality and performance.
   - **XOR**. This option uses software to emulate the highlighting layer. Choosing the **XOR** option uses a Boolean pixel operation to simulate the drawing operations but can produce different colors depending on the color of the underlying pixels.
   - **Software Overlay**. This option also uses software to emulate the highlighting layer and provides good quality at reasonable performance. However, on some systems selecting this option results in poor performance.
   - **Blend**. The Blend method combines the color of the underlying pixel with the desired color producing an approximation of the transient graphics. You should choose this option only if none of the other options are satisfactory. Performance and quality may be affected if you choose the **Blend** option.

3. Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Using display lists*
- *Choosing a translucency mode*
- *Controlling drag mode*
- *Choosing background colors*

Choosing a translucency mode

The translucency mode controls the speed and accuracy with which Abaqus/CAE displays translucent objects in the viewport. The effect of changing the translucency mode is most apparent when you use the view manipulation tools to manipulate the display of the model. The **Fast** setting provides the best graphics performance, while the **Accurate** setting provides the best rendering of translucent objects.

In most circumstances you should not have to change the default translucency method setting. When you start a session, Abaqus/CAE detects the graphics hardware installed in your system and selects the appropriate translucency mode. However, if you are using a less capable graphics adapter, the default setting selected by Abaqus/CAE may not be optimal for your models.

1. Locate the translucency mode options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.

2. Drag the **Translucency mode** slider to the setting that you want. For some graphics cards, Abaqus/CAE supports three intermediate translucency mode settings between **Fast** and **Accurate**; for others, Abaqus/CAE supports only one intermediate setting between these two options.
3. Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Using display lists*
- *Choosing a highlight method*
- *Controlling drag mode*
- *Choosing background colors*

Controlling drag mode

Drag mode controls the appearance of your model during rotation, pan, or zoom view manipulations. Drag mode can be set to **Fast (wireframe)** or **As is**.

Fast (wireframe)

When you set drag mode to **Fast (wireframe)**, a wireframe outline is drawn during the view manipulation. **Fast (wireframe)** is the default drag mode setting on compliant systems.

As is

When you set drag mode to **As is**, everything displayed in the window will continue to be displayed during the view manipulation. On older or slower systems the display may lag behind movement of the mouse while you manipulate the view, especially if the model is complex. For newer systems with graphics hardware acceleration the **As is** setting can be accommodated without significant loss of performance.

Set the drag mode to **As is** to observe the model during the view manipulation; for example, to locate areas of high stress concentration as you rotate a contour plot.

In addition, you can choose whether Abaqus/CAE will auto-fit the image to the current viewport after you rotate the view. Automatically fitting the image to the viewport is equivalent to clicking the auto-fit tool  in the View Manipulation toolbar. Auto-fit adjusts your view of the model so that the model fills the viewport and is centered within it. The orientation remains fixed, as indicated by the view triad.

1. Locate the drag mode options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.
2. From the **View Manipulation** field, select **Fast (wireframe)** or **As is** drag mode.
3. If desired, toggle on **Auto-fit after rotations**.
4. Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Using display lists*
- *Choosing a highlight method*
- *Choosing a translucency mode*
- *Choosing background colors*

Choosing background colors

The appearance of your models is affected by the difference in contrast between the model colors and the colors in the viewport background. You can improve this contrast by changing the color or colors displayed in the viewport background. Abaqus/CAE provides two color selections: you can display a single color that floods the entire viewport background, or you can create a gradient background that blends two different colors.

Abaqus/CAE implements your background color selections for every viewport in your session.

1. Locate the background color options.

   From the main menu bar, select **View->Graphics Options**.

   Abaqus/CAE displays the **Graphics Options** dialog box.

2. From the **Viewport Background** field, select one of the following:

   - **Solid** to choose one color for the viewport background
   - **Gradient** to choose the top and bottom colors used to draw the background as a gradual blend from the top of the viewport to the bottom of the viewport

3. Select the viewport background color or colors.

   a. Click a color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

   The color sample changes to the selected color.

   d. For gradient backgrounds you can select both the top and bottom colors directly; or you can select the top color, then click **Auto-Select** to have Abaqus/CAE select the bottom color based on a variation of the top color.

   e. Sketches or X–Y plots may not display clearly with a gradient background. For these views Abaqus/CAE displays a solid background even when you have specified a gradient background. To disable this behavior, toggle off **Allow override**.

4. Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE implements your background color selections for every viewport in your session. Your changes are saved for the duration of the session.

Additional information

- *Using display lists*
- *Choosing a highlight method*
- *Choosing a translucency mode*
- *Controlling drag mode*

Printing viewports

This chapter describes how you send an image of selected viewports either directly to a printer or to a file.

For more information on configuring printers, see the *Abaqus Configuration Guide*.

In this section:

- *Understanding printing*
- *Controlling the destination and appearance of printed images*

Understanding printing

Abaqus/CAE allows you to take a snapshot of one or more viewports and their contents and to send the image either directly to a printer or to a file for later use; for example, to include in a presentation, embed in a printed report, or display in an HTML document.
Additional options allow you to select the appearance of viewports in the resulting image, as well as the color, resolution, and size of the image.

This section describes basic concepts you should understand before sending output to a printer or to a file.

In this section:

- *Printed image formats*
- *Windows and PostScript image layout*
- *Windows printer image size*
- *EPS, TIFF, PNG, and SVG image size*
- *Hard-copy image quality*
- *Importing Abaqus/CAE images into other software products*

Printed image formats

You can print images directly to a printer or save images in several file formats.

Abaqus/CAE allows you to print images directly to a Windows printer. The printer driver creates and sends the necessary information to the printer in whatever format is required.

If Windows printer drivers are not available or if you are using another platform, you can use a print command to create and send a PostScript file directly to a PostScript printer. You can also save images in a Portable Network Graphics (PNG), Scalable Vector Graphics (SVG), Tag Image File Format (TIFF), PostScript (PS), or Encapsulated PostScript (EPS) file. The following list describes these file formats:

PNG

Portable Network Graphics (PNG) is an industry standard for storing raster images. The use of PNG files has been popularized by the World Wide Web, and PNG images are displayed by most popular web browsers running on a variety of operating systems. A PNG file consists of color information and a compressed raster representation of the image. By default, Abaqus/CAE limits PNG images of viewports to 8-bit color (256 colors). However, you can also use 24-bit color (on Windows) or the system color setting (on Linux).

SVG

Scalable Vector Graphics (SVG) is an industry-standard vector graphics language written in XML.

TIFF

Tag Image File Format (TIFF) is a well-established raster image format that is recognized by many software applications. The TIFF format supports both color and greyscale. By default, Abaqus/CAE limits TIFF images of viewports to 8-bit color (256 colors). However, you can also use 24-bit color (on Windows) or the system color setting (on Linux).

PostScript

PostScript is the recognized standard for desktop publishing. PostScript is actually a programming language whose instructions and data are usually stored in an ASCII format that can be transferred easily between operating systems. The PostScript format is used when you use a print command to print to a PostScript printer or when you save the image in a PostScript file. When you select the PostScript format, Abaqus/CAE generates either a compressed raster representation or a vector representation of your image. For efficiency when producing raster images, you should minimize the size of your image and limit the resolution of the image to, at most, the resolution of the device on which the image is to be printed or displayed.

Encapsulated PostScript

Encapsulated PostScript (EPS) is a variation of PostScript that describes a single graphic designed to be included in a larger document without modification. EPS files are identical to PostScript files except for some information that describes the size and positioning of the image. As a result, the above discussion about vector and raster representations of your image applies equally to the EPS format. Most word processing and graphics applications support the inclusion of EPS files.

Additional information

• *Controlling the destination and appearance of printed images*

Windows and PostScript image layout

When you print a snapshot of selected viewports directly to a Windows printer or to a PostScript printer or file, the layout of the image is determined by the available page size, the orientation, and the aspect ratio of the viewports:

Available page size

The available page size is calculated from the total page size and the margin information that you supply, as illustrated by the sample dimensions shown in *Figure 1*.

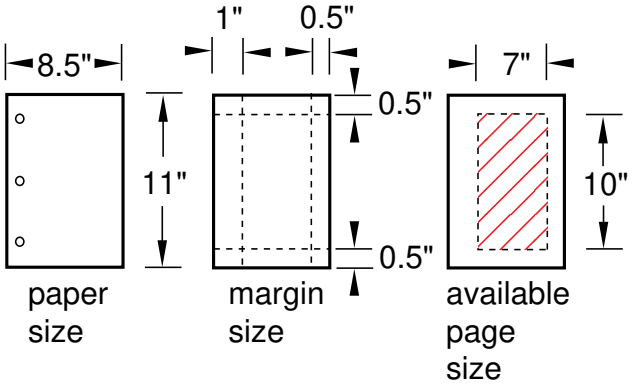


Figure 1: The available page size.

The total page size is determined by the paper size. For a Windows printer, click the **Printer Properties** button in the **Print** dialog box to open the **Document Properties** dialog box and change the paper size. For a PostScript printer or file, click the **Postscript Format Options** button in the **Print** dialog box to change the paper size.

Orientation

The orientation of your page can be either portrait or landscape.

Aspect ratio

The aspect ratio is the ratio between the overall width and the overall height of the viewports that you select for printing. Abaqus/CAE always maintains the aspect ratio of the objects, as shown in *Figure 2*. You can control the aspect ratio by manipulating the viewports on the canvas before printing them.

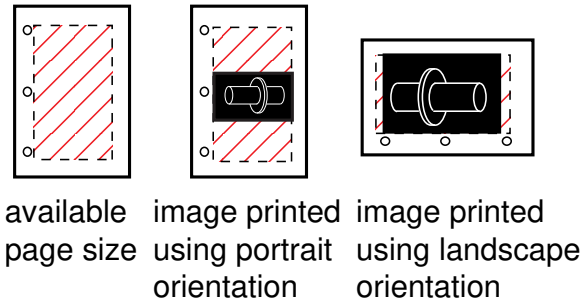


Figure 2: Scaling the objects to maintain the aspect ratio.

The default method for determining the image size on Windows printers—the only method for determining the image size on PostScript printers and files—scales the image to fit the aspect ratio and the available page size.

For detailed instructions, see *Customizing the image sent to a Windows printer*, and *Customizing the image sent to a PostScript printer or file*.

Additional information

- *Controlling the destination and appearance of printed images*
- *EPS, TIFF, PNG, and SVG image size*

Windows printer image size

When you print a snapshot of selected viewports directly to a Windows printer, the image size is determined by one of three methods:

Fit to page

Use this method to scale the printed image to fit within the available page size. This method is used by default.

Use size on screen

Use this method to match the printed image size to the current size on the canvas. If the image size exceeds the available page size, you must resize the image, change the page size or margins, or select a different method to print your image.

Use settings

Use this method to specify the size of the printed image directly. You specify either the width or the height, and Abaqus/CAE adjusts the other dimension to maintain the image aspect ratio. If the image exceeds the available page size, you must resize the image, change the page size or margins, or select a different method to print the image.

Additional information

- *Customizing the image sent to a Windows printer*
- *Controlling the destination and appearance of printed images*
- *Windows and PostScript image layout*
- *EPS, TIFF, PNG, and SVG image size*

EPS, TIFF, PNG, and SVG image size

When you print a snapshot of selected viewports to an Encapsulated PostScript (EPS), TIFF, PNG, or SVG format file, Abaqus/CAE determines the size of the image based on the size you specify and the overall aspect ratio of the viewports. You can control the aspect ratio by manipulating the viewports on the canvas.

In the options dialog box (**EPS Options**, **TIFF Options**, **PNG Options**, or **SVG Options**) you can choose one of the following methods to specify the size of the printed image:

- Use the size of the image on the screen. (Abaqus/CAE indicates the current image size in the options dialog box.) This method is the default.
- Set the width or height. You specify only one dimension; Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports. When you are creating an EPS-format file, you specify the width or height in either inches or millimeters. When you are creating a TIFF, PNG, or SVG format file, you specify the width or height in screen pixels; increasing the number of pixels increases the image size.

Additional information

- *Windows and PostScript image layout*
- *Controlling the destination and appearance of printed images*

Hard-copy image quality

When you print a snapshot of selected viewports directly to a PostScript printer or save it in a PostScript or Encapsulated PostScript (EPS) file, Abaqus/CAE creates either a vector or raster representation of the image (for more information, see *Printed image formats*).

Vector representation images are resolution independent, so their quality depends only on the resolution of your printer.

For PostScript and EPS images, you can use the **Resolution** field in the corresponding options dialog box to specify the resolution of the image you save or print. At higher resolution, raster images appear to be smoother and less jagged. Low-resolution vector images may have holes if scaled to a larger size.

Although a higher resolution image has higher quality, more data are required to define the image; the resulting file can consume a large amount of disk space. A lower resolution image will normally print and display faster. In general, you should select the lowest resolution that still produces an acceptable image. You may want to save a lower resolution image while you produce draft copies of your work and switch to a higher resolution for the finished version.

The resolution of your printer sets an upper limit on the printed image resolution. For example, if you save an image at a resolution of 600 dots per inch (dpi) and print it on a printer that has a resolution of 300 dpi, the printed image will have a resolution of only 300 dpi.

Raster representation image quality may also be affected by changes you make to the image with external software after the image has been created, such as scaling and rotation. Scaling and rotation may distort a raster image. Consequently, before you print a raster representation of your image, you should adjust the viewports on your canvas to match the dimensions and orientation that will appear in the final application. Scaling and rotation do not distort or diminish the quality of vector representation images.

For vector representation PostScript and EPS images, you can use the **Shading Quality** field in the corresponding options dialog box to specify the quality of the lighting on curved surfaces in the image. This option does not affect the image resolution or the file size. A finer shading quality will produce an image closer to the raster representation. A coarser shading quality will normally print and display faster. Similar to the resolution for raster images, you should select the coarsest shading quality that still produces an acceptable image.

Vector PostScript and EPS images do not support translucency; all translucent or transparent objects will appear opaque when printed using vector PostScript or EPS format.

Additional information

- *Understanding printing*
- *Controlling the destination and appearance of printed images*

Importing Abaqus/CAE images into other software products

Many popular software applications, such as word processors, allow you to import files containing graphic images generated by Abaqus/CAE; and most of these applications allow you to preview the imported image. In addition, if you are using a Windows system, you can use **[Ctrl] + C** to copy the image in the current viewport to the system clipboard and **[Ctrl] + V** to paste it into another application. Windows stores the image in the clipboard in a bitmap (`.bmp`) format at the resolution of the screen. Although the image quality will be satisfactory when viewed online, it may be unacceptable when printed. If you expect to print an image, you should save it in PostScript or Encapsulated PostScript format.

Controlling the destination and appearance of printed images

This section describes the options available for controlling the destination and appearance of printed images.

In this section:

- *Printing to a printer or to a file*
- *Selecting which part of the image to print*
- *Choosing the color of your image*
- *Choosing the destination of your image*
- *Customizing the image sent to a Windows printer*
- *Customizing the image sent to a PostScript printer or file*
- *Customizing the image saved in an Encapsulated PostScript file*
- *Customizing the image saved in TIFF, PNG, or SVG files*

Printing to a printer or to a file

Abaqus/CAE allows you to print a snapshot of one or more viewports on the canvas and to send the image either directly to a printer or to a file for later use; for example, to include in a presentation, embed in a printed report, or display in an HTML document. The printed image will reproduce the layering of viewports on the canvas; that is, if one viewport obscures another on the canvas, the obscured portion will not appear in the printed image. You can select the format of the printed image; and additional options allow you to select the appearance of viewports in the resulting image, and the color, resolution, orientation, and size of the image.

To create a printed image, select **File->Print** from the main menu bar. To configure your image, use the **Print** dialog box that appears. For detailed help on the items within the dialog box, request context-sensitive help on the individual items.

When you have finished selecting options, click **OK** in the **Print** dialog box to send the image to the selected destination. Abaqus/CAE closes the **Print** dialog box, sends the image to the selected destination, and saves your print options for the duration of the session.

Additional information

- *Understanding printing*
- *Controlling the destination and appearance of printed images*

Selecting which part of the image to print

When you print an image directly to a printer or to a file, you can use the **Print** dialog box to select which viewports on the canvas to include in the printed image. You can select the following:

All or current viewports

Select **All Viewports** to print all viewports on the canvas. Viewports that are on the canvas but are not visible because they are outside the drawing area will still be printed. If viewports are overlaid, the printed image will reproduce the layering on the canvas; that is, if one viewport obscures another, the obscured portion will not appear in the printed image. By default, Abaqus/CAE prints all the viewports on the canvas.

Select **Current Viewport** to print the most recently used viewport only. See *Selecting viewports*, for more information on selecting viewports.



Note:

Minimized viewports are not printed, regardless of your selections in the **Print** dialog box.

Viewport decorations

Toggle on **Print viewport decorations (if visible)** to select whether your image will include viewport decorations. Decorations are defined as the viewport border and the viewport title.

Viewport backgrounds

Toggle on **Print viewport backgrounds** to control the appearance of a viewport's background in your printed image. When this option is selected and viewport background images or movies are active, Abaqus/CAE includes the viewport background image or the current frame of the movie in the background of the printed image. If **Print viewport backgrounds** is active but viewport background images or movies are inactive, Abaqus/CAE displays the viewport background colors in the background.

The **Print viewport backgrounds** option is available only when you *choose either a greyscale or a color image*; when you choose black and white, Abaqus/CAE always prints a black image on a white background.



Note:

Printing without the viewport background (so that the background appears transparent or white) usually produces the most attractive hard-copy image.

Viewport compass

Toggle on **Print viewport compass (if visible)** to include the 3D compass in your printed image. If the compass is not currently visible in a viewport (see *Customizing the 3D compass*), the compass will not appear in the printed image regardless of the setting for this option.

1. From the main menu bar, select **File->Print**.

    **Tip:** You can also click  in the File toolbar.

   The **Print** dialog box appears.

2. From the **Print** field at the top of the dialog box, select either:

- **All Viewports** to print all viewports, even if they lie outside the drawing area.
- **Current Viewport** to print the viewport that you manipulated most recently.

3. Toggle **Print viewport decorations (if visible)**.

   When **Print viewport decorations (if visible)** is on, all viewport titles and borders that are visible on the canvas will be printed.

   When **Print viewport decorations (if visible)** is off, none of the viewport titles or borders will be printed. To print the current viewport without the border, you must toggle this option off.

4. Toggle **Print viewport backgrounds**.

   When **Print viewport backgrounds** is on, your image will inherit the background color or background images of viewports on your monitor.

   When **Print viewport backgrounds** is off, the appearance of viewport backgrounds depends on the format you choose for your image:

   - When you choose **PS** (PostScript) or **EPS** (Encapsulated PostScript) format or print directly to a Windows printer, viewports in your image will have a white background.
   - When you choose **TIFF**, **PNG**, or **SVG** format, viewports in your image will have a transparent background.

5. Toggle **Print viewport compass (if visible)**.

   When **Print viewport compass (if visible)** is on, any 3D compass that is currently visible in a viewport will appear in your printed image.

   When **Print viewport compass (if visible)** is off, the 3D compass will not appear anywhere in your printed image.

6. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

   Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

• *Understanding printing*
• *Controlling the destination and appearance of printed images*

Choosing the color of your image

When you print an image from the canvas directly to a printer or a file, you can use the **Print** dialog box to select the color of your image. The following color options are available:

Black&White

Use this option to print black images on a white background. This option is useful for printing wireframe and hidden-line images of parts, assemblies, and meshes, including any partitions and datum geometry. You can also print black and white images of undeformed and deformed shape plots. When you choose **Black&White**, Abaqus/CAE always prints a black image on a white background, and the viewport background is printed as either transparent or white. This option should not be used for printing images that depend heavily on color, such as contour plots.

Greyscale

Use this option to print greyscale versions of color images, where each color is approximated by a shade of gray. (Abaqus/CAE converts each color to one of 256 true shades of gray.) This option is useful for printing color images, such as contour plots, to a black and white laser printer. To improve the appearance of images sent to a printer, you may want to print viewports with the background turned off (so that it appears white or transparent).

Color

Use this option to print an approximation of the colors you see. If you try to print a color image to a black and white printer, the printer converts the colors to shades of gray.

By default, Abaqus/CAE uses up to 256 different colors for raster printed images (PNG, TIFF, and raster-format PostScript). However, for *TIFF or PNG* printed images you can choose to allow more colors, which increases the file size but allows the image to more closely match the display. For more information, see *Customizing the image saved in TIFF, PNG, or SVG files*.

1. From the main menu bar, select **File->Print**.

    **Tip:** You can also click  in the File toolbar.

   The **Print** dialog box appears.

2. From the **Rendition** text box in the **Settings** field, select one of the following color options:
   - Select **Black&White** to create a black image on a white background.
   - Select **Greyscale** to print a greyscale approximation of a color image.
   - Select **Color** to print a color approximation of the colors on your screen.

3. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

   Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

- *Understanding printing*
- *Controlling the destination and appearance of printed images*

Choosing the destination of your image

You can choose to send an image directly to a printer, or you can save the image in a file.

If you send the image directly to a printer, you can select the Windows printer that you want to use. If you are on another platform or if there are no printer drivers on your Windows system, Abaqus/CAE selects the PostScript format, and you can specify the print command. Additional options allow you to choose the number of copies, paper size, orientation, margins, image quality, and whether or not to include the date and SIMULIA logo.

If you choose to save the image in a file, you must provide a file name and select one of the following file formats:

PNG

Select **PNG** if you want to incorporate the saved image in a separate document; for example, an HTML file for display on the World Wide Web. Additional options for this format allow you to specify the size of the image. For more information, see *Customizing the image saved in TIFF, PNG, or SVG files*.

SVG

Select **SVG** if you want to incorporate the saved image in a separate document; for example, an HTML file for display on the World Wide Web. Additional options for this format allow you to specify the size of the image. For more information, see *Customizing the image saved in TIFF, PNG, or SVG files*.

TIFF

Select **TIFF** if you want to incorporate the saved image in a separate document; for example, a word processing file. Additional options for this format allow you to specify the size of the image. For more information, see *Customizing the image saved in TIFF, PNG, or SVG files*.

PostScript

Select PostScript (**PS**) if you want the saved image to be identical to the image that Abaqus/CAE would print to a PostScript printer. Additional options for this format allow you to choose the paper size, orientation, margins, and resolution of your image, and whether or not to include the date and SIMULIA logo. For more information, see *Customizing the image sent to a PostScript printer or file*.

Encapsulated PostScript

Select Encapsulated PostScript (**EPS**) if you want to incorporate the saved image in a separate document; for example, a word processing file. Additional options for this format allow you to specify the size and resolution of the image. For more information, see *Customizing the image saved in an Encapsulated PostScript file*.

1. From the main menu bar, select **File->Print**.

   
   Tip: You can also click  in the File toolbar.

   The **Print** dialog box appears.

2. From the **Destination** buttons in the **Settings** field, select one of the following:

   #### Printer

   Choose **Printer** to send your image to a printer. On Windows systems you can select from a list of installed printers, similar to other Windows programs. On other systems, or on Windows systems that do not have a printer driver installed, you can type a print command in the **Print**

command text field. This command should be the same command that you would use at your workstation to print a PostScript file. Do not include a file name in the print command; Abaqus/CAE automatically appends the file name to your command. See your systems administrator for details on the valid commands at your site.

Whether you select a printer or enter a print command, click the arrows in the **Copies** field to set the desired number of copies to print, or type the number of copies you want into the text field. You can print up to 100 copies. If desired, click **Page Setup** and the **Printer Properties** button (for Windows printers) or the **Postscript Format Options** button (for systems using a print command) to specify the page size, printed image quality, and other options.

File

Choose **File** to send your image to a file. There are two ways to supply the file name:

File name

Type the name in the **File name** text field. You can type any characters that are legal Windows or Linux file names; for example, on a Windows system:

- `stressfield.png`
- `..\..\nozzle\presentation\injector_mesh`
- `~\pump\actuator\strainpattern.eps`

If you do not type a file extension, Abaqus/CAE will append an extension (`.png`, `.svg`, `.tif`, `.ps`, or `.eps`) to the file name.

Select

Use the **Select** button to supply a file name using the standard file browser. For more information on file selection, see *Using file selection dialog boxes*.

3. If you selected to print the image to a file, click the arrow next to the **Format** field to select a **PNG**, **SVG**, **TIFF**, PostScript (**PS**), or Encapsulated PostScript (**EPS**) format file. If desired, click the respective options button to specify additional options.
4. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

   Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

- *Printed image formats*
- *Hard-copy image quality*
- *Controlling the destination and appearance of printed images*

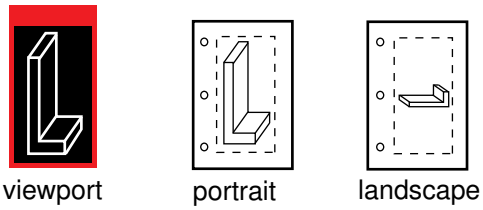
Customizing the image sent to a Windows printer

When you print viewports on the canvas directly to a Windows printer, you can use the **Page Setup** dialog box to customize the resulting printed image.

You can configure the following:

Orientation

You can choose either **Portrait** or **Landscape** orientation. Portrait and landscape orientations are illustrated in the following figure:



Units

You can choose either **Inch** or **Millimeter** units. Your selection determines the units used for the **Image Size** and **Margins** settings that also appear in this dialog box.

Quality

You can select from three levels of quality. Each setting limits the maximum file size that is sent to the printer:

- **Coarse** limits the file to 2 megabytes,
- **Medium** limits the file to 10 megabytes, and
- **Fine** limits the file to 50 megabytes.

The quality level works in conjunction with the image size and the printer settings to determine the resolution of the printed image.



Note:

Click the **Printer Properties** button in the **Print** dialog box to access the printer settings.

Date and logo

By default, Abaqus/CAE includes the date and time and the Abaqus/CAE logo across the top of an image sent directly to a Windows printer. You can choose to remove the date and time or the logo from your output.

Image Size

You can choose one of three methods to determine the size of the printed image:

- **Fit to page** fits the selected image within the current page size and margins.
- **Use size on screen** prints the image as it appears on the screen. Parts of the image that do not fit within the current page size and margins are cut off in the printed output.

- **Use settings below** allows you to enter a width or height; Abaqus/CAE adjusts whichever dimension you do not edit to preserve the current aspect ratio.

Margins

You can provide the **Top**, **Bottom**, **Left**, and **Right** margins. Abaqus/CAE computes the maximum image size as the page size minus the margins. You can specify zero-width margins; however, printers cannot print to the edge of the paper and typically have margins of at least 0.25 inches (6 mm). The page size is set in the printer settings.



Note:

Click the **Printer Properties** button in the **Print** dialog box to access the printer settings.

Abaqus/CAE maintains the margins you specify regardless of the orientation of the paper. For example, assume you chose a **Portrait** image and entered a value for the **Top** margin. If you now choose a **Landscape** image, Abaqus/CAE uses the value you entered for the **Top** margin to compute the **Left** margin. Similarly, the value you entered for the **Right** margin becomes the **Top** margin.

You can also use the **Print** dialog box to set the number of copies to print and to access specific settings for the selected printer. Click the **Printer Properties** button to open the ***printer name*** **Document Properties** dialog box. The available printer settings are determined by the installed printer driver and configuration, not by Abaqus/CAE.

For more information, see *Windows and PostScript image layout*, and *Hard-copy image quality*.

1. From the main menu bar, select **File->Print**.

    **Tip:** You can also click  in the File toolbar.

   The **Print** dialog box appears.
2. From the **Destination** radio buttons, choose **Printer**.
3. From the **Printer** list, select the name of the printer that you want to use.
4. Click the arrows to the right of the **Copies** text field to increase or decrease the number of copies to print or type the number directly in the text field. You can print 1 to 100 copies.
5. From the lower-right corner of the **Print** dialog box, click **Page Setup**.

   The **Page Setup** dialog box appears.
6. From the **Orientation** field, choose the paper orientation.
7. From the **Units** field, choose the units to use for the image size and margins.
8. From the **Quality** field, click the arrow and select **Coarse**, **Medium**, or **Fine** image quality.
9. If desired, toggle off **Print date** to remove the date and time from your output.
10. If desired, toggle off **Print SIMULIA logo** to remove the logo from your output.
11. From the **Image Size** field, choose the size for your printed image.
12. From the **Margins** field, type the **Top**, **Bottom**, **Left**, and **Right** margins (in the units that you selected previously).
13. Click **OK** to save your customization settings and to close the **Page Setup** dialog box.
14. If desired, click the **Printer Properties** button in the **Print** dialog box to open the **Document Properties** dialog box to access options specific to your printer.



Note:

The **Document Properties** dialog box is a Windows dialog box, not part of Abaqus/CAE. If you have questions regarding the information in this dialog box, you should address them to your system administrator or consult the documentation for your printer or printer driver.

15. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

    Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

- *Windows and PostScript image layout*
- *Controlling the destination and appearance of printed images*

Customizing the image sent to a PostScript printer or file

When you print viewports on the canvas to a PostScript file or directly to a PostScript printer, you can use the **PostScript Options** dialog box to customize the resulting printed image.

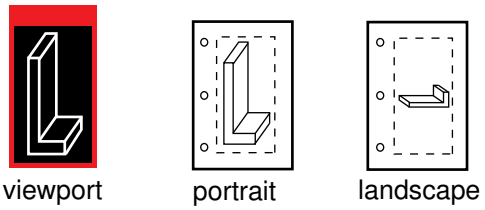
You can configure the following:

Paper Size

You can choose from a list of standard page sizes.

Orientation

You can choose either **Portrait** or **Landscape** orientation. Portrait and landscape orientations are illustrated in the following figure:



Margins

You can provide the **Top**, **Bottom**, **Left**, and **Right** margins. Abaqus/CAE computes the maximum image size as the page size minus the margins. You can specify zero-width margins; however, printers cannot print to the edge of the paper and typically have margins of at least 0.25 inches (6 mm). Abaqus/CAE maintains the margins you specify regardless of the orientation of the paper. For example, assume you chose a **Portrait** image and entered a value for the **Top** margin. If you now choose a **Landscape** image, Abaqus/CAE uses the value you entered for the **Top** margin to compute the **Left** margin. Similarly, the value you entered for the **Right** margin becomes the **Top** margin.

Text Rendering

You can specify how you want text in the viewports to appear in the printed image. You can either use PostScript fonts or request that text characters be output as small bitmaps.

Resolution

You can select from a list of standard resolutions. (For more information, see *Printed image formats*.) The maximum effective resolution of a raster PostScript image is limited to the resolution of the device on which the image will be displayed. By default, Abaqus/CAE sets the resolution of a PostScript image to 150 dpi. To save disk space, you should select the minimum acceptable resolution when generating raster PostScript images. For more information, see *Hard-copy image quality*.

Image Format

You can choose either **Vector** (default) or **Raster** format. Vector images are scalable and resolution independent. Raster (or bitmap) images are pixelated and resolution dependent and tend to decrease in quality when they are scaled.

Shading Quality

For vector images you can choose how fine curved surfaces will be shaded.

Date and logo

By default Abaqus/CAE includes the date and time and an Abaqus/CAE logo across the top of a PostScript image. You can choose to remove the date and time or the logo from your output.

If you are printing to a PostScript printer, the **Print** dialog box also allows you to type a printer command and set the number of copies to print.

For more information, see *Windows and PostScript image layout* and *Hard-copy image quality*.

1. From the main menu bar, select **File->Print**.

   Tip: You can also click in the File toolbar.

   The **Print** dialog box appears.
2. From the **Destination** radio buttons, choose **Printer** to send your image to a PostScript printer or **File** to send your image to a PostScript file.
3. If you are sending your image to a PostScript printer:
   a. In the **Print command** text field, type the print command.
   b. Click the arrows to the right of the **Copies** text field to increase or decrease the number of copies to print or type the number directly in the text field. You can print 1 to 100 copies.
4. If you are sending your image to a PostScript file:
   a. In the **File name** text field, type the file name or click to select the file name from the standard file browser.
   b. From the **Format** field, select **PS**.
5. From the bottom of the **Print** dialog box, click the **Postscript Format Options** button.

   The **PostScript Options** dialog box appears.
6. From the **Paper Size** field, select a standard page size.
7. From the **Orientation** field, choose the paper orientation.
8. From the **Margins** field, type the **Top**, **Bottom**, **Left**, and **Right** margins in inches.
9. From the **Text Rendering** field, choose one of the following:
   - Choose **Always use PostScript printer fonts** to print only font families that are commonly available on a PostScript printer (Courier, Helvetica, Times, and Symbol). Any other font is replaced by Courier, the default font.
   - Choose **Use PostScript printer fonts when available** to print any viewport text that appears in Courier, Helvetica, Times, or Symbol font. Text in any other font is output as small bitmaps for each character. This option requires more processing and results in a larger PostScript file. No fonts are replaced by the default font.
   - Choose **Always use displayed fonts (WYSIWYG)** to output all characters as small bitmaps.
10. From the **Resolution** field, click the arrow and select from the list of resolutions.
11. From the **Image Format** field, select **Vector** or **Raster**.
12. For vector images, select the **Shading Quality**.

13. If desired, toggle off **Print date** to remove the date and time from your output.
14. If desired, toggle off **Print SIMULIA logo** to remove the logo from your output.
15. Click **OK** to save your PostScript customization settings and to close the **PostScript Options** dialog box.
16. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

    Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

- *Windows and PostScript image layout*
- *Controlling the destination and appearance of printed images*

Customizing the image saved in an Encapsulated PostScript file

When you print viewports to an **EPS** (Encapsulated PostScript) file, you can customize the resulting image.

The **Encapsulated PostScript Options** dialog box allows you to configure the following:

Image Size

You can save an image that is the same size as the image on the screen, or you can specify the size of the image in inches or millimeters. You specify either the width or the height; Abaqus/CAE calculates the other dimension to maintain the aspect ratio of the viewports.

Text Rendering

You can specify how you want text in the viewports to appear in the printed image. You can use either PostScript fonts or request that text characters be output as small bitmaps.

Resolution

You can select from a list of standard resolutions. (For more information, see *Printed image formats*.) The maximum effective resolution of a raster representation EPS image is limited to the resolution of the device on which the image will be displayed. By default, Abaqus/CAE sets the resolution of an EPS image to 150 dpi. To save disk space, you should select the minimum acceptable resolution when generating raster PostScript images. For more information, see *Hard-copy image quality*.

Image Format

You can choose either **Vector** (default) or **Raster** format. Vector images are scalable and resolution independent. Raster (or bitmap) images are pixelated and resolution dependent and tend to decrease in quality when they are scaled.

Shading Quality

For vector images you can choose how fine curved surfaces will be shaded.

1. From the main menu bar, select **File->Print**.

    **Tip:** You can also click  in the File toolbar.

   The **Print** dialog box appears.
2. From the **Destination** radio buttons, choose **File**.
3. In the **File name** text field, type the file name or click  to select the file name from the standard file browser.
4. From the **Format** field, select **EPS**.
5. From the bottom of the **Print** dialog box, click the **Encapsulated Postscript Format Options** button.

   The **Encapsulated PostScript Options** dialog box appears.
6. From the **Image Size** field, choose one of the following:
   - Choose **Use size on screen** to save an EPS image that is the same size as the overall width and height of the viewports that you select for printing. Abaqus/CAE displays the resulting size to the right of the **Use size on screen** radio button.

- Choose **Use settings below** to specify the width or height of the resulting image in either inches or millimeters.

7. From the **Text Rendering** field, choose one of the following:
   - Choose **Always use PostScript printer fonts** to print only font families that are commonly available on a PostScript printer (Courier, Helvetica, Times, and Symbol). Any other font is replaced by Courier, the default font.
   - Choose **Use PostScript printer fonts when available** to print any viewport text that appears in Courier, Helvetica, Times, or Symbol font. Text in any other font is output as small bitmaps for each character. This option requires more processing and results in a larger PostScript file. No fonts are replaced by the default font.
   - Choose **Always use displayed fonts (WYSIWYG)** to output all characters as small bitmaps.
8. From the **Resolution** field, click the arrow and select from the list of resolutions.
9. From the **Image Format** field, select **Vector** or **Raster**.
10. For vector images, select the **Shading Quality**.
11. Click **OK** to save your customization settings and to close the **Encapsulated PostScript Options** dialog box.
12. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

    Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

• *Printed image formats*
• *Controlling the destination and appearance of printed images*

Customizing the image saved in TIFF, PNG, or SVG files

When you print viewports to a TIFF, PNG, or SVG format file, you can customize the resulting image. You can save an image that is the same size as the image on the screen, or you can specify the size of the image in pixels.

For more information, see *EPS, TIFF, PNG, and SVG image size* and *Printed image formats*.

1. From the main menu bar, select **File->Print**.

   Note:

   You can also click  in the File toolbar.

   The **Print** dialog box appears.

2. In the **File name** text field, type the file name or click  to select the file name from the standard file browser.
3. From the **Format** field, select **TIFF**, **PNG**, or **SVG**.
4. From the bottom of the **Print** dialog box, click the **TIFF Format Options**, **PNG Format Options**, or **SVG Format Options** button.

   The appropriate dialog box appears.

5. From the **Image Size** field, choose one of the following:

   - Choose **Use size on screen** to save an image that is the same size as the overall width and height of the viewports that you select for printing. Abaqus/CAE displays the resulting size to the right of the **Use size on screen** radio button.
   - Choose **Use settings below** to specify the width or height of the resulting image in units of pixels. Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports. The maximum allowed image dimension (width or height) is 4096 pixels.

6. Click **OK** to save your customization settings and to close the dialog box.
7. For color PNG and TIFF images, you can use the default 8-bit color depth (256 colors) or toggle off **Reduce to 256 colors** to use more colors, which increases the file size but allows the image to more closely match the display.

   If you do not use the default setting, the number of colors available for an image depends on the system type and settings. For Windows systems, Abaqus/CAE uses 24-bit color (1.67 million colors). For Linux systems, Abaqus/CAE uses the same color setting as the display.

8. When you have finished with the **Print** dialog box, click **OK** to generate the desired output.

   Abaqus/CAE generates the output and closes the **Print** dialog box. Your settings in the **Print** dialog box are saved for the duration of the session.

Additional information

- *Printed image formats*
- *EPS, TIFF, PNG, and SVG image size*
- *Controlling the destination and appearance of printed images*

Working with Abaqus/CAE Model Databases, Models, and Files

Almost every modeling operation you perform while working in an Abaqus/CAE module contributes to the definition of a model in a model database.
This part describes Abaqus/CAE models and model databases, the files created by the modeling process, and how you work with these models and files.

In this section:

- *Understanding and working with Abaqus/CAE models, model databases, and files*
- *Importing and exporting geometry data and models*

Understanding and working with Abaqus/CAE models, model databases, and files

This chapter discusses models and model databases and describes the various files that Abaqus/CAE generates and reads.

A finished model contains all the data that Abaqus/CAE needs to create and submit the analysis to Abaqus/Standard or Abaqus/Explicit. Models are stored in a model database.

In this section:

- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Accessing an output database on a remote computer*
- *Understanding the files generated by creating and analyzing a model*
- *Abaqus/CAE command files*
- *Using the File menu*
- *Managing model and output databases*
- *Managing models*
- *Managing session objects and session options*
- *Controlling the input file generated by Abaqus/CAE*
- *Managing macros*

What is an Abaqus/CAE model database?

A model database (file extension `.cae`) stores models and analysis jobs. (For more information on analysis jobs, see *Understanding analysis jobs*.) You can have multiple model databases stored on your workstation or network, but Abaqus/CAE can work on only one of them at any time. A model database can contain more than one model; if you plan to work on multiple models simultaneously, they must be stored in one model database. The model database in use is known as the current model database; Abaqus/CAE displays the name of the current model database across the top of the main window, as shown in *Figure 1*.

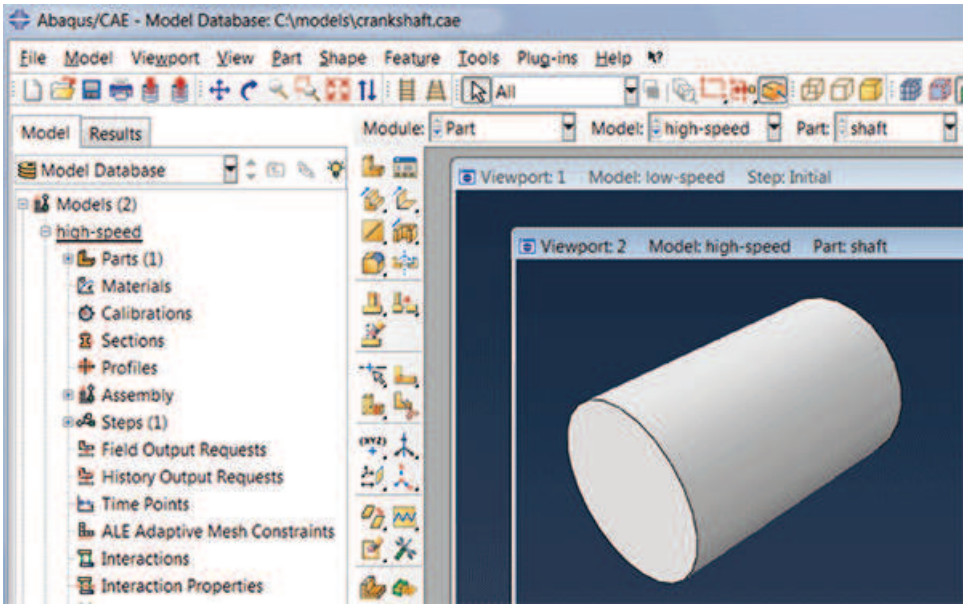


Figure 1: Abaqus/CAE displays the model database name and the model name.

When you first start Abaqus/CAE, the **Start Session** dialog box allows you to either create a new, empty model database or to open an existing model database. Anything you create or define in Abaqus/CAE is stored in this model database. You save the contents by selecting **File->Save** or **File->Save As** from the main menu bar.

Abaqus/CAE never saves the model database unless you perform an explicit save operation; there is no timer-based automatic saving, for example. However, while you work on your model, Abaqus/CAE maintains a record of all the operations that changed the model database. Although you may not have saved the model database, you can always replay the operations that replicate its current state. For more information on recreating the model database, see *Recreating an unsaved model database*. Abaqus/CAE is backward compatible and can open model databases created by previous releases of Abaqus/CAE.

After you begin an Abaqus/CAE session, you can open an existing model database by selecting **File->Open** from the main menu bar, or you can create a new model database by selecting **File->New**. If you open or create another model database after you have made changes to the current one, Abaqus/CAE asks if you want to save the changes before it closes the current model database.

You can open a model database in the Visualization module to probe or query its nodes and elements and to plot contours or symbols for selected attributes. For more information, see *Understanding the role of the Visualization module*.

Additional information

- *Understanding and working with Abaqus/CAE models, model databases, and files*
- *Managing model and output databases*

What is an Abaqus/CAE model?

This section describes an Abaqus/CAE model.

In this section:

- *What does an Abaqus/CAE model contain?*
- *What are the model attributes?*

What does an Abaqus/CAE model contain?

An Abaqus/CAE model contains the following kinds of objects:

- parts
- materials and sections
- assembly
- sets and surfaces
- steps
- loads, boundary conditions, and fields
- interactions and their properties
- meshes

A model database can contain any number of models so that you can keep all models related to a single problem in one database. (For more information, see *What is an Abaqus/CAE model database?*.) You can open multiple models from the model database at the same time, and you can work on different models in different viewports. The viewport title bar (if visible) displays the name of the model associated with the viewport. The model associated with the current viewport (indicated by a red border) is called the current model, and there is only one current model. *Figure 1* shows two viewports displaying two different models (`high-speed` and `low-speed`) in the same model database (`crankshaft.cae`); the current viewport in *Figure 1* is displaying the `high-speed` model.

You use the **Model Manager** or the **Model** menu items from the main menu bar to create and manage your models. You use the **Model** list located in the context bar to switch to a different model in the current model database.

You can create a copy of a model within a model database; in addition, you can copy the following objects between models:

- Sketches
- Parts (part sets are also copied)
- Instances
- Materials
- Sections (including connector sections)
- Profiles
- Amplitudes
- Interaction properties

For detailed instructions, see *Manipulating models within a model database*, and *Copying objects between models*.

You can also import a model from another model database file, which creates a full copy of the model in the current model database. For more information, see *Importing a model from an Abaqus/CAE model database*.

Abaqus/CAE checks that your model is complete when you submit it for analysis. For example, if you request a dynamic analysis, you must specify the density of the materials so that the mass and inertia properties of the model can be calculated. If you did not provide a material density in the Property module, the Job module reports an error; for more information, see *Monitoring the progress of an analysis job*.

In some modules Abaqus/CAE does not support functionality from Abaqus/Standard or Abaqus/Explicit that you may want to include in the analysis. You may be able to add such functionality by using the **Keywords Editor** to edit the Abaqus keywords associated with a model. Select **Model->Edit Keywords->*model name*** from the main menu bar to start the **Keywords Editor**. (You can review the keywords supported by Abaqus/CAE by selecting **Help->Keyword Browser** from the main menu bar.)

You can specify that a model uses information from a previous analysis. When you submit the model for analysis, Abaqus/CAE continues the analysis from a selected step. For more information, see *Configuring restart output requests*, and *Restarting an analysis*.

What are the model attributes?

The model attributes describe characteristics of a model and are stored with a model in the model database.

The following list describes the attributes of an Abaqus/CAE model:

- Description. If you have many similar models in a model database, you can use the description to distinguish between the models. The description that you enter is stored with the model attributes; the description is written above the header of the input file but is not written to the output database. For more information, see *Adding descriptions to your Abaqus/CAE model*.
- Type. You can choose between a **Standard & Explicit** model (the default) or an **Electromagnetic** model. Once you select a model type, Abaqus/CAE filters the set of options available in the main menu bar, toolboxes, and Model Tree so that they are appropriate to your model type selection.
- Physical constants for the model. You can enter values for the **Absolute zero temperature** and the **Stefan-Boltzmann constant**. These values are needed to specify surface emissivity and radiation conditions in heat transfer analyses.

  You can also enter a value for the **Universal gas constant**, and you can choose an option from the **Specify acoustic wave formulation** list.
- Restart information that will start the analysis using data from a previous analysis. You can specify the following:
  - The name of the job from which Abaqus/CAE will read the restart information.
  - The name of the step from which Abaqus/CAE will restart the analysis.
  - The increment or the interval of the step from which Abaqus/CAE will restart the analysis.

  For more information, see *Restarting an analysis* and *Restarting an Analysis*.
- Submodel information that will be used to drive submodel boundary conditions or loads in the model. You can specify the following:
  - The job from which the global solution will be used to drive the submodel boundary conditions or loads.
  - Whether a shell global model will be used to drive a solid submodel.

  For more information, see *Submodeling*.
- Model instance information. You can control whether constraints, connector section assignments, and surface-to-surface contact and self-contact interactions defined in the initial step will be copied to the current working model when you create model instances from this model. For more information, see *Working with model instances*.

Select **Model->Edit Attributes->*model name*** from the main menu bar to edit the attributes of the selected model.

Additional information

- *Specifying model attributes*

Accessing an output database on a remote computer

This section describes how you can create and start a network connector.
You can use the network connector to navigate the directory structure on a remote host and to access a remote output database.

In this section:

- *What is a network ODB connector?*
- *How secure is the access to a network ODB connector?*
- *Tuning the cache size to increase the performance of a network ODB*

What is a network ODB connector?

A network ODB connector creates a connection to a remote machine and allows you to access a remote output database. For example, you can submit an analysis to a high-performance Linux system and view the results on a local Windows workstation while the analysis is still running.

You can create a network ODB connector from any platform—Windows or Linux. However, the network ODB server must reside on a Linux platform; you cannot access an output database that resides on a remote Windows system. You can access only a remote output database; you cannot access a remote model database.

Select **File->Network ODB Connector->Create** from the main menu bar to create a connection with a directory on a remote host. When you are creating a network ODB connector, you can use Abaqus/CAE to automatically start the network ODB server and to establish the communication port numbers on the host and remote systems. Alternatively, you can start the network ODB server manually from the command line using the **abaqus networkDBConnector** execution procedure. If you start the server from the command line, you enter the communication port number returned by the execution procedure when you subsequently create the network ODB connector. For more information, see *Network Output Database File Connector*.

After you create a network ODB connector, you must start it by selecting **File->Network ODB Connector->Start->*Connector name*** from the main menu bar. The remote system must have Abaqus installed for Abaqus/CAE to establish the network connection. For more information, see *Creating a network ODB connector*, and *Managing network ODB connectors*.

After you create and start a network connector, you can use it to navigate the directory structure on a remote host. When you select **File->Open** from the main menu bar to open a database from Abaqus/CAE, a **Network connectors** entry appears under **Directory** in the file selection dialog box. The entry appears regardless of whether you are trying to open an output database or a model database; however, you cannot use a network connector when opening a model database. For more information, see *Using file selection dialog boxes*.

The network connector allows you to do the following:

- Open a remote output database in read-only mode, and view the contents of the output database using the Visualization module. For more information, see *Opening a model database or an output database*.

  The behavior of the Visualization module does not change when the output database is remote; for example, you can view the output database while the analysis is running on a remote machine, and more than one user can view the output database. However, you cannot click **Results** in the **Job Manager** to open the remote output database associated with a remote analysis.
- Import a part from a remote output database. For more information, see *Importing parts*.
- Import a model from a remote output database. For more information, see *Importing a model from an output database*.
- Upgrade a remote output database.

After most Visualization module operations and during animations, Abaqus/CAE monitors the output database for updated results and updates the current viewport accordingly. If you are displaying data from a remote output database, the performance of Abaqus/CAE may be degraded if the time taken to monitor the database over the network is significant. To increase the performance, you can reduce the frequency with which Abaqus/CAE monitors the output database for updates or you can disable the monitoring. For more information, see *Controlling results caching*.

How secure is the access to a network ODB connector?

Abaqus/CAE maintains a secure connection to the network ODB connector by generating a key that is passed back and forth between the server and the client. If a file called `.abaqus_net_passwd` is present in your home directory on the remote server, Abaqus/CAE uses the password in the file for authentication instead of the key generated by Abaqus/CAE. Abaqus/CAE checks that you are the only user with permission to read and write to the password file. In addition, you must update the file after 30 days, and the password must be at least eight characters long. Abaqus uses password files to authenticate the connection between the client and the server if you start the network ODB server manually. These files are described in *Network Output Database File Connector*.

Tuning the cache size to increase the performance of a network ODB

When you start an Abaqus/CAE session, a cache is created in the scratch file directory. Abaqus/CAE uses this cache for local data storage when you use a network ODB connector to read from a remote output database. The cache greatly increases the performance of the Visualization module in Abaqus/CAE when accessing data from the remote output database.

Abaqus/CAE allows the cache to grow to a size that is sufficient to contain all of the data in all of the open remote output databases. However, Abaqus/CAE limits the cache size to 80% of the total free space in the directory. For example, if the scratch directory has 35 gigabytes of unused space, Abaqus/CAE will allow the cache to grow to 28 gigabytes. Alternatively, you can limit the size of the cache using the **nodb_cache_limit** parameter in the Abaqus environment file, `abaqus_v6.env`. You must set the **nodb_cache_limit** parameter to the number of megabytes to which the cache size will be limited. For example,

```
nodb_cache_limit=20000
```

will set the maximum cache size to 20 gigabytes. Abaqus/CAE uses this cache space only as needed during a session, and the actual cache size may be significantly less than the limit you specified. The minimum value of **nodb_cache_limit** is 500, indicating that the cache size is limited to 500 megabytes. If you set the maximum cache size to be greater than the available free space, Abaqus/CAE reduces it to a value that is equal to the available free space.

Abaqus/CAE uses the cache to increase its performance when reading data from a remote output database. The speed at which data can be accessed over a network is significantly lower than the speed at which data can be accessed from a local disk drive. As a result, the performance of remote output databases will be significantly slower than the performance of a local output database. The cache reduces this performance difference by retaining data that have been transferred over the network, thereby reducing the need for data transfer over the network. However, if the cache is not large enough, Abaqus/CAE will have to transfer more data over the network and performance will suffer.

In most cases you will not have to tune the size of the cache using the **nodb_cache_limit** parameter. However, you may have to reduce the size of the cache if it is consuming too much disk space and reducing the speed of other applications on your system. Similarly, you may have to increase the size of the cache if it is too small to support all of your remote output databases and the performance of Abaqus/CAE is degraded. If you cannot increase the size of the cache, you should close some of your remote output databases.

If the desired cache size is larger than the space available in the scratch file directory, you can move the scratch file directory to a larger disk drive using the Abaqus**scratch** environment file parameter. For more information, see *Environment File Settings*, and *Managing Memory and Disk Resources*.

Understanding the files generated by creating and analyzing a model

When you start a session and begin defining your model, Abaqus/CAE generates the following file:

The replay file (*abaqus* . `rpy`)

The replay file contains Abaqus/CAE commands that record almost every modeling operation you perform during a session. For more information, see *Replaying an Abaqus/CAE session*.

When you select **File->Save** from the main menu bar and save the model database, Abaqus/CAE saves the following files:

The model database file (*model_database_ name* . `cae`)

The model database file contains models and analysis jobs. For more information, see *What is an Abaqus/CAE model database?*.

The journal file (*model_database_ name* . `jnl`)

The journal file contains the Abaqus/CAE commands that will replicate the model database that was saved to disk. For more information, see *Recreating a saved model database*.

When you continue to work on your model, Abaqus/CAE continues to record your actions in the replay file. In addition, Abaqus/CAE saves the following file:

The recover file (*model_database_ name* . `rec`)

The recover file contains the Abaqus/CAE commands that will replicate the version of the model database in memory. The model database recovery file contains only the commands that changed the model database since you last saved it. For more information, see *Recreating an unsaved model database*.

When you submit a job for analysis, Abaqus/Standard and Abaqus/Explicit create a set of files; for a complete list of these files, see *File Extension Definitions*. The following list describes some of the files that Abaqus/Standard and Abaqus/Explicit create and their relationship to Abaqus/CAE:

Input files (*job_name* . `inp`)

Abaqus/CAE generates an input file that is read by Abaqus/Standard or Abaqus/Explicit when you submit a job for analysis. For more information, see *Basic steps for analyzing a model*.

Output database files (*job_name* . `odb`)

Output database files contain the results from your analysis. You use the Step module's *output request managers* to choose which variables are written to the output database during the analysis and at what rate. An output database is associated with the job you submit from the Job module; for example, if you named your job `FrictionLoad`, the analysis creates an output database called `FrictionLoad.odb`.

When you open an output database, Abaqus/CAE loads the Visualization module and allows you to view a graphical representation of the contents. You can also import a part from an output database as a mesh. You can save *X–Y* data objects to an output database file if you open the file with write permission; otherwise, you cannot modify the contents of the output database once it has been created.

The output database lock file (*job_name*`.lck`)

The lock file (*job_name*`.lck`) is written whenever an output database file is opened with write access, including when an analysis is running and writing output to an output database file. The lock file prevents you from having simultaneous write permission to the output database from multiple sources. It is deleted automatically when the output database file is closed or when the analysis that creates it ends.

The restart file (*job_name*`.res`)

The restart file is used to continue an analysis that stopped before it was complete. You use the Step module to specify which analysis steps should write restart information and how often. If you are using Abaqus/Explicit, the restart information you supply in the Step module controls the data written to the state file (*job_name*`.abq`). For more information, see *Configuring restart output requests*.

The data file (*job_name*`.dat`)

The data file contains printed output from the analysis input file processor, as well as printed output of selected results written during the analysis. Abaqus/CAE automatically requests that the default printed output for the current analysis procedure be generated at the end of each step; you cannot use Abaqus/CAE to exert any additional control over the contents of the data file.

The message file (*job_name*`.msg`)

The message file contains diagnostic or informative messages about the progress of the solution. You can control the diagnostic information that is output to the message file using the Step module. For more information, see *Diagnostic printing*.

The status file (*job_name*`.sta`)

The status file (*job_name*`.sta`) contains information about the progress of the analysis. In addition, you use the Step module to request that the value of a single degree of freedom at a single node be output to the status file. For more information, see *Degree of freedom monitor requests*.

The results file (*job_name*`.fil`)

The results file contains selected results from the analysis in a format that can be read by other applications, such as postprocessing programs. A submodel analysis can read the global model results from either an output database or a results file. By default, an analysis from Abaqus/CAE does not create a results file. For more information, see *Submodeling*, and *Submodeling*.



Note:

The errors and warnings that Abaqus/Standard and Abaqus/Explicit write to the data, message, and status files while analyzing a job can be monitored by the Job module; for more information, see *Monitoring the progress of an analysis job*.

When you open an output database file in the Visualization module and create new field output variables (see *Creating and saving new field output*, for more information), Abaqus/CAE generates the following file:

The scratch output database file (*job_name*`.ods`)

The scratch output database file (*job_name*`.ods`) contains a "session step" in which field output variables that you create (by operating on either fields or frames) are saved. This file is deleted automatically when the original output database file (from which the field output originates) is closed or when the Abaqus/CAE session ends.

In most cases the files generated by Abaqus/CAE are written to the work directory. The work directory is the directory from which you started the Abaqus/CAE session unless you changed the directory by selecting **File->Set Work Directory** from the main menu bar. For more information, see *Setting the work directory*.

Abaqus/CAE command files

This section describes the command files that you can use to reproduce your work and to customize Abaqus/CAE.

In this section:

- *Replaying an Abaqus/CAE session*
- *Recreating a saved model database*
- *Recreating an unsaved model database*
- *Creating and running your own scripts*
- *Creating and running a macro*
- *Customizing your Abaqus/CAE environment*

Replaying an Abaqus/CAE session

Almost every operation that you perform in Abaqus/CAE is recorded automatically in the replay file (`abaqus.rpy`) in the form of Abaqus Scripting Interface commands. Executing the replay file is equivalent to replaying the original sequence of operations including any redundant procedures and any mistakes and subsequent corrections that you made. The replay file also includes canvas operations, such as creating a new viewport.

Abaqus/CAE retains the five most recent versions of the replay file. The most recent version of the replay file is called `abaqus.rpy`; it is created when you start a session. The four older versions have a number appended to the end of the file name; the file name with the lowest number indicates the oldest replay file, and the file name with the highest number indicates the second most recent replay file.

You can execute the commands in a replay file when you start Abaqus/CAE or during a session; however, the result may be different if the replay file generates an error.

From the Abaqus execution procedure

To run a replay file from the Abaqus execution procedure, type `abaqus cae` (or `abaqus viewer`) `replay=replay_file_name.rpy`. If executing the replay file generates an error, Abaqus/CAE ignores the error and continues to the next command in the replay file. As a result, Abaqus/CAE always attempts to execute every command in the replay file. You cannot use the `replay` option to execute a script with control flow statements. For more information, see *Abaqus/CAE Execution*.

During an Abaqus/CAE session

To run a replay file during a session, select **File->Run Script** from the main menu bar. If the replay file generates an error, Abaqus/CAE stops executing the replay file and displays an error message in the command area. It is recommended that you run a replay file from the Abaqus execution procedure.

You can also execute a replay file using the Abaqus Python development environment (AbaqusPDE). The Abaqus Scripting Interface commands in the replay file must be run in the kernel workspace in the AbaqusPDE. For more information on the AbaqusPDE, see *The Abaqus Python Development Environment*.

Recreating a saved model database

When you save a model database (by selecting **File**->**Save** or **File**->**Save As** from the main menu bar), Abaqus/CAE also saves a model database journal file (*model_database_name*`.jnl`) containing the Abaqus Scripting Interface commands that will recreate the model database. Should the saved model database become corrupted, you can recreate it by starting Abaqus/CAE with the recover option. (Type `abaqus cae recover=model_database_name.jnl`.) The recover option executes the commands in the specified model database journal file.

The model database journal file differs from the replay file in that it does not contain every operation performed during a session. The model database journal file contains only the commands that change the saved model database; for example, commands that create or edit a part, change the time incrementation of an analysis step, or modify the mesh. Operations that do not change the model database are not saved in the journal file; for example, sending an image to a printer, creating a viewport, rotating the model, or viewing results in the Visualization module.

As you continue to work on your model, the model database in memory will differ from the most recently saved model database. The model database journal file is updated only when you perform an explicit save of the model database using **File**->**Save** or **File**->**Save As**. If you copy the model database to a different location, you should also copy the associated model database journal file. Otherwise, you will not be able to recreate the model database.

Recreating an unsaved model database

When you start a new session and make changes to your model, Abaqus/CAE records those changes to a model database recovery file (`abaqus`*n*`.rec`). If you subsequently save the model database, Abaqus/CAE appends the commands in the recovery file to the journal file for that model database (*model_database_name*`.jnl`) and deletes the recovery file. If you make further changes to your model, Abaqus/CAE creates a new recovery file (*model_database_name*`.rec`) to record the changes since your last save. Upon your next save, the commands in the recovery file are appended to the journal file and the recovery file is deleted. The journal file contains all the commands necessary to rebuild the entire model database. For example, *Table 1* shows the changes that Abaqus/CAE makes to the model database, recovery, and journal files for a model named `engine`.

Table 1: Modeling changes and their effect on the model database, recovery, and journal files.

User action	Abaqus/CAE action	Files
Start Abaqus/CAE session	None	None
Make model changes	Record commands in recover file	`abaqus1.rec`
Save the model database	Create model database file Copy recover commands to journal file Delete recover file	`engine.cae` `engine.jnl`
Make more changes to the model	Record commands in recover file	`engine.rec` `engine.cae` (out of date) `engine.jnl` (out of date)
Save model database	Update model database file Append recover commands to journal file Delete recover file	`engine.cae` (updated) `engine.jnl` (updated)

If your Abaqus/CAE session exits unexpectedly—for example, because of a power loss to your computer—the recovery file will still be available to Abaqus/CAE for your next session. Abaqus/CAE first checks for the presence of a recovery file of the form `abaqus`*n*`.rec`; if such a file exists, it might be from a previous session that stopped unexpectedly, or it might be from another Abaqus/CAE session that you started in the same directory. Because Abaqus/CAE cannot tell the difference between these two cases and cannot determine automatically whether you want to implement the changes, Abaqus/CAE prompts you with three options: recover the changes and delete the recovery file, do not recover changes and delete the recovery file, or disregard the recovery file because its changes belong to another Abaqus/CAE session. When you recover changes, you can skip the last command in the recovery file if you think the last command you issued caused the termination of the session.

If a recovery file belongs to a model database (*model_database_name*`.rec`), Abaqus/CAE will not detect the recovery file until you attempt to open that model database. Upon your attempt to open the model database, Abaqus/CAE prompts you to recover or disregard the changes. If you recover the changes, Abaqus/CAE appends the changes in the database recovery file to the journal file and deletes the database recovery file; if you choose to disregard the changes, Abaqus/CAE deletes the recovery file and does not implement any of the model changes described in the file.

Creating and running your own scripts

Almost every operation that you perform during an Abaqus/CAE session can be duplicated by a script (*script_name*`.py`) containing a set of Abaqus Scripting Interface commands. Conversely, running a script from within Abaqus/CAE is equivalent to performing the corresponding operations using the menus, toolboxes, and dialog boxes that Abaqus/CAE provides.

You can create scripts that duplicate operations you perform routinely during a session; for example, you might write a script that defines the material properties of a commonly used material or one that produces a contour plot of a particular variable shown in a particular view orientation.

Abaqus/CAE commands are written in the Python scripting language, and you can use Python to enhance the scripts generated by Abaqus/CAE. Commands are stored as ASCII text in the replay, journal, and recovery files and in Abaqus/CAE scripts that you create. As a result, you can use a standard text editor to edit the contents of the files. For more information on commands, see the *Abaqus Scripting User's Guide*.

To run a script, select **File->Run Script** from the main menu bar, and select the script to run from the **Run Script** dialog box.



Note:

You should use the recover option from the Abaqus/CAE execution procedure to run a journal file and recreate a saved model database. (Type `abaqus cae recover=model_database_name.jnl`.) Selecting **File->Run Script** to run a journal file may result in an incomplete model database.

You can also create and run scripts using the Abaqus Python development environment (AbaqusPDE). The Abaqus Scripting Interface commands in the scripts must be run in the kernel workspace in the AbaqusPDE. For more information on the AbaqusPDE, see *The Abaqus Python Development Environment*.

Creating and running a macro

The **Macro Manager** allows you to record a sequence of Abaqus Scripting Interface commands in a macro file while you interact with Abaqus/CAE. Each command corresponds to an interaction with Abaqus/CAE, and replaying the macro reproduces the sequence of interactions. You can use a macro to automate tasks that you find yourself performing repeatedly, such as printing the current viewport or applying a predefined view. For more information on Abaqus Scripting Interface commands, see the *Abaqus Scripting User's Guide*.

Macros are stored in a file called `abaqusMacros.py`. Abaqus/CAE searches three directories for `abaqusMacros.py`, in the following order:

- The site directory of the Abaqus installation.
- Your home directory.
- The current working directory.

The `abaqusMacros.py` file can exist in more than one of these directories. The **Macro Manager** contains a list of the existing macros that Abaqus/CAE detected in all of the `abaqusMacros.py` files. If a macro uses the same name in more than one `abaqusMacros.py` file, Abaqus/CAE uses the last macro encountered.

To create, delete, or run a macro, select **File->Macro Manager** from the main menu bar. For more information, see *Managing macros*.

Customizing your Abaqus/CAE environment

You use the Abaqus environment file (`abaqus_v6.env`) to specify parameters that control Abaqus/Standard and Abaqus/Explicit. In addition, you can use the environment file to specify a set of commands that are executed when you start an Abaqus/CAE session. Examples of commands that configure how you want a job to run on a remote host computer are given in *Submitting a job remotely*.

Using the File menu

A variety of operations are available from the File menu.

Use the items under **File** on the main menu bar to do the following:

- Select **File->New Model Database->With Standard/Explicit Model** to create a new model database for an Abaqus/Standard or an Abaqus/Explicit analysis. You can also click  in the File toolbar. For more information, see *Creating a new model database*.
- Select **File->New Model Database->With Electromagnetic Model** to create a new model database for an electromagnetic analysis. For more information, see *Creating a new model database*.
- Select **File->Open** to open an existing model database or output database. You can also click  in the File toolbar. For more information, see *Opening a model database or an output database*.
- Select **File->Network ODB Connector** to create a connection with a remote host that you can use to read a remote output database. For more information, see *Creating a network ODB connector*.
- Select **File->Close ODB** to close an output database. For more information, see *Closing the current output database*.
- Select **File->Set Work Directory** to change the work directory. For more information, see *Setting the work directory*.
- Select **File->Save** to save the current model database. You can also click  in the File toolbar. For more information, see *Saving the current model database*.
- Select **File->Save As** to save the current model database to a new file with a different name. For more information, see *Saving the current model database to a new file with a different name*.
- Select **File->Compress MDB** to compress the current model database. For more information, see *Compressing the file size of the current model database*.
- Select **File->Save Display Options** to save your customized part, assembly, and Visualization module display settings. For more information, see *Understanding Abaqus/CAE GUI settings*, and *Saving your display options settings*.
- Select **File->Save Session Objects** to save session-specific object definitions such as view cuts, display groups, or paths to a file, model database, or output database. For more information, see *Managing session objects and session options*.
- Select **File->Load Session Objects** to load previously saved session-specific object definitions into the current session. For more information, see *Managing session objects and session options*.
- Select **File->Import->Sketch** to import a planar sketch. For more information, see *Importing sketches*.
- Select **File->Import->Part** to import a part. For more information, see *Importing parts*.
- Select **File->Import->Model** to import a model. For more information, see *Importing a model*.
- Select **File->Export->Sketch** to export the current sketch. For more information, see *Exporting a sketch to an ACIS-, IGES-, or STEP-format file*.
- Select **File->Export->Part** to export the current part. For more information, see *Exporting a part to an ACIS-, IGES-, STEP-, or VDA-format file*.
- Select **File->Export->Assembly** to export the part instances in the assembly. For more information, see *Exporting the assembly to an ACIS-format file*.
- Select **File->Export->VRML** to export the current viewport to a VRML-format file. For more information, see *Exporting viewport data to a VRML-format file*.
- Select **File->Export->3DXML** to export the current viewport to a 3D XML-format file. For more information, see *Exporting viewport data to a 3D XML-format file*.

- Select **File->Export->OBJ** to export the current viewport to an OBJ-format file. For more information, see *Exporting viewport data to an OBJ-format file*.
- Select **File->Run Script** to execute a file containing Abaqus Scripting Interface commands. For more information, see *Replaying an Abaqus/CAE session*, and *Creating and running your own scripts*.
- Select **File->Macro Manager** to store your actions in a macro file as a sequence of Abaqus Scripting Interface commands. You can also run a macro and rename an existing macro. For more information, see *Creating and running a macro*.
- Select **File->Print** to print all or selected viewports and annotations. You can also click  in the File toolbar. For more information, see *Printing viewports*.
- Select **File->Abaqus PDE** to open the Abaqus Python development environment. The AbaqusPDE is a separate application used to create, edit, test, and debug scripts. For more information, see *About the Abaqus Python development environment*.
- Select **File->Exit** to exit the Abaqus/CAE session. For more information, see *Exiting an Abaqus/CAE session*.

Additional information

- *Understanding the files generated by creating and analyzing a model*
- *Abaqus/CAE command files*

Managing model and output databases

This section describes how you use the main menu bar's **File** menu to manage model and output databases.

In this section:

- *Creating a new model database*
- *Opening a model database or an output database*
- *Upgrading a model database or an output database*
- *Creating a network ODB connector*
- *Customizing a network ODB connector*
- *Managing network ODB connectors*
- *Closing the current output database*
- *Setting the work directory*
- *Saving the current model database*
- *Saving the current model database without a license*
- *Saving the current model database to a new file with a different name*
- *Compressing the file size of the current model database*

Creating a new model database

You can create and store multiple model databases on your computer, but you can have only one model database open at any time.
Choose one of the following options to create a new model database:

- Click  in the File toolbar or select **File->New Model Database->With Standard/Explicit Model** to create a new model database for an Abaqus/Standard or an Abaqus/Explicit analysis.
- Select **File->New Model Database->With Electromagnetic Model** to create a new model database for an electromagnetic analysis.

If you have made any changes to the current model database, Abaqus/CAE asks if you want to save your changes before it closes the current model database and creates the new one. The new database then becomes the current database. To save the new model database, select **File->Save** from the main menu bar and enter the name of the database. After you save the model database, Abaqus/CAE displays its name in the title bar of the main window.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model database?*
- *Understanding the files generated by creating and analyzing a model*
- *Using the File menu*

Opening a model database or an output database

Select **File->Open** from the main menu bar to open either:

- A model database (file extension `.cae`)
- An output database (file extension `.odb`)

From the **Open Database** dialog box that appears, select the **File Filter** and the file to open and click **OK**.

You can open multiple output databases and display the combined contents of the output databases in an overlay plot in a single viewport by using the **Append to layers** option. For details about working with overlay plots, see *Overlaying multiple plots*.

By default, output database files are opened as read-only. You can choose to open an output database file with write privileges; you must do so if you want to copy any *X–Y* data objects to the output database (see *Copying a session X–Y data object to an output database file*, for more information). Output database files that reside on remote machines can be opened only as read-only; you cannot write to a remote output database file.

Output and model database files from previous releases of Abaqus must be upgraded to the current release when they are opened (for more information, see *Upgrading a model database or an output database*). When you open multiple output database files, all the files must be upgraded already to the current release; otherwise, Abaqus/CAE will print a warning in the message area, and the files requiring upgrade will not be opened.

If you are using an earlier release of Abaqus/CAE, you cannot open a model or output database file created from a later release.

1. From the main menu bar, select **File->Open**.

    **Tip:** You can also click  in the File toolbar to open a model database or an output database.

   Abaqus/CAE displays the **Open Database** dialog box.

2. From the **File Filter** menu at the bottom of the **Open Database** dialog box, select one of the following:

Model Database (*.cae)

Abaqus/CAE lists all the files in the selected directory with the file extension `.cae`.

Output Database (*.odb*)

Abaqus/CAE lists all the files in the selected directory with the file extension `.odb`.

Model & Output Databases (*.cae, *.odb*)

Abaqus/CAE lists all the files in the selected directory with file extension `.cae` or `.odb`.

3. If you selected **Output Database (*.odb*)** in Step 2, use the following options to filter the list of files further or to change the file opening behavior:

Network connectors

If you previously created and started a network ODB connector, the **Directory** field includes a **Network connectors** item that allows you to access a remote directory and open a remote output database. For more information, see *Creating a network ODB connector*.

Read-only

By default, output database files are opened as read-only. To open an output database file with write privileges, toggle off **Read-only** near the bottom of the **Open Database** dialog box before clicking **OK**. You must open an output database file with write privileges if you want to copy *X–Y* data objects from your session to the file or permanently upgrade an old file to the current release of Abaqus/CAE. You can open a remote output database file only as read-only.

Append to layers

Toggle on **Append to layers** and select multiple files if you want to open more than one output database and display the combined contents in an overlay plot in a single viewport. You can use any of the following methods to select multiple output database files:

- Select files using **[Shift] + Click** or **[Ctrl] + Click**.
- Type a comma-separated list of file names in the **File Name** field, such as

  ```
  lug.odb,hinge.odb
  ```
- Type a list of file names surrounded by double quotes in the **File Name** field, such as

  ```
  "lug.odb" "hinge.odb"
  ```



Note:

If you open an output database file while the analysis that creates it is running but before output results are written, you may have to close the file and reopen it after the results are available.

4. Click **OK** to open the selected file or files.

   Abaqus/CAE saves the selected filter type for use as the default the next time you open a file and closes the **Open Database** dialog box.

   If you opened a model database, Abaqus/CAE displays its name in the title bar of the main window. All operations now refer to the new model database. If you have modified the current model database, Abaqus/CAE asks if you want to save it before opening the selected model database.

   If you opened one or more output databases, Abaqus/CAE starts the Visualization module in the current viewport and displays the model that is last, alphabetically, in the undeformed plot state. Any other selected output databases are opened but not displayed unless you toggled on **Append to layers**, in which case the selected output databases are all plotted in the same viewport.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *Upgrading a model database or an output database*

Upgrading a model database or an output database

Output and model database files from previous releases of Abaqus must be upgraded to the current release when they are opened.

To upgrade an output database permanently, you must either open the file with write permissions and convert it when prompted or use the **abaqus upgrade** utility (see *Output Database Upgrade Utility*). You can use the **abaqus upgrade** utility to upgrade a remote output database only from the system on which the database resides.

When a model database or an output database from a previous release is opened, Abaqus/CAE does one of the following:

- If Abaqus/CAE has permission to write to the original file (i.e., it is an output database file that you have chosen to open with write privileges or a model database file), you are prompted to convert the file to the current release. During the conversion Abaqus/CAE creates a backup of the original model or output database and the journal file associated with the model database; the converted database file and the new journal file are saved in the current directory (the directory from which you opened Abaqus/CAE) with the original file names. If the database was opened from a directory other than the current directory, that directory will still contain the original version of the file with the original file name.

  When the conversion is complete, Abaqus/CAE creates a log file called *file_name*`-upgrade.log` that indicates the result of the conversion. For upgrades of model database files, Abaqus/CAE also displays a dialog box that provides the name of the conversion log file and includes the **View the conversion log file** option. Toggle on this option, and click **OK** to display the conversion log file in an Abaqus/CAE dialog box from which you can browse the log file or search its contents for error messages.
- If you are opening a local output database file as read-only, Abaqus/CAE automatically creates a converted version of the output database that is saved to a temporary location. The converted output database file is saved to the directory defined by the `$TMPDIR` (Linux) or `TEMP` (Windows) environment variable on your system. This temporary version of the output database file is deleted when you exit Abaqus/CAE.
- If you are opening a remote output database file, Abaqus/CAE tries to create a converted version of the output database file that is saved to a temporary location. The converted output database file is saved to the `/tmp` directory on the remote system or to the directory defined by the `$TMPDIR` environment variable on the remote system. This temporary version of the output database file is deleted when you exit Abaqus/CAE.

When you upgrade an older model database, the upgrade process may place keywords that you added manually to a model in the wrong location in the upgraded input file. As a result, you may experience problems when you submit the model for analysis in the Job module. If that is the case, you should open the upgraded model, return to the **Keywords Editor**, and click **Discard All Edits** to delete all of the keywords that you added. You can then recreate the keywords in the correct location in the input file.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *Opening a model database or an output database*

Creating a network ODB connector

You can use a network ODB connector to access an output database on a remote computer. For example, you can submit an analysis to a high-performance Linux compute server and view the results on a local Windows workstation. You can create a network ODB connector from any platform—Windows or Linux. However, the server for the network ODB server must reside on a Linux platform. Abaqus/CAE maintains a secure connection to the network ODB connector by generating a key that is passed back and forth between the server and the client. For more information, see *How secure is the access to a network ODB connector?*.

Select **File->Network ODB Connector->Create** from the main menu bar to create a connector. After you create a network ODB connector, you must start it by selecting **File->Network ODB connector->Start->*Connector name*** from the main menu bar. The remote system must have Abaqus installed for Abaqus/CAE to establish the network connection. For more information, see *Managing network ODB connectors*.

In most cases you will use Abaqus/CAE to start the network ODB server on the remote system and to assign port numbers. Abaqus/CAE can start the server only if the user name on the remote host is the same as the user name on the local system. If you experience problems establishing communication or if the user names are different, you can start the server by running the **abaqus networkDBConnector** execution procedure on the remote system. For more information, see *Network Output Database File Connector*.

1. From the main menu bar, select **File->Network ODB Connector->Create**.
2. From the **Network ODB Connector** editor that appears, enter the name of the remote connector. When you subsequently open an output database, the **Open Database** dialog box displays the name of the remote connector. Abaqus/CAE also displays this name in the **Network ODB Connector Manager**.
3. From the **Basic** tabbed page of the **Network ODB Connector** editor, enter the following:

Host name

The name of the remote system in the form of a URL or an IP address; for example, `computeserver.mycompany.com`.

Directory

The directory to open on the remote system. The directory that you enter must contain the remote output database that you want to access, or it must include subdirectories that contain the remote output database.

4. In most cases you will be able to click **OK** to close the dialog box and to establish a remote connection using the default configuration options. However, if you have difficulty establishing communication with the remote system or if your site requires a particular configuration, you may need to customize the network ODB connector. For more information, see *Customizing a network ODB connector*.

Additional information

• *Accessing an output database on a remote computer*

Customizing a network ODB connector

If you have difficulty establishing communication with the remote system or if your site requires a particular configuration, you may need to customize the network ODB connector. Use the **Advanced** tabbed page of the **Edit Network ODB Connector** dialog box to customize a network ODB connector.

1. From the main menu bar, select **File->Network ODB Connector->Edit->*connector name***.
2. From the **Edit Network ODB Connector** dialog box that appears, click the **Advanced** tab.
3. From the **Advanced** tabbed page, choose how the server will be started.
   - Choose **Automatically start server** to indicate that Abaqus/CAE will start the network ODB server when you start the network ODB connector.
   - Choose **Use manually started server** to indicate that you have already started a network ODB server using the **abaqus networkDBConnector** execution procedure.
4. Select the shell that will be used by the local system to execute commands on the network ODB server. You must ensure that the shell command that you select is installed and found in the PATH environment variable on the local system.
   - Select **ssh** to use the secure shell command. The secure shell command uses identity authentication and encryption when communicating with the server and provides more security than the remote shell command. The ssh daemon service must be running on the remote machine.
   - Select **rsh** to use the remote shell command. The rsh daemon service must be running on the remote machine.

   Note: The secure shell command and the remote shell command must be configured so that they do not prompt the user for a password. For more information, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.
5. If you chose **Automatically start server** to indicate that Abaqus/CAE will start the network ODB server, perform the following steps:
   a. Choose from the following to specify the port numbers:
      - Choose **Auto-assign port** to allow the host and remote systems to establish their own network communication port numbers.
      - Choose **Specify port** to force the host and remote systems to use a specified port number. In the **Port** field that appears, enter the desired port number. The port number must be a valid port number. You cannot use a port number that is reserved by the system or a port that is already in use.

   b. In the **Remote Abaqus execution procedure** field, enter the command to run Abaqus on the remote system. The default command is the command that you used to start the current session of Abaqus/CAE; however, your site may have a customized command for executing Abaqus.
   c. In the **Server timeout** field, enter the network ODB server timeout in minutes. The default value is one day (1440 minutes). The server exits if it does not receive any communication from the client during the time specified. Regardless of this setting, if you started the server using Abaqus/CAE, the server exits when you end your Abaqus/CAE session. You can also stop the server by selecting **File->Network ODB Connector->Stop->*server name*** from the main menu bar.
   d. Click **OK** to create the network ODB connector and to close the editor. You still need to start the network ODB connector to make it active and to open a remote output database; for more information, see *Managing network ODB connectors*.

6. If you chose **Use manually started server** to indicate that you already started the network ODB server from the command line, perform the following steps:
   a. Enter the **Port** number returned by the **abaqus networkDBConnector** execution procedure.
   b. Click **OK** to create the network ODB connector and to close the editor. You still need to start the network ODB connector to make it active and to open a remote output database; for more information, see *Managing network ODB connectors*.

      If you started the server manually from the command line, you can close it using the **stop** parameter of the **abaqus networkDBConnector** execution procedure, or you can wait for the server to timeout. The **abaqus networkDBConnector** execution procedure is described in *Network Output Database File Connector*.

Additional information

- *Accessing an output database on a remote computer*
- *Creating a network ODB connector*

Managing network ODB connectors

Select **File->Network ODB Connector->Manager** from the main menu bar to manage your network connectors. The manager also monitors the status of your network ODB connectors.

You can use the **Network ODB Connector Manager** to create a network ODB connector. For more information, see *Creating a network ODB connector*. You can also do the following:

- Edit a network ODB connector.
- Copy a network ODB connector to another connector with a different name.
- Rename a network ODB connector.
- Delete a network ODB connector.
- Start a network ODB connector. After you create a network ODB connector, you must start it to make it active.

  Note:

  You can also start a connector by selecting **Network connectors** from the **Directory** field in the **Open Database** dialog box. Abaqus/CAE displays a list of network connectors, and you can double-click a connector to start it. You display the **Open Database** dialog box by selecting **File->Open** from the main menu bar.

- Stop a network ODB connector that you previously started. You must stop the connector before you can edit, rename, or delete it.

Additional information

• *Accessing an output database on a remote computer*

Closing the current output database

Select **File->Close ODB** from the main menu bar to close an output database. Closing an output database releases computer resources, such as memory.

1. From the main menu bar, select **File->Close ODB**.

   The **Close Output Database** dialog box appears with a list of all the output databases that are open, the date they were last updated, and the viewports that reference each open output database.

2. Select the output database to close, and click **OK** to close the dialog box.

   Abaqus/CAE closes the selected output database and clears any viewports that were displaying data from that output database.

Additional information

- *Understanding the files generated by creating and analyzing a model*
- *Using the File menu*

Setting the work directory

The work directory is the directory into which Abaqus/CAE writes files that it generates when you submit a job for analysis, such as input files and output database files.

Select **File->Set Work Directory** from the main menu bar to change the work directory. The Work Directories toolbar is updated to show the new setting. When you start an Abaqus/CAE session, the work directory is the directory from which you started Abaqus/CAE. Changing the work directory does not change the location where the replay file is saved, nor does it change the default directory for opening or saving files such as the model database file. However, Abaqus/CAE does use the new work directory to save any files that do not display a path when you save them. For example, the report file (`abaqus.rpt`) is written to the work directory.

When you use the file selection dialog boxes, you can click the work icon  to access the work directory. (The file selection dialog box displays the full path to the directory you are accessing.) For more information, see *Using file selection dialog boxes*.

Additional information

- *What is an Abaqus/CAE model database?*
- *Using the File menu*
- *Understanding the files generated by creating and analyzing a model*
- *Components of the toolbars*

Saving the current model database

Until you save the current model database for the first time, it exists only in memory.

Select **File->Save** from the main menu bar or click  in the File toolbar to save the current model database, if it is new, or to append changes made during the current session to a previously saved model database. After you save the model database, Abaqus/CAE displays its name in the title bar of the main window.

Before you save the current model database for the first time, it exists only in memory and has no name. When you save the current model database for the first time, Abaqus/CAE displays the **Save Model Database As** dialog box to allow you to enter a name; subsequent saves use this name and append changes made during the current session to the previously saved model database. If you omit the file extension, Abaqus/CAE appends `.cae` to the file name.

For information on saving the model database to a new file using a different name, see *Saving the current model database to a new file with a different name*. For more information on saving files, see *Using file selection dialog boxes*.

You should save the model database periodically. Abaqus/CAE never saves the model database unless you perform an explicit save operation; there is no timer-based automatic saving, for example. If you try to save a model database that has not been modified, no action is taken.

Abaqus/CAE asks you if you want to save a modified model database before you exit the session.

The **File->Save** command does not compress the model database even if you have deleted items from the model. To reduce the file size when you have deleted model contents, use the **File->Compress MDB** command or save the model database to a new file name (using the **File->Save As** command).

Additional information

- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *Using file selection dialog boxes*

Saving the current model database without a license

If your system loses contact with the license server or your license is otherwise lost during a session, you can save the model database in its current state. Abaqus/CAE displays a message dialog containing the following options:

- Try to reacquire a license or check to see if the server is available
- Save the model database
- Exit Abaqus without saving the model database

By default, Abaqus/CAE will attempt to reacquire a license or reconnect to the license server. To save the model database, select the second option and provide a file name; if you have already saved a model during the current session, Abaqus/CAE shows the last file name and path as the defaults to save the current model database. When you click **OK**, Abaqus/CAE saves the model.



Note:

If you are in a procedure, such as creating a sketch or editing a material, only the portions of the model completed prior to the start of the procedure can be saved.

If you choose to exit without saving the model, Abaqus/CAE will attempt to recover the model information the next time that you start a session. Once you have saved the model, the save option is removed from the dialog box—you can either continue trying to obtain a license or exit the session.

You should save the model database periodically. Abaqus/CAE never saves the model database unless you perform an explicit save operation; there is no timer-based automatic saving, for example. If you try to save a model database that has not been modified, no action is taken.

Additional information

- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *Using file selection dialog boxes*

Saving the current model database to a new file with a different name

Select **File->Save As** from the main menu bar to save the current model database to a new file with a different name. If you deleted items from your model during the current session, using **File->Save As** may decrease the size of your file (for more information about compressing files, see *Compressing the file size of the current model database*). From the **Save Model Database As** dialog box that appears, enter a new name for the model database and click **OK**. If you omit the file extension, Abaqus/CAE appends `.cae` to the file name. See *Saving the current model database*, for information on saving the model database using the same name.

Using **File->Save As** with the same file name will not decrease the size of your file.



Note:

You cannot save a model database using the name `abaqus`.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Compressing the file size of the current model database

Select **File->Compress MDB** from the main menu bar to compress the current model database (MDB). Compressing the MDB attempts to reduce the file size. The change will be most noticeable if you have deleted multiple items from your model.

Abaqus/CAE uses the compression function if you select **File->Save As** to save a file with a new file name.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model database?*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Managing models

This section describes how you manage models within the current model database.

For general information on managing objects, see *Managing objects* and *Managing objects using manager menus*.

In this section:

- *Manipulating models within a model database*
- *Opening an existing model*
- *Copying objects between models*
- *Specifying model attributes*

Manipulating models within a model database

A model database can contain many models. Although you can have only one model database open at any time, you can open more than one model at a time. The main window's title bar displays the name of the model database, and the title bar of each viewport displays the name of the model associated with the viewport. The current viewport is indicated by a dark gray title bar; the model associated with the current viewport is known as the current model. The name of the current model is also displayed in the **Model** list in the context bar.

To create a new model, select **Model->Create** from the main menu bar and enter the name of the model in the **Edit Model Attributes** dialog box that appears.

To open a model and associate it with the current viewport, select the desired model from the **Model** list in the context bar. The **Model** list contains all the models in the current model database.

To copy, rename, or delete models, select the **Copy Model**, **Rename**, or **Delete** items listed under the **Model** menu on the main menu bar. The **Copy Model**, **Rename**, and **Delete** items contain submenus listing all the models in the current model database. For general information on how to use these menus, see *Managing objects using manager menus*.

You can also create, copy, rename, and delete models using the **Model Manager**. To display the **Model Manager**, select **Model->Manager** from the main menu bar. The **Model Manager** dialog box contains functions identical to those listed under the **Model** menu but with a convenient browser that lists all the models available in the current model database. For general information on how to use managers, see *Managing objects*.

You can copy a model to a new model in a model database. In addition, you can copy objects such as sketches, parts, and materials between the models in a model database; for more information, see *Copying objects between models*. You can also copy a model from another Abaqus/CAE model database to a new model in the current model database; for more information, see *Importing a model from an Abaqus/CAE model database*.

Additional information

- *Managing models*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *Copying objects between models*
- *Managing objects*

Opening an existing model

To open a model and associate it with the current viewport, select the desired model from the **Model** list in the context bar. The **Model** list contains all the models in the current model database.

Abaqus/CAE switches to the selected model and associates it with the current viewport (indicated by a red border). The new model appears in the list of models in the context bar.

You can have multiple models open at any one time; the title bar of a viewport indicates the model associated with the current viewport. You do not have to save the current model prior to opening an existing model because Abaqus/CAE stores all models in the model database.

Additional information

- *What is an Abaqus/CAE model?*

Copying objects between models

You can copy objects such as parts, instances, materials, discrete fields, and analytical fields between the models in a model database.
Select **Model->Copy Objects** from the main menu bar to copy objects between models in the current model database.

You can copy the following objects:

- Sketches
- Parts (part sets are also copied)
- Instances (part instances and model instances)
- Materials
- Sections (including connector sections)
- Profiles
- Amplitudes
- Interaction properties
- Discrete fields
- Analytical fields
- Constraints

When you select part instances to copy, the corresponding part is also selected by default; you can deselect the part if it already exists. You cannot copy other individual objects, such as the assembly, loads, or steps; however, you can achieve a similar effect by copying the entire model to a new model and editing the objects in the new model. For more information, see *Manipulating models within a model database*. Dependent objects are not copied automatically when you copy an object between models. For example, if you copy a section, the associated material is not copied along with the section; you must copy the material in a separate copy operation.

If you are copying a part and the assembly context of the model to which you are copying the object is displayed in the viewport, the assembly will be regenerated only if an instance of the part being copied exists in the assembly of the model to which you are copying the object.

If you are copying a model instance and the target model contains an instance with the same name, the instance in the target model is first deleted and a new instance is created by copying the model instance from the source model. Any features, such as sets and surfaces, associated with the deleted instance are invalidated.

1. From the main menu bar, select **Model->Copy Objects**.

   The **Copy Objects** dialog box appears.
2. From the dialog box, select the model to copy objects from.
3. Use the following techniques to specify the objects to copy from the selected model:

   - Click the arrow next to the desired object category. From the list of objects that appears, toggle the names of the objects of your choice. An object category is unavailable if it contains no objects.
   - Toggle the desired object category. This action selects or deselects all objects within that category.

   The check box next to an object category displays a black check mark on a white background when all objects within that category are selected. The check box displays a dark gray check mark on a light gray background if only some of the objects within that category are selected. You must select at least one object or object category to copy.
4. From the bottom of the **Copy Objects** dialog box, select the model to copy the selected objects to.
5. Click **OK** to copy the selected objects and to close the **Copy Objects** dialog box.

Abaqus/CAE copies the selected objects. If an object with the same name already exists in the model to which you are copying the object, Abaqus/CAE asks for confirmation that you want to overwrite the existing object. Click **Yes to All** to overwrite all existing objects with the same name as the objects you are copying.

Additional information

- *What is an Abaqus/CAE model?*
- *Manipulating models within a model database*

Specifying model attributes

You specify the attributes that describe characteristics of the model, such as name, model description, physical constants, and restart information.

You specify the following model attributes that describe characteristics of the model:

- Name.
- Model type.
- Description of the model.
- Whether parts and assemblies should be included when you write the model to an input file.
- Physical constants for the model.
- If desired, the restart information that will start the analysis using data from a previous analysis. For more information, see *Restarting an analysis* and *Restarting an Analysis*.
- Whether the global model that will drive the submodel boundary conditions or loads. You can also specify that the global model is a shell driving a solid submodel. For more information, see *Submodeling*.
- Whether constraints, connector section assignments, and surface-to-surface contact and self-contact interactions defined in the initial step will be copied to the current working model when you create model instances from this model. For more information, see *Working with model instances*.

1. From the main menu bar, display the **Edit Model Attributes** dialog box using one of the following methods:
   - To specify model attributes in a new model, select **Model->Create** from the main menu bar.
   - To specify model attributes in an existing model, select **Model->Edit Attributes->*model name*** from the main menu bar.
2. If you are creating a new model, select the model type:
   - Select **Standard & Explicit** (default) to create a model for an Abaqus/Standard or an Abaqus/Explicit analysis.
   - Select **Electromagnetic** to create a model for an electromagnetic analysis.

   You cannot change the model type in an existing model.
3. If desired, enter or revise a description for the model.

   a. Click  in the **Edit Model Attributes** dialog box.

   The model description editor appears.

   b. In the model description editor, type information that you want to record about the model.

   c. Click **OK** to store the description and to close the model description editor.

   The description that you enter is saved in the model database and is written above the header of the input file when you submit the model for analysis; the description is not written to the output database. For more information, see *Adding descriptions to your Abaqus/CAE model*.
4. If you want Abaqus/CAE to write input files without parts and assemblies, toggle on **Do not use parts and assemblies in input files**. For more information about this option, see *Writing input files without parts and assemblies*.
5. In the **Physical Constants** portion of the dialog box, do the following:
   - To specify surface emissivity and radiation conditions in heat transfer analyses, enter values for the absolute zero temperature and the Stefan-Boltzmann constant.

- To specify the universal gas constant, enter a value in the **Universal gas constant** field.
- To identity the type of incident wave loading for an incident wave interaction in acoustic analyses, toggle on **Specify acoustic wave formulation**, click the arrow to the right of the text field, and select the formulation.
  - Select **Scattered wave** to obtain the scattered wave field solution that will be produced by incident wave loading.
  - Select **Total wave** to obtain the total acoustic pressure wave solution.

6. If desired, click the **Restart** tab to specify restart information that will start the analysis using data from a previous analysis. Toggle on **Read data from job** and do the following:
   - Type the name of the job from which Abaqus/CAE will read the restart information.
   - Type the name of the step from which Abaqus/CAE will restart the analysis.
   - Choose the increment, interval, iteration, or cycle of the step from which Abaqus/CAE will restart the analysis.
7. If desired, click the **Submodel** tab and do the following:
   - Toggle on **Read data from job** and enter the name of the output database from which the global solution will be used to drive the submodel boundary conditions or loads. You can also enter the name of a results file, if an output database is not available.
   - Specify whether the submodel will be a solid that is driven by a global shell model.

   For more information, see *Creating a submodel*.
8. By default, constraints, connector section assignments, and surface-to-surface contact and self-contact interactions defined in the initial step (along with their contact interaction properties) will be copied to the current working model when you create model instances from this model. To change this behavior, click the **Model Instances** tab and toggle off the objects that you do not want copied.
9. Click **OK** to save your data and to close the dialog box.

Additional information

- *Defining incident waves*
- *Configuring restart output requests*
- *Controlling a restart analysis*
- *Submodeling*
- *Working with model instances*
- *What is an Abaqus/CAE model?*
- *Manipulating models within a model database*

Managing session objects and session options

This section describes how you save session objects and session options to a file and how you load these objects and options for use in subsequent sessions.

In this section:

- *Saving session objects and session options to a file*
- *Loading session objects and session options from a file*

Saving session objects and session options to a file

By default, many objects and options in Abaqus/CAE persist only for the current session. To retain these session objects or session options for use in a future Abaqus/CAE session, save them to the model database, to an output database, or to a settings file in XML format.

If you save settings to an output database, Abaqus/CAE loads those settings when you open that file; if you save to a model database or a settings file, you must load the settings from that file into your session to use them.

When you save session objects or options, you can include all the session-specific settings you specified in that session, or you can select settings from individual categories. For example, you can save all the display groups and path definitions in your session to a file while excluding all view cut definitions. The **Save Session Objects & Options** dialog box saves all the definitions in a particular category when you select the category; you cannot save individual display groups to a file while excluding others. If you want only a subset of display groups to be available for use in subsequent sessions, you must delete the other display groups from your session before saving the display groups to a file.

You must pay attention to object dependencies when you save session objects and options to a file. For example, a free body cut might refer to a previously defined display group, so it would make sense to save both display groups and free body cuts if you want to retain the free body cut in the future. Likewise, if you want to save the list of active view cuts and free body cuts to a file, you should also save the view cuts and free body cuts themselves.

Session objects are typically items that you define in a session, such as a display group or view cut; while session options are typically settings in a dialog box, such as the **Common Plot Options** dialog box. You can save the following session objects:

- display groups
- paths
- *X*–*Y* data objects
- free body definitions
- view cuts including cut-specific options (in the Visualization module only)
- the active status for free body cuts and view cuts
- the currently selected view and the 11 views specified on the Views toolbar
- spectrums

You can save the following session options:

- **ODB Display Options**
- **Result Options**
- **Common Plot Options**
- **Contour Plot Options**
- **Superimpose Plot Options**
- **Material Orientation Plot Options**
- **Ply Stack Plot Options**
- **Symbol Plot Options**
- **Free Body Plot Options**
- **View Cut Options** (common cut options from the **Free Body** and **Slicing** tabbed pages only)
- **Color Mapping**

You can save session objects and options to an output database only when it has been opened with write privileges. For more information, see *Opening a model database or an output database*.

1. Select **File->Save Session Objects** from the main menu bar.

   Tip: You can also click from the File toolbar.

   The **Save Session Objects & Options** dialog box appears.

2. From the **Destination** options, select the type of file to which you want to save session objects and options and, if applicable, select the file:

   - Select **File** to save to a settings file in XML format, and specify the file name.
   - Select **MDB (.cae)** to save to the current model database.
   - Select **ODB** to save to an output database, and specify one of the output databases that are currently open in your session.

3. Specify the categories of session objects or session options that you want to save:

   - If you want to save all session objects or all session options, toggle on **Objects** or **Options**, respectively.
   - If you want to select categories of session objects or session options individually, expand the **Objects** or **Visualization Options** container, and toggle on the categories that you want to save.

4. Click **OK**. If you selected **MDB (.cae)** as the destination file, you must select **File->Save** or **File->Save As** to save the model database as well.

Additional information

- *What is an X–Y data object, and what is an X–Y plot?*
- *Creating or editing a free body cut*
- *Understanding how to create display groups*
- *Displaying a cut section and its resultant force and moment vectors*

Loading session objects and session options from a file

You can load session objects or session options from a file into your session. Session options or objects can be loaded from the current model database, an output database, or a settings file in XML format.

When you load session objects or options, you can load all the session-specific settings you specified in that file, or you can select settings from individual categories. For example, you can load all the display groups and path definitions in your session to a file while excluding all view cut definitions. The **Load Session Objects & Options** dialog box loads all the definitions in a particular category when you select the category; you cannot load individual display groups while excluding others. If you want only a subset of the display groups that are included in a file, you must load the display groups included in the file and delete the ones you do not want to use.

1. Select **File->Load Session Objects** from the main menu bar.

    **Tip:** You can also click  from the File toolbar.

   The **Load Session Objects & Options** dialog box appears.

2. From the **Source** options, do the following:

   - Select **File** to load from a settings file in XML format, and specify the file name.
   - Select **MDB (.cae)** to load from the current model database.
   - Select **ODB** to load from an output database, and specify one of the output databases that are currently open in your session.

3. Specify the categories of session objects or session options that you want to load:

   - If you want to load all session objects or all session options, toggle on **Objects** or **Visualization Options**, respectively.
   - If you want to select categories of session objects or session options individually, expand the **Objects** or **Visualization Options** container, and toggle on the categories that you want to load.

4. Click **OK**.

Additional information

- *What is an X–Y data object, and what is an X–Y plot?*
- *Creating or editing a free body cut*
- *Understanding how to create display groups*
- *Displaying a cut section and its resultant force and moment vectors*

Controlling the input file generated by Abaqus/CAE

This section describes techniques you can use to control the input file generated by Abaqus/CAE.

In this section:

- *Adding unsupported keywords to your Abaqus/CAE model*
- *Adding descriptions to your Abaqus/CAE model*
- *Resolving conflicts in the input file*
- *Writing input files without parts and assemblies*

Adding unsupported keywords to your Abaqus/CAE model

Abaqus/CAE uses your model definition to generate Abaqus/Standard or Abaqus/Explicit keywords and data that are placed in an input file when you submit the analysis job. Currently Abaqus/CAE may not support Abaqus/Standard or Abaqus/Explicit functionality that you might like to include in your model. If that is the case, you may be able to add the functionality using the **Keywords Editor**. Select **Model->Edit Keywords->*model _name*** from the main menu bar to start the **Keywords Editor**.

To use the **Keywords Editor**, you should be familiar with the syntax of Abaqus keywords and data. For example, the Step module does not allow you to provide boundary impedances or nonreflecting boundaries for acoustic and coupled acoustic-structural analysis. To provide boundary impedances or nonreflecting boundaries, you can use the **Keywords Editor** to add the **IMPEDANCE* keyword to the model.

When you submit the model for analysis in the Job module, Abaqus/CAE incorporates changes you made using the **Keywords Editor** in the input file that is submitted for analysis. Keywords that you add to your model using the **Keywords Editor** persist even after you modify or regenerate the model using Abaqus/CAE, because Abaqus/CAE stores the contents of the **Keywords Editor** along with the model definition in the model database. As a result, the contents of the input file may not be valid (for example, if you added keywords that refer to steps that you subsequently deleted), and the analysis may fail.



Warning:

If you used the **Keywords Editor** to edit the original model, Abaqus/CAE ignores those changes when you restart the analysis. For more information, see *Rules governing a restart analysis*.

Abaqus/CAE indents some of the keywords in the **Keywords Editor** to make the input file easier to read on the screen. The keywords that are indented can never appear on their own in the input file and must always be used in conjunction with another keyword. The indentation that you see in the **Keywords Editor** is not included in the input file that is generated by Abaqus/CAE.

The **Keywords Editor** does not allow you to edit the geometry of your model; you must use Abaqus/CAE to make geometry changes. Therefore, the **Keywords Editor** is available only after you have generated the mesh. If the input file generated by Abaqus/CAE is exceptionally long, the **Keywords Editor** displays the input file one buffer at a time. You can use the scroll bar to scroll through each buffer and the controls at the bottom of the editor to navigate to different buffers.

If you use the **Keywords Editor** to modify the input file and then use Abaqus/CAE to modify the model, Abaqus/CAE merges the changes into the input file. If a conflict arises during the merge, Abaqus/CAE issues a warning message.

It is recommended that you not edit keywords that are supported by Abaqus/CAE; for example, you should use the Property module, not the **Keywords Editor**, to change the properties of a material. This approach maintains consistency between directly supported aspects of a model and those added by the **Keywords Editor**. If you do edit a keyword using the **Keywords Editor** and then use Abaqus/CAE to make a change to your model that refers to the same keyword, a conflict will occur.

To minimize the length of the input file displayed in the **Keywords Editor**, Abaqus/CAE hides the data lines for keywords that require a large amount of data, notably **NODE*, **ELEMENT*, and **DISTRIBUTION*. If you modify the line immediately following such keywords in the **Keywords Editor**, conflicts occur when Abaqus/CAE merges and writes the input file.

For tips on resolving input file conflicts generated by the **Keywords Editor**, see *Resolving conflicts in the input file*.

You can review the keywords supported by Abaqus/CAE by selecting **Help->Keyword Browser** from the main menu bar.

1. From the main menu bar, select **Model->Edit Keywords->*model _name***.

   The **Keywords Editor** appears and displays the keywords associated with the model you select.



Note:

The keywords are available to be edited only after you have generated a mesh.



Note:

The contents are truncated if you include an asterisk (*) in an object name or description.

2. Each keyword in the input file is displayed in its own block. Buttons in the lower left corner of the **Keywords Editor** allow you to do the following:

Add After

Add an empty block of text below the selected block; text that you add to a new block appears in blue.

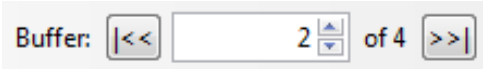
Remove

Remove the selected block of text that was added using the **Keywords Editor**. You cannot remove a block generated by Abaqus/CAE.

Discard Edits

Discard the changes you made to a block generated by Abaqus/CAE during the most recent use of the **Keywords Editor**.

Buffer *x* of *y* :



Display the buffer of the input file as specified in the box. Use the scroll bar to scroll within a buffer.

|<<

Display the first buffer of the input file.

>>|

Display the last buffer of the input file.

In addition, you can click any block and edit the text inside. Cyan text indicates a block generated by Abaqus/CAE that you edited.

3. From the buttons across the bottom of the **Keywords Editor**, click **OK** to include your changes and to close the editor. Click **Cancel** to close the editor and to disregard your changes. Click **Discard All Edits** to keep the editor open and to remove all of the changes that you made to the input file.

Additional information

- *What is an Abaqus/CAE model?*
- *Abaqus keyword browser table*

Adding descriptions to your Abaqus/CAE model

In Abaqus/CAE you can enter descriptions for the model and for the materials in your model in the **Edit Model Attributes** dialog box and the **Edit Material** dialog box, respectively. When you submit the job for analysis, Abaqus/CAE generates an input file and writes these descriptions to the input file using comment lines. The comment lines for the model description immediately precede the header of the input file, and the comment lines for the material descriptions immediately precede the material definitions.

For related information on using descriptions, see *Importing descriptions*.

Additional information

- *Specifying model attributes*
- *Creating or editing a material*

Resolving conflicts in the input file

If you edit a keyword using the **Keywords Editor** and then use Abaqus/CAE to make a change to your model that refers to the same keyword, Abaqus/CAE cannot determine which version of the keyword to incorporate in the input file and writes text to the input file signaling the problem. As a result, an error is generated when you submit the model for analysis. If you display the input file using the **Keywords Editor**, any keywords or data lines that conflict are indicated by a *Conflicts statement. In addition, the *Conflicts statement indicates whether the text was generated by Abaqus/CAE or by the **Keywords Editor**. You should use the **Keywords Editor** to remove any unwanted keywords or data lines. You should also remove all the *Conflicts statements.

Certain input file conflicts cannot be resolved using the **Keywords Editor**; for example, conflicts caused by modifications to the input file line immediately following hidden data lines (see *Adding unsupported keywords to your Abaqus/CAE model*). In these situations you can use the Job module to write the complete input file to a text file (see *Writing the input file only*, for details). You can modify this input file using a text editor, then manually submit the modified input file for analysis (see *Abaqus/Standard and Abaqus/Explicit Execution*). To perform the analysis in Abaqus/CAE, create a new job in the Job module using the modified input file as the job source (see *Creating a new analysis job*).

Writing input files without parts and assemblies

Abaqus/CAE uses your model definition to generate an Abaqus/Standard or Abaqus/Explicit input file when you submit the analysis job.
The Abaqus/CAE model contains parts and assemblies; and, by default, an input file generated by Abaqus/CAE contains parts and assemblies. Some Abaqus functionality is not supported in a model that contains parts and assemblies.

Abaqus/CAE attempts to preserve the node and element labels of the model when writing an input file without parts and assemblies. If there are no conflicts between any part or part instance labels, Abaqus/CAE maintains the labels in the model when it writes the nodes and elements to the input file. Conversely, if any conflicts arise between any part or part instance labels, such as two part instances with the same node and element labels, Abaqus/CAE displays a warning before it writes an input file with renumbered node and element labels.

If you want Abaqus/CAE to write input files without parts and assemblies, you can use one of the following methods to change the format of input files generated by Abaqus/CAE:

When you start an Abaqus/CAE session

To change the format of the input files generated by Abaqus/CAE, you can modify the **cae_no_parts_input_file** parameter in the Abaqus environment file (`abaqus_v6.env`) as follows:

```
cae_no_parts_input_file=ON
```

If you use this method, the input file format cannot be changed during the Abaqus/CAE session. For more information on defining environment file parameters, refer to the *Abaqus Configuration Guide*.

During an Abaqus/CAE session

There are two methods that you can use to change the format of the input files generated by Abaqus/CAE.

- Select **Model->Model Attributes->*model name*** for the model that you want to change, then toggle on **Do not use parts and assemblies in input files**.
- Enter the following Abaqus Scripting Interface command into the command line interface at the bottom of the Abaqus/CAE main window:

  ```
  mdb.models[modelName].setValues(noPartsInputFile=ON)
  ```

  Note:

  The command line interface is hidden by default, but it uses the same space that is occupied by the message area at the bottom of the main window. To access the command line interface, click  in the bottom left corner of the main window.

If you use either method, the changed input file format is a part of the model definition. When you exit the Abaqus/CAE session and return to the model database at a later time, the changed input file format is retained.

If you find yourself repeatedly changing the format of the input file generated by Abaqus/CAE, you can create and run a macro that contains the previous Abaqus Scripting Interface command. For more information on macros, see *Managing macros*.



Warning:

When Abaqus/CAE writes an input file without parts and assemblies, it tries to preserve the node and element labels generated in Abaqus/CAE. However, if you use the **Keywords Editor** to edit keywords in a model that uses the parts and assemblies input file format and then change the format of the input file generated by

Abaqus/CAE, conflicts may occur when the input file is written and the keywords are regenerated. For more information, see *Resolving conflicts in the input file*.

Additional information

- *The Abaqus/CAE main window*
- *Managing macros*

Managing macros

When you create a macro, Abaqus/CAE records a sequence of Abaqus Scripting Interface commands in a macro file while you interact with Abaqus/CAE. Each command corresponds to an interaction with Abaqus/CAE, and replaying the macro reproduces the sequence of interactions.
To manage macros containing a set of Abaqus Scripting Interface commands, select **File->Macro Manager** from the main menu bar.

Macros are stored in a file called `abaqusMacros.py`. Abaqus/CAE searches three directories for `abaqusMacros.py`, in the following order:

- The site directory of the Abaqus installation.
- Your home directory.
- The current working directory.

The `abaqusMacros.py` file can exist in more than one of these directories. The **Macro Manager** contains a list of the existing macros that Abaqus/CAE detected in all the `abaqusMacros.py` files. If a macro uses the same name in more than one `abaqusMacros.py` file, Abaqus/CAE uses the last macro encountered.

Your macro will run only in the same context in which it was recorded. For example, if you create a macro that copies a part named `gear1` to a new part named `gear2`, the macro will be executed in a new Abaqus/CAE session only if a part named `gear1` exists. When using a plug-in, you may be required to import additional Python modules only the first time you access the plug-in, and this initialization may not be recorded in the macro.

The Abaqus Scripting Interface commands are stored in ASCII text. You can edit `abaqusMacros.py` with a standard text editor; however, any errors that you introduce into the file will prevent the **Macro Manager** from displaying. For more information on commands, see the *Abaqus Scripting User's Guide*.



Tip: If you edit any of the `abaqusMacros.py` files using a text editor, you can click **Reload** from the buttons at the bottom of the **Macro Manager** to update the macro list without closing the manager.

Additional information

• *Creating and running a macro*
• *Abaqus Scripting User's Guide*

Create a macro

1. From the main menu bar, select **File->Macro Manager**.

   The **Macro Manager** dialog box appears.
2. From the buttons across the bottom of the **Macro Manager** dialog box, click **Create**.
3. Enter a name for the macro in the **Create Macro** dialog box that appears, and click **Continue**. You cannot overwrite an existing macro.

   Each of your interactions with Abaqus/CAE is stored as a command in the `abaqusMacros.py` file. A **Recording macro** dialog box appears to remind you the macro is recording. In addition, the **Create**, **Delete**, **Run**, and **Reload** buttons are not available in the **Macro Manager** while the macro is recording.
4. Click the **Stop recording** button to save the macro in `abaqusMacros.py`.

   Abaqus/CAE updates the macro list in the **Macro Manager**.

Delete a macro

1. From the main menu bar, select **File->Macro Manager**.

The **Macro Manager** dialog box appears.

2. Select the macro to delete. You can select more than one macro.
3. From the buttons across the bottom of the **Macro Manager** dialog box, click **Delete**.
4. From the dialog box that appears, click **OK** to confirm your action.

   Abaqus/CAE deletes the macro from `abaqusMacros.py` and updates the macro list in the **Macro Manager**. You cannot recover a deleted macro.

Run a macro

1. From the main menu bar, select **File->Macro Manager**.

   The **Macro Manager** dialog box appears.
2. Select the macro to run.
3. From the buttons across the bottom of the **Macro Manager** dialog box, click **Run**. You can run only one macro; the **Run** button is not available if you selected more than one macro.

   Abaqus/CAE runs the commands in the selected macro and displays a message in the message area when the macro execution completes.

Importing and exporting geometry data and models

This section describes the files that can be imported into and exported from Abaqus/CAE.

In this section:

- *Importing files into and exporting files from Abaqus/CAE*
- *Valid parts, precise parts, and tolerance*
- *Controlling the import process*
- *Understanding the contents of an IGES file*
- *What can you import from a model?*
- *A logical approach to successful import of IGES files*
- *Importing sketches and parts*
- *Importing a model*
- *Exporting geometry, model, and mesh data*

Importing files into and exporting files from Abaqus/CAE

Abaqus/CAE can import part and assembly geometries from a variety of external sources and CAD systems. The associative interfaces for Abaqus/CAE provide straightforward and powerful techniques for importing geometry. More traditional techniques are also available for importing and exporting geometry using standard CAD file formats. Understanding the capabilities of each format, as well as the limitations of representing the geometry of a part in a file, will help you select the import or export technique that is appropriate for your application.

In this section:

- *What kinds of files can be imported and exported from Abaqus/CAE?*
- *What can I do with the associative interfaces?*
- *What can I do with the Elysium plug-ins?*
- *Importing an assembly*
- *Know what you want as an end product*
- *Vendors interpret the standards differently*
- *How do solid modelers represent a solid?*

What kinds of files can be imported and exported from Abaqus/CAE?

Abaqus/CAE reads and writes geometry data stored in Abaqus file formats as well as non-Abaqus file formats.

Abaqus file formats

Abaqus/CAE reads and writes geometry data stored in the following Abaqus file formats:

Abaqus output database (*output_database_name*`.odb`)

An output database contains the data generated during an Abaqus/Standard or Abaqus/Explicit analysis. You can import parts from an output database in the form of meshes. A mesh part contains no feature information and is extracted from the output database as a collection of nodes, elements, surfaces, and sets. If the output database contains multiple part instances, you can select the part instances to import. Abaqus/CAE imports each part instance as a separate part. You can import either the undeformed or the deformed shape. If you import the deformed shape, you can specify the step and the frame from which to import.

To verify the quality of the mesh, you can display the part in the Mesh module and select **Mesh->Verify** from the main menu bar. In addition, you can use the Mesh module to change the element type assigned to the mesh and to edit the original mesh definition. For more information, see *Importing a part from an output database*, *What can I do with the Edit Mesh toolset?*, and *Assigning Abaqus element types*.

You can also import a model from an output database. The model that is imported will contain parts representing each of the undeformed part instances in the output database along with a mesh representation of the undeformed assembly. The model will also contain any sets, surfaces, materials, section definitions, and beam profiles that were defined in the output database. For more information, see *Importing a model from an output database*.

Abaqus/CAE model database (*model_database_name*`.cae`)

Abaqus/CAE can import a model into the current model database from a different Abaqus/CAE model database. For more information on importing model data from another model database, see *Importing a model from an Abaqus/CAE model database*.

Abaqus/Standard and Abaqus/Explicit input files

Abaqus/CAE generates an input file when you submit a job for analysis. You can import input files into Abaqus/CAE. Abaqus/CAE translates the keywords and data lines in the imported input file into a new model; however, a limited set of Abaqus/Standard and Abaqus/Explicit keywords is supported, as described in *Importing a model from an Abaqus input file*. For more information on creating and submitting jobs, see *Basic steps for analyzing a model*.

Abaqus substructure files (*substructure_name*`.sim`)

Abaqus/CAE can import a substructure definition from a SIM database as a new part definition. The `.sim` file you import must reside in the same directory as the supporting Abaqus files to which the SIM database refers; these supporting files might include data in the formats `.prt`, `.mdl`, or `.stt`. For step-by-step instructions, see *Importing a substructure into a model database as a part*.

Supported non-Abaqus file formats

Abaqus/CAE reads and writes geometry data stored in the following non-Abaqus file formats:

3D XML (*file_name*`.3dxml`)

3D XML is an XML-based format developed by Dassault Systèmes for encoding three-dimensional images and data. The format is open and extendable, allowing three-dimensional graphics to be easily shared and integrated into existing applications and processes. 3D XML files can be many times smaller than typical model database files. The 3D XML Player from Dassault Systèmes is required to view 3D XML files or to integrate them into business applications. You can also view 3D XML files in CATIA V5. This export capability cannot be used to transfer geometry or models to **3D**EXPERIENCE platform apps.

You can export viewport data from Abaqus/CAE in 3D XML or compressed 3D XML format. For more information, see *Exporting viewport data to a 3D XML-format file*. You cannot import 3D XML into Abaqus/CAE.

ACIS (*file_name*`.sat` or *file_name*`.asat`)

ACIS is a library of solid modeling functions, and most CAD products can generate ACIS-format parts. You can import ACIS-format parts, and you can export parts or the assembly in ACIS format. In addition, you can import and export sketches in ACIS format. For more information, see *Importing parts from an ACIS-format file*, *Importing sketches*, and *Exporting geometry, model, and mesh data*.

ANSYS input files (*file_name*`.cdb`)

ANSYS Mechanical and ANSYS Multiphysics software are computer-aided engineering products that allow you to perform finite element analysis and computational fluid dynamics. You can import models from files in ANSYS input file format into Abaqus/CAE. For more information, see *Importing a model from an ANSYS input file*, and *Importing a model*.

AutoCAD (*file_name*`.dxf`)

Two-dimensional profiles stored in AutoCAD (`.dxf`) files can be imported as stand-alone sketches. However, Abaqus/CAE supports only a limited number of AutoCAD entities, and you should use this format only if no other formats are available. For more information and details on the AutoCAD entities supported by Abaqus/CAE, see *Importing sketches*. Unsupported entities include:

- Polyline
- Spline
- Block
- Dimension
- Shape

CATIA V5 parts and assemblies (*file_name*`.CATPart` or `.CATProduct`)

With the optional CATIA V5 Associative Interface add-on feature for Abaqus/CAE, you can import CATIA V5-format parts and assemblies. For more information, see *Importing a part from a V5-format file*. You cannot export parts from Abaqus/CAE in CATIA V5 format.

CATIA V6 parts and assemblies (*file_name*`.CATPart` or `.CATProduct`)

You use the CATIA V6 Associative Interface to import CATIA V6-format parts and assemblies. The CATIA V6 parts and assemblies are first converted to CATIA V5 format, and Abaqus/CAE imports the resulting CATIA

V5`CATPart` or `CATProduct` files. For more information, see *Importing a part from a V5-format file*. You cannot export parts from Abaqus/CAE to CATIA V6.

Elysium assembly files (*file_name* `.eaf`)

Elysium assembly files are created by the associative interface applications, which are plug-ins for third-party CAD systems that allow you to transfer models from the CAD system to Abaqus/CAE using a technique called associative import (see *What can I do with the associative interfaces?*). The associative interface plug-ins save model information in the assembly file, and you can use the assembly file to associatively import the model from the third-party CAD system into Abaqus/CAE. For more information, see *Importing an assembly from an Elysium assembly file*. You cannot export assemblies from Abaqus/CAE in Elysium assembly file format.

IGES (*file_name* `.igs` or `.iges`)

The Initial Graphics Exchange Specification (IGES) is a neutral data format designed for graphics exchange between computer-aided design (CAD) systems.

You can import IGES-format parts, and you can export parts in IGES format. In addition, you can import and export sketches in IGES format. For more information, see *Importing a part from an IGES-format file*; *Importing sketches*; and *Exporting geometry, model, and mesh data*.

The IGES-format allows for many interpretations, and most of the parts that you import into Abaqus/CAE using IGES-format will need to be repaired before you can use them. Thus, it is recommended that you try to use another format, if possible.

Nastran input files (*file_name* `.bdf`, *file_name* `.dat`, *file_name* `.nas`, *file_name* `.nastran`, *file_name* `.blk`, or *file_name* `.bulk`)

You can import Nastran model data from a Nastran input file into Abaqus/CAE, and you can export data from an Abaqus/CAE model and job into Nastran bulk data file format. Imported and exported models include many common entities in the Nastran bulk data. For more information on supported entities for import of Nastran input files into Abaqus/CAE, see *Translating Nastran Bulk Data in Text Files to Partial Abaqus Input Files*. For more information on supported entities for export of Abaqus/CAE jobs and models to Nastran, see *Translating an Abaqus Input File to a Partial Nastran Bulk Data Text File*.

NX Elysium Neutral File (*file_name* `.enf_abq`)

Abaqus provides a translator plug-in for NX that will generate a geometry file using the Elysium Neutral File (`.enf`) format. You can use Elysium Neutral Files to import NX parts. In addition, you can use Elysium Neutral Files to import an entire NX assembly into the Abaqus/CAE assembly, or you can choose to import only selected part instances from the assembly. For more information, see *Importing a part from an Elysium Neutral file*, and *Importing an assembly from an Elysium Neutral file*. You cannot export parts or assemblies from Abaqus/CAE in Elysium Neutral File format.

OBJ (*file_name* `.obj`)

OBJ is an open file format that describes the geometry in terms of the position of the vertices, the edges between them, and the faces that comprise each polygon in the geometry. Data in OBJ format are saved as a text file.

You can export geometry data or mesh data from Abaqus/CAE in OBJ format. For more information, see *Exporting viewport data to an OBJ-format file*. You cannot import OBJ-format data into Abaqus/CAE.

Parasolid (*file_name*`.x_t`, *file_name*`.x_b`, *file_name*`.xmt_txt`, or *file_name*`.xmt_bin`)

Parasolid is a library of solid modeling functions developed by UGS. A variety of CAD products can generate Parasolid-format parts, such as NX, SOLIDWORKS, Solid Edge, FEMAP, and MSC.Patran. You can import Parasolid-format parts. You can also import an entire Parasolid assembly into the Abaqus/CAE assembly, or you can choose to import only selected part instances. For more information, see *Importing a part from a Parasolid-format file* and *Importing an assembly from a Parasolid-format file*. You cannot export parts or assemblies from Abaqus/CAE in Parasolid format.

Pro/ENGINEER Elysium Neutral File (*file_name*`.enf_abq`)

Abaqus provides a translator plug-in for Pro/ENGINEER that will generate a geometry file using the Elysium Neutral File (`.enf`) format. You can use Elysium Neutral Files to import Pro/ENGINEER parts. In addition, you can use Elysium Neutral Files to import an entire Pro/ENGINEER assembly into the Abaqus/CAE assembly, or you can choose to import only selected part instances from the assembly. For more information, see *Importing a part from an Elysium Neutral file* and *Importing an assembly from an Elysium Neutral file*. You cannot export parts or assemblies from Abaqus/CAE in Elysium Neutral File format.

SOLIDWORKS parts and assemblies (*file_name*`.sldprt` or *file_name*`.sldasm`)

You can import SOLIDWORKS native file formats for parts and assemblies. For more information, see *Importing a part from a SOLIDWORKS-format file*. You cannot export parts from Abaqus/CAE in SOLIDWORKS format.

STEP (*file_name*`.stp` or `.step`)

The STandard for the Exchange of Product model data (STEPISO 10303–1) is designed as a high-level replacement for IGES that attempts to overcome some of the shortcomings of IGES. The STEPAP203 standard is designed to provide a computer-interpretable representation of a mechanical product throughout its lifecycle, independent of any particular system.

You can import STEP-format parts, and you can export parts in STEP format. In addition, you can import and export sketches in STEP format. For more information, see *Importing a part from a STEP-format file* and *Exporting geometry, model, and mesh data*.

STEP-format parts are similar to IGES-format parts in that most of the parts that you import into Abaqus/CAE using STEP-format will need to be repaired before you can use them. Thus, it is recommended that you try to use another format, if possible.

VDA-FS (*file_name*`.vda`)

The Verband der Automobilindustrie Flächen Schnittstelle (VDA-FS) surface data format is a geometry standard developed by the German automotive industry. Both VDA-FS and IGES files contain a mathematical representation of the part in an ASCII format; however, the VDA-FS standard concentrates on geometry information. Additional information covered by the IGES standard, such as dimensions, text, and colors, is not stored in a VDA-FS file.

You can import VDA-FS-format parts, and you can export parts in VDA-FS format. For more information, see *Importing a part from a VDA-FS-format file* and *Exporting geometry, model, and mesh data*.

VDA-FS format parts are similar to IGES-format parts in that most of the parts that you import into Abaqus/CAE using VDA-FS format will need to be repaired before you can use them. Thus, it is recommended that you try to use another format, if possible.

VRML (*file_name* `.wrl`)

Virtual Reality Modeling Language (VRML) is the ISO standard for displaying three-dimensional images in a web browser or a stand-alone VRML client. It is an open, platform-independent, vector-based, three-dimensional modeling language that encodes computer-generated graphics to allow them to be shared easily across a network. VRML files use meters for all lengths and distances. VRML-format files can be many times smaller than typical model database files. A special plug-in viewer, such as Cortona or Cosmo, is required to view VRML files.

You can export viewport data from Abaqus/CAE in VRML format or compressed VRML format. For more information, see *Exporting viewport data to a VRML-format file*.

What can I do with the associative interfaces?

The associative interfaces are optional add-on products that simplify the process of transferring model data between CAD systems and Abaqus/CAE. The associative interfaces use the CAD Connection toolset to create a connection between Abaqus/CAE and a CAD system running an associative interface plug-in. Associative interface plug-ins are available for the following CAD systems:

- CATIA V6
- CATIA V5
- NX (Unigraphics)
- SOLIDWORKS
- Pro/ENGINEER

The associative interfaces allow for associative import of models from a CAD system to Abaqus/CAE. You can export an entire assembly from the CAD system into Abaqus/CAE with a single mouse click, and you can modify your model in the CAD system and use the connection to quickly update the model in Abaqus/CAE. Features that you created in Abaqus/CAE—such as loads, boundary conditions, sets, and surfaces—are updated when you import the modified model. In addition to modified parts, any changes that you make to the position of instances in the assembly will also be exported to Abaqus/CAE.

Associative import is useful when you are running the CAD system and Abaqus/CAE on the same computer and are iterating on the design of the model based on the results of the analysis. *Figure 1* shows the connection between SOLIDWORKS and Abaqus/CAE using associative import.

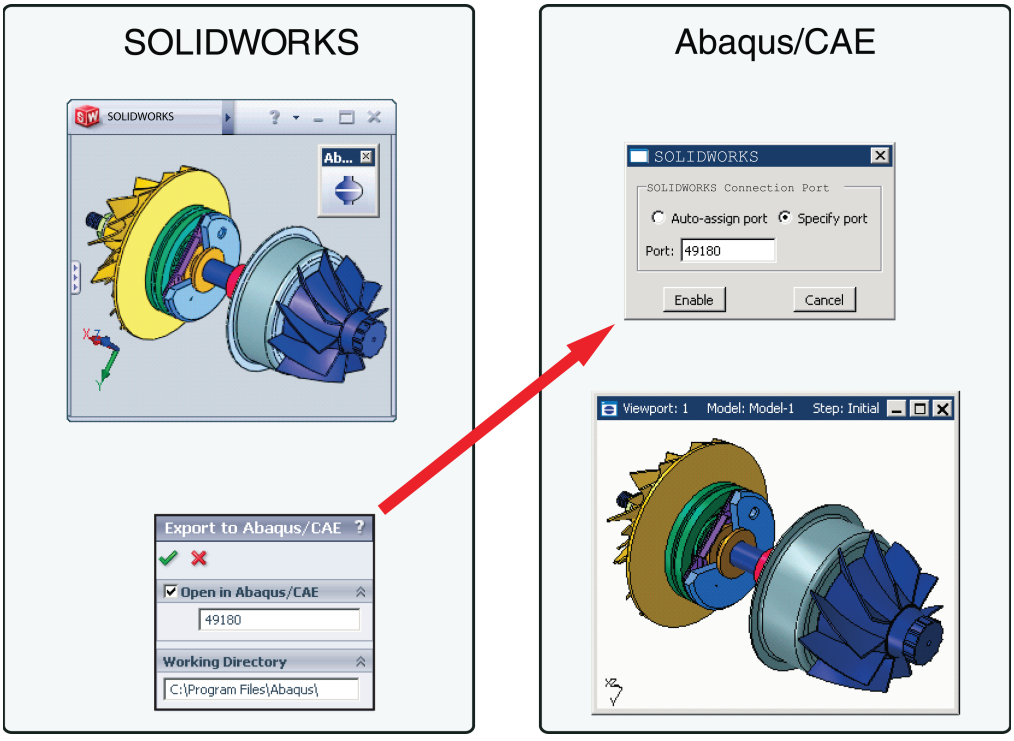


Figure 1: Associative import using the SOLIDWORKS Associative Interface.

You can modify the model with either the CAD system or Abaqus/CAE while you iterate on the design. If you import a model into Abaqus/CAE from the CAD system and then add features to the model in Abaqus/CAE, Abaqus/CAE regenerates these features in the model the next time you import it from the CAD system. Geometry modifications that you make in Abaqus/CAE (such as partitions, fillets, etc.) are also regenerated each time you import the model from

the CAD system. Abaqus/CAE may fail to regenerate features if the changes that you make using the CAD system significantly change the topology of the model. Geometry modifications made in Abaqus/CAE are not propagated to the original CAD model, although the Pro/ENGINEER Associative Interface does provide a method for revising certain geometric entities in the original Pro/ENGINEER model from within Abaqus/CAE (see *Updating geometry parameters in an imported model*).

The associative interfaces also allow you to save assemblies in a CAD system to an assembly file format that can subsequently be imported to Abaqus/CAE at a later time. For example, the CATIA V5 Associative Interface saves your CATIA V5 Assembly (`.CATProduct`) in an assembly file (`.eaf`) format that you can manually import into Abaqus/CAE. Similarly, the CATIA V6 Associative Interface converts your CATIA V6 product to a CATIA V5 Assembly and then saves it in an assembly file format that you can manually import into Abaqus/CAE.

For more information about creating a connection between Abaqus/CAE and CAD systems, see *The CAD Connection toolset*. For information about the versions of the CAD software supported by the associative interfaces, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

What can I do with the Elysium plug-ins?

You can use a Neutral File-based translator from Elysium, Inc., to import parts into Abaqus/CAE that were created by CAD software that creates Parasolid-format files, such as SOLIDWORKS, NX, Solid Edge, FEMAP, and MSC.Patran.

You can also use the Elysium translator to import the entire assembly or only selected part instances from the assembly.

In addition, you can import a part, the assembly, or selected part instances from the assembly into Abaqus/CAE using the Elysium Neutral File format. Abaqus provides a translator plug-in from Elysium that will generate a geometry file using the Elysium Neutral File format. The plug-in is available for the following products:

- CATIA V5
- NX
- Pro/ENGINEER

For information about the versions of the CAD software supported by the Elysium translators, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Importing an assembly

A file from a CAD system, such as CATIA, can contain a single part or an assembly of parts. Abaqus/CAE allows you to select **File->Import** from the main menu bar and choose either **Part** or **Assembly**. Both options allow you to import all of the parts in an assembly; however, the end result is slightly different.

Importing parts

If you choose to import parts from a file that contains an assembly of parts, you can import either all of the parts from the file or only a specified part. If you import all of the parts, Abaqus/CAE creates a group of parts that corresponds to each part instance in the original assembly. To recreate the original assembly, you must use the Assembly module to instance each imported part. However, the relationship between the parts and the part instances in the original assembly is lost during the import process. For example, if the original assembly contained a bolt that was instanced nine times, Abaqus/CAE creates nine identical parts. When you recreate the assembly in the Assembly module, Abaqus/CAE creates a part instance for each of the nine bolts. Although the relationship between the parts and part instances is lost, Abaqus/CAE does retain the position of the parts. As a result, when you instance each part, it appears in the correct position in the assembly.

Importing an assembly

If you choose to import an assembly, you can import the entire assembly or you can import only selected part instances. Abaqus/CAE appends your selection to the existing assembly and retains the original positioning of the instances. In addition, Abaqus/CAE creates parts that correspond to the imported part instances and maintains the relationship between the parts and their instances. For example, if a bolt is instanced nine times in the assembly, Abaqus/CAE imports nine instances of the bolt but creates only a single part.

Importing an assembly also offers the following advantages:

- In most cases Abaqus/CAE retains the names of the parts and the part instances from the original file.
- If the parts and part instances in the original file were colored by the third-party CAD system, Abaqus/CAE retains those colors during the import procedure. For information about modifying the color coding, see *Color coding geometry and mesh elements*.

Know what you want as an end product

Many of the problems associated with importing a complex solid part into Abaqus/CAE can be alleviated if you recognize what you want as an end product: a finite element mesh of the parts to be analyzed using Abaqus/Standard or Abaqus/Explicit.

A small feature in the imported part will result in a fine mesh in the area of the detail. The fine mesh will influence the mesh in adjacent regions and may dominate the time taken to perform the analysis. If you are not interested in analyzing the feature, you should use the CAD system to remove the detail from the part before you import the part into Abaqus/CAE. Removing small features may solve precision errors in the imported part. Examples of small features include:

- Fillets
- Chamfers
- Holes

Simplifying a solid part will increase your chances of successfully importing it into Abaqus/CAE. You must decide the level of detail that will produce meaningful results from the analysis.

Finally, you should consider the type of mesh required by the analysis. If you plan to mesh the part with triangular or tetrahedral elements or with quadrilateral elements generated by the advancing front algorithm, you can use the Virtual Topology toolset in the Mesh module to remove small details from the part before you generate the mesh. For more information, see *The Virtual Topology toolset*.

Vendors interpret the standards differently

Using an accepted industry standard, such as IGES and VDA-FS, to exchange geometric information between a CAD system and Abaqus/CAE is not a guarantee of success. When a CAD system exports a part, the system maps its proprietary representation of the part into an array of entities available from the standard. Similarly, when Abaqus/CAE imports the file, it converts the entities defined by the standard into its internal representation—ACIS.

ACIS recognizes only some of the entities defined in the IGES and VDA-FS standards and also expects a certain level of smoothness or continuity in the trimmed surfaces. Although the exporting CAD system is not aware of the requirements of Abaqus/CAE, setting the correct export options will increase your chance of success. For more information, see *How do solid modelers represent a solid?*, and *Understanding the contents of an IGES file*.

In addition, you may experience problems because CAD systems interpret the industry standards differently. In many cases there is more than one way to define a geometric entity, resulting in a particular “flavor” of the file format. In more extreme cases, vendors violate the standard, especially when creating trimmed surfaces.

How do solid modelers represent a solid?

Abaqus/CAE provides tools that allow you to import a sketch, a part, or an assembly into the current model. When you import a sketch or a planar part or assembly, the process is usually straightforward. However, when you import a solid, you may find that you have to perform some additional steps to obtain a satisfactory result. To understand what the steps accomplish and why you need to perform them, you need to understand how solids are represented by solid modelers.

As modeling systems have evolved during the last 30 years, the techniques used to represent a solid body have also evolved. Each new generation of modelers incorporates more knowledge about the construction of the body and relies less on the sheer volume of data points to describe a solid.

Wireframe

The original CAD systems used two-dimensional wireframes to replicate traditional mechanical drawings. Later versions introduced three-dimensionality in the form of isometric views and perspective views. In a wireframe a solid body is represented by a set of curves that define the edges of the body; however, the system has no information about the surfaces between the edges. A wireframe model defines an object by its edges and vertices; as a result the wireframe representation of a solid has limited use. For example, you cannot calculate the volume of the solid, and you cannot mesh the solid.

Trimmed surface

Later systems introduced the concept of trimmed surfaces, as shown in *Figure 1*.

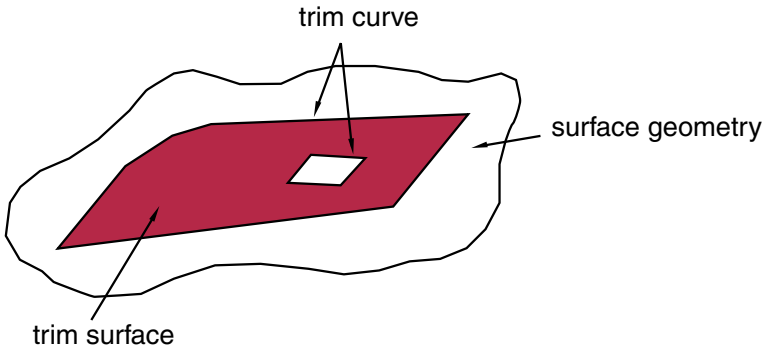


Figure 1: A trimmed surface.

A trimmed surface is defined by a combination of surface geometry and trim curves. The surface geometry is a general expression for the surface; the trim curves form a closed loop on the surface geometry and define the boundary of the surface. A surface can also have multiple internal trim curve surfaces. A solid is defined as a set of trimmed surfaces that form an enclosed volume.

For example, a cylinder can be defined from three trimmed surfaces as shown in *Figure 2*.

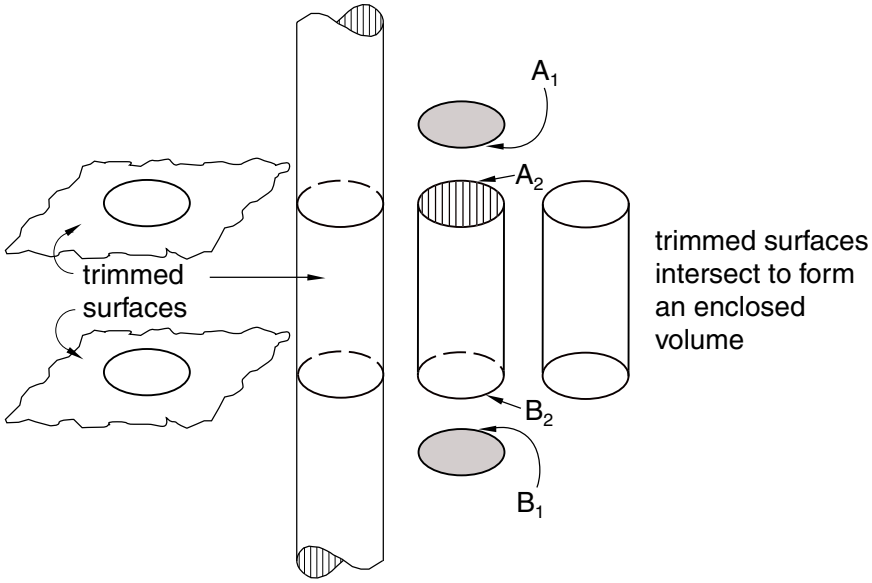


Figure 2: Three trimmed surfaces define a cylinder.

Each surface includes a set of edges that define the boundary of the surface. The flat trimmed surface at the top of the cylinder contains trim curve A_1. The flat trimmed surface at the base of the cylinder contains trim curve B_1. The cylindrical trimmed surface is bounded by two trim curves—A_2 and B_2. When two trimmed surfaces intersect, the definition of the resulting edge is duplicated by each surface. There is no information to indicate that a group of trimmed surfaces comprise a solid.

The edges of the solid are well defined when the intersecting surfaces are planar, cylindrical, or spherical. However, when more complex curves intersect, the resulting edge must be described by a polynomial expression that approximates the intersection of the two faces. The accuracy of the approximation depends on the order of the polynomial expression. If an edge does not lie on either surface, a gap is created and the solid is considered invalid. *Figure 3* illustrates a gap between trimmed surfaces.

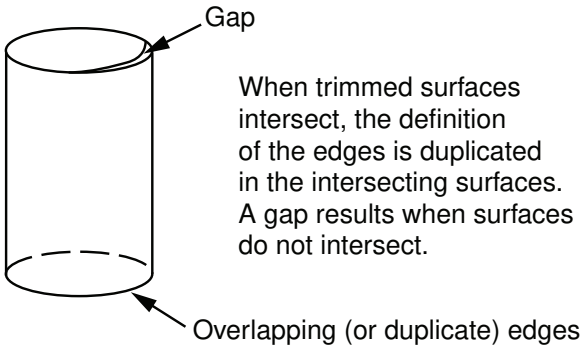


Figure 3: A gap between trimmed surfaces.

You can use the Geometry Edit toolset to stitch gaps. For more information, see *What is stitching?*.

Both the IGES and VDA-FS standards use trimmed surfaces. Most of the problems that can occur during import arise from the translation of a trimmed surface from the original file into a form recognized by Abaqus/CAE.

B-rep

More recently, solid modelers have introduced the concept of a “boundary representation,” or “B-rep,” to define a solid object. A B-rep solid is similar to a solid defined by a set of trimmed surfaces; however, a B-rep solid

includes additional information about the faces, edges, and vertices that are generated when the surfaces intersect to form the solid. Abaqus/CAE uses ACIS to store geometric entities, and ACIS uses the concept of B-reps. *Figure 4* illustrates the cylinder defined as a B-rep solid.

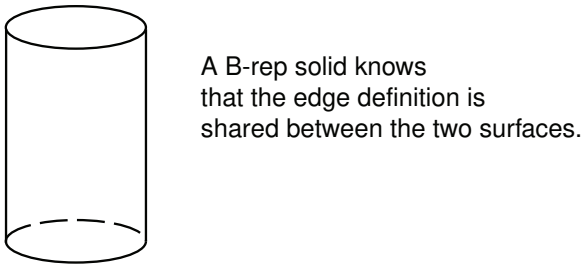


Figure 4: A B-rep solid.

Unlike a solid represented by only trimmed surfaces, a B-rep solid does not duplicate edges that are shared between two surfaces. In a B-rep solid one trimmed surface defines the shared edge and the second edge refers to that definition. For the B-rep solid to be recreated correctly, it must be possible to duplicate the trim curve defined by the first surface along the geometry of the second surface. In some cases ACIS will stitch adjacent edges to create the B-rep solid.

The IGES and STEP standards include the concept of B-reps, although IGES calls them Manifold Solid B-rep Objects or MSBOs. If you import an IGES or STEP file containing two or more trimmed surfaces that are close enough to be stitched together into a B-rep solid, Abaqus/CAE groups the surfaces together as a single solid entity. The VDA-FS standard does not include the concept of B-rep solids; VDA-FS uses only trimmed surfaces to define a solid.

Valid parts, precise parts, and tolerance

A part that you import into Abaqus/CAE must be valid if you wish to analyze the part with Abaqus/Standard or Abaqus/Explicit. The following sections describe valid and precise parts and how Abaqus/CAE uses imprecise modeling and tolerance to construct an imported part.

In this section:

- *What is a valid and precise part?*
- *How are precision and tolerance related?*
- *Working with invalid parts*

What is a valid and precise part?

When you import a solid part, Abaqus/CAE tries to create a closed solid part. Similarly, when you import a shell part, Abaqus/CAE tries to create a connected shell part. If the part is imported successfully, the part is considered valid and precise. However, if the precision of the original part is less than the precision used by Abaqus/CAE, the part may be imprecise or invalid. In most cases you can continue working with an imprecise part. You can also work with an invalid part if you repair it or choose to ignore the invalid status.

The terms “imprecise” and “invalid” are described in more detail below.

Imprecise

A valid part can be either precise or imprecise. If Abaqus/CAE must use a looser tolerance in some areas to recreate a closed volume from the imported part, the part is considered imprecise. You can complete most modeling operations with imprecise parts.

You should try to work with an imprecise part. If Abaqus/CAE cannot proceed, you can suppress the imprecise region or use the geometry edit tools to try and make the part precise. However, if the part contains many complex surfaces, the geometry edit tools may not be able to make the part precise and using the tools may be time consuming. If you cannot work with the imprecise part and you cannot make the part precise, you should return to the CAD application that generated the original file and increase the precision.

Invalid

If the errors are so large that Abaqus/CAE cannot recreate a closed volume from the imported part, the part is considered to be invalid. For example, large gaps between edges cause a part to be invalid. Similarly, points on edges that are far away from an underlying surface cause a part to be invalid.

If the part is invalid, you can use the Geometry Edit toolset to try to make it valid. If you cannot repair a part, you can indicate that you want to ignore the invalid part status and continue to use the part as if it were valid (for more information, see *Working with invalid parts*). However, operations on invalid geometry may fail, give inconsistent results, or cause instabilities in Abaqus/CAE. If you encounter problems with your model after ignoring the invalidity of a part, consider attempting to fix the geometry in the CAD application that generated the original file.

If you do not repair or ignore the status of an invalid part, the only way that you can use it in Abaqus/CAE is to apply a display body or a rigid body constraint to the part in the Interaction module. A display body is included in the model for display purposes only. If you apply a display body constraint, you do not have to mesh the instance and can continue to analyze your model. For more information, see *Display bodies*.

How are precision and tolerance related?

Precision and tolerance are important considerations for successfully importing a part into Abaqus/CAE. For trimmed surfaces the tolerance defines the maximum allowable deviation between an edge and the surface bounded by the edge. The accuracy of the polynomial defining an edge of a trimmed surface depends on the tolerance of the CAD system. Abaqus/CAE uses ACIS to represent a part or the assembly. ACIS uses a precision of 10^{-6} to define a geometric entity.

To successfully import a solid defined by trimmed surfaces or B-reps, the tolerance of the original file and the tolerance of Abaqus/CAE must match within an acceptable limit. If the tolerance of the original file is considerably looser than the tolerance of Abaqus/CAE, the import may fail because Abaqus/CAE cannot reconstruct the solid from the trimmed surfaces and B-rep information.

After you import a part into Abaqus/CAE, the healing process is designed to improve the part's accuracy. Abaqus/CAE tries to change neighboring entities so that their geometry matches exactly. Converting to a precise representation usually results in precise geometry. However, this can be a lengthy operation that increases the complexity of the imported part. As a result, subsequent processing and analysis of the part may be slower. Moreover, if the part contains many complex surfaces, converting to a precise representation is likely to fail. If possible, you should return to the CAD application that generated the original file and increase the precision.

You can use the Query toolset in the Part module to highlight regions of an imported part that have geometry precision and validity errors. An imprecise vertex can be thought of as a vertex surrounded by an imaginary sphere, where the diameter of the sphere is equal to the local precision. When you heal a part, ACIS assumes that any point inside the imaginary sphere is coincident with the vertex, as shown in *Figure 1*.

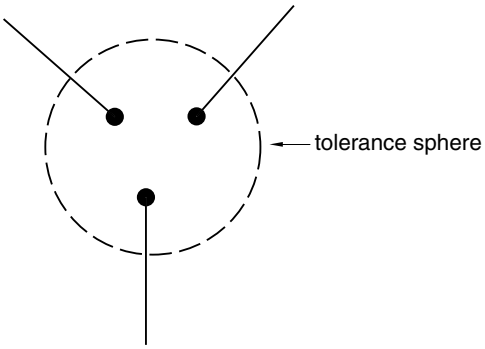


Figure 1: An imaginary sphere defines an imprecise vertex.

Similarly, an imprecise edge can be thought of as an edge surrounded by an imaginary tube, where the diameter of the tube is equal to the local precision. When you heal a part, ACIS assumes that any point inside the imaginary tube is lying on the edge, as shown in *Figure 2*.

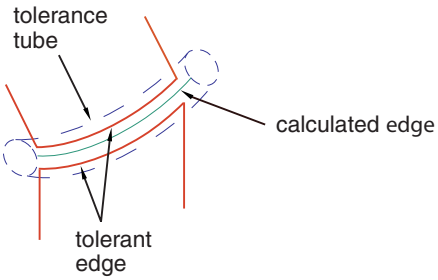


Figure 2: An imaginary tube defines an imprecise edge.

During the healing process ACIS also uses tolerance to determine if a trim curve is positioned on the underlying surface geometry, as shown in *Figure 3*.

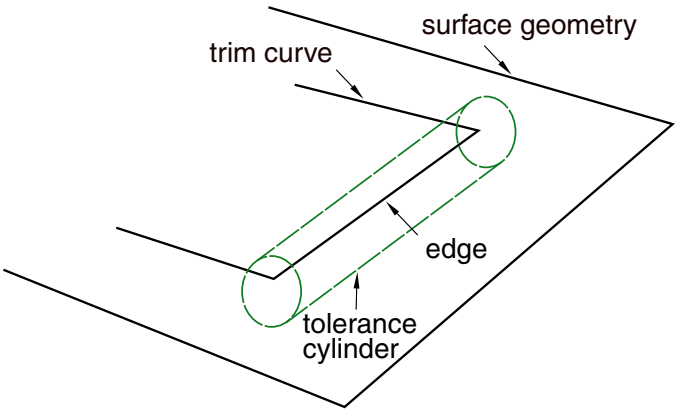


Figure 3: Tolerance determines if a trim curve lies on the underlying surface.

Working with invalid parts

If geometry errors prevent Abaqus/CAE from recreating a closed volume for a solid or shell part, the part is considered to be invalid. In most cases, parts are invalid because they were imported into Abaqus/CAE with errors. Parts may also change validity due to changes in the ACIS interpretation of the geometry between different releases of Abaqus/CAE. If you have invalid parts that are essential to an analysis and cannot be repaired, you can ignore the invalid part status. If a valid part becomes invalid during a database upgrade, you can ignore the invalid geometry without unlocking the part; this allows you to remesh the part if needed without regenerating it.

There are two methods that you can use to ignore the invalid status of a part:

- Select an invalid part in the **Part Manager**, and click **Ignore Invalidity**.
- Select an invalid part or an invalid independent part instance in the Model Tree, click mouse button 3, and select **Ignore Invalidity**.

When you choose to ignore the invalid status of a part or part instance, Abaqus/CAE displays a warning. The warning indicates that while ignoring the invalidity of the part allows you to perform all geometry and meshing functions, ignoring the invalidity may cause some part functions to fail or lead to other undesirable behavior. If you accept this warning, Abaqus/CAE changes the part status to **Invalid (ignored)**. The part status is displayed in the **Part Manager** and in the Model Tree. *Table 1* shows the icons and text used to indicate the status of invalid parts.

Table 1: Part status in the Model Tree and the Part Manager.

Model Tree symbol	Part Manager text	Meaning
	**Invalid**	An unlocked invalid part.
	**Invalid (ignored)**	An unlocked invalid part with the invalid status ignored.
	**Locked, Invalid**	A locked invalid part.
	**Locked, Invalid (ignored)**	A locked invalid part with the invalid status ignored.

Regardless of whether you are trying to repair the geometry or working with the invalid status ignored, Abaqus/CAE does not always recompute the validity of parts as you make changes. Recomputing the validity, especially for complex parts, can take a substantial amount of time. To recompute and update the validity of a part, select **Update Validity** in the **Part Manager** or in the menu that appears when you click mouse button 3 on an invalid part or part instance in the Model Tree. You can also use the **Geometry diagnostics** query to update validity. (For more information on geometry diagnostics, see *Using the geometry diagnostic tools*.)

Controlling the import process

When you import a part from a file generated by a third-party CAD system, Abaqus/CAE allows you to control how it interprets the contents of the file. This section describes the options that are available to you.

In this section:

- *Repairing a part during import*
- *What are the part attributes?*
- *Scaling a part during the import process*

Repairing a part during import

You can import a part and subsequently use the Geometry Edit toolset in the Part module to apply any repair operations that might be required to make the part usable by Abaqus/CAE; see *The Geometry Edit toolset* for more information. Alternatively, you can import a part and repair the part during the import process, as described in this section.

When you import a part, Abaqus/CAE scans the contents of the file and displays a dialog box with a **Name-Repair** tabbed page that allows you to control the following:

Name

The name of the part.

Repair Options

For most of the file formats supported, Abaqus/CAE automatically repairs the part during the import process. However, Abaqus/CAE provides the following additional options when you are importing an IGES- or a VDA-FS-format file:

- **Convert to analytical representation**
- **Stitch gaps**

In most cases, the default settings for these options provide the best results. For more information, see *Editing techniques*.

Part Filter

The following file formats can include several parts in a single file:

- ACIS
- Elysium Neutral File (CATIA V5 or Pro/ENGINEER)
- Parasolid
- STEP

Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value. Alternatively, you can toggle on **Import part number** and enter the number of a single part to import from the file. Abaqus/CAE indicates if any of the parts have validity or precision problems.

In some cases when you import a part into Abaqus/CAE, the geometry of the part contains additional edges and vertices that serve no purpose. The additional geometry splits faces into additional faces and edges into additional edges, resulting in unnecessary complexity. The additional geometry will influence your mesh unduly, and you should use the Geometry Edit toolset to remove the redundant edges and vertices. You can also use virtual topology in the Mesh module to combine small faces and edges and to ignore unnecessary vertices and edges; for more information, see *The Virtual Topology toolset*.

What are the part attributes?

A part has the following attributes:

Modeling Space

When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space of the part being imported as follows:

- If Abaqus/CAE determines the part is three-dimensional, it sets the modeling space to three-dimensional.
- If Abaqus/CAE determines the part is planar, you can choose whether the modeling space is two- or three-dimensional.
- If Abaqus/CAE determines the part is planar and that its geometry does not cross the *Y*-axis, you can choose whether the modeling space is axisymmetric, two-dimensional, or three-dimensional. If you choose axisymmetric, the *Y*-axis is assumed to be the axis of revolution, and you can add a twist degree of freedom.

For more information, see *Part modeling space*.

Type

Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

Although you cannot define an imported part to be an analytical rigid part, you can import the geometry of the analytical rigid part into a sketch. You can then create a new analytical rigid part and copy the imported sketch into the Sketcher toolset.

Scaling a part during the import process

When you import a part, you can choose to maintain the dimensions stored in the file, or you can change the scale of the part during the import process. You can do the following:

- Enter a scale factor that Abaqus/CAE will apply to all of the coordinates in the file. Any offset from the origin will be scaled accordingly.
- If you are importing a part from an ACIS file, you can read the scale factor, the rotation matrix, and the translation matrix from the file.

You can also change the scale of a part while copying it to a new part. For more information, see *Copying a part*.

Understanding the contents of an IGES file

The IGES neutral file format is an international standard that allows you to transfer geometric data between Abaqus/CAE and other CAD applications. You can use IGES-format files to import and export sketches and parts. This section describes the IGES format and the options that are available to you when importing and exporting IGES files.

For a detailed description of how to import and export IGES-format files, see *Importing a part from an IGES-format file*, and *Exporting geometry, model, and mesh data*.

In this section:

- *What are the IGES options in Abaqus/CAE?*
- *What is an IGES entity?*
- *What is in the IGES log file?*
- *Exporting to an IGES file*

What are the IGES options in Abaqus/CAE?

The IGES options allow you to control the following:

Trim Curve Preference

An IGES file can contain curves defined using real space, parameter space, or both. When you import a part from an IGES file into Abaqus/CAE, the surface and the trim curves are converted into an internal representation of the part. By default, Abaqus/CAE uses the information stored in the IGES file to decide how the trim curve is defined; alternatively, you can force Abaqus/CAE to always use either real space or parameter space.

- **As per IGES file**. This is the default option. When this option is selected, Abaqus/CAE uses either the **Always use parametric data** option or the **Always use 3D data** option to decide how the trim curve is defined. Information in the IGES file determines which of the two options is used.
- **Always use parametric data**. This option computes the trim curve parametrically using the surface on which the curve is lying. Each of the data points on the trim curve is located by a surface parameter (u, v). Abaqus/CAE evaluates the surface corresponding to the data point and generates three-dimensional coordinates for the point.

  If the underlying surface has too many sharp deflections that cannot be accurately defined parametrically, the trim curve may not lie on the surface when Abaqus/CAE tries to reconstruct the part. This produces a trimming error and may result in gaps between edges.
- **Always use 3D data**. This option computes the trim curve from the three-dimensional coordinates in space—the part's coordinate system—together with an indication that the trim curve lies on the parametric surface. Each of the data points has its own three-dimensional geometrical point; as a result the trim curve must be re-evaluated each time the surface is moved. The trim curve can move only along the surface. The **Always use 3D data** option should allow trim curves to stay with their underlying surface; however, this is not guaranteed. If you select the **Always use 3D data** option, the import will take longer to complete.

MSBO

A Manifold Solid B-rep Object (MSBO, entity type 186) is an IGES term for a B-rep solid. Like all B-rep solids, the MSBO entity indicates the overall topology of a solid entity by referencing all the trimmed surfaces that define the solid. Abaqus/CAE sets the MSBO option automatically after scanning an IGES-format file and finding the entity.

For an IGES file to contain an MSBO entity, the CAD package that created the file must include the MSBO in the export procedure. CATIA V5 allows for the MSBO entity; SOLIDWORKS does not.

Levels

CAD applications can store entities in an IGES-format file in a sequence of levels (also referred to as layers). For example, different levels can contain geometry, dimensions, text annotations, construction lines, notes, or a legend. When you import a part from an IGES-format file, Abaqus/CAE scans the file and the **Create Part from IGES File** dialog box displays a list of the levels that were found in the IGES file.

If a level contains something other than geometry, such as dimensions, Abaqus/CAE ignores the level during the import. However, if a level does contain geometry, you can delete the level from the list in the dialog box and Abaqus/CAE will not import the level. You must import at least one level. As a result, if the IGES file contains only one level, the **Level** field is not available. Abaqus/CAE ignores any levels that you enter that are not present in the IGES file.

If you encounter problems during the import, you may want to modify the default settings.

What is an IGES entity?

When you import an IGES-format file, Abaqus/CAE scans the contents of the file before displaying the **Create Part from IGES File** dialog box. You can then use buttons on the **IGES Options** tabbed page to view the following:

IGES Header

The IGES header information includes details about the application that wrote the IGES file. It also includes information about the author of the file and the date when the file was written, along with the scale, resolution, and units.

Entity List

An entity can be a geometric entity, such as a point, an arc, or a line. Alternatively, an entity can be separate from the geometry, such as a comment. IGES allocates a number to each entity; for example, a circular arc is entity number 100. The IGES entity list displays a list of each type of entity found in the file. The list includes the IGES entity number, a description of the entity, and the number found in the file.

For a complete list of the IGES entities that can be imported into Abaqus/CAE, see *IGES entities recognized by Abaqus/CAE when importing a part or a sketch*.

What is in the IGES log file?

When you import a part from an IGES file, Abaqus/CAE creates a file called `abaqus_read_iges.log` in the directory from which you started the session. The IGES log file contains information about the entities that were translated along with any problems that were encountered.

Abaqus/CAE overwrites the `abaqus_read_iges.log` file after each IGES import. The following information is written to the IGES log file:

- The global header information.
- A summary of the IGES entities found.
- IGES read options. The options displayed are the options used by the ACIS geometry engine in Abaqus/CAE during the import. You can control only the following options:
  - MSBO
  - Read trim curves
- A conversion log.
- Errors.

Exporting to an IGES file

CAD applications store data in IGES-format files using their own interpretation of the IGES standard. Abaqus/CAE is able to interpret IGES-format files generated by most applications. In addition, when you export a part or assembly to an IGES-format file, Abaqus/CAE allows you to specify the application that will be reading the file, and the data are written out in the appropriate tailored format or flavor. You can choose one of the following flavors:

- Standard
- MSBO
- AutoCAD
- SOLIDWORKS
- JAMA (Japanese Automotive Manufacturer's Association)

By default, Abaqus/CAE exports data to an IGES file using a standard flavor. Abaqus/CAE writes the geometry data to a single layer in the IGES file.

What can you import from a model?

You can import a model from an input file, an output database, or an Abaqus/CAE model database. You can import the complete model from an input file or model database; however, you can import only parts, section definitions, materials, and beam profiles from an output database.
The following sections describe how you can import a model.

For a detailed description of how to import a model, see *Importing a model*.

In this section:

- *Importing a model from an Abaqus/CAE model database*
- *Importing a model from an Abaqus input file*
- *Importing a model from an output database*
- *Importing a model from a Nastran input file*
- *Importing a model from an ANSYS input file*

Importing a model from an Abaqus/CAE model database

You can import a model into the current Abaqus/CAE model database from a different model database by selecting **File->Import->Model** from the main menu bar. Select a model database (`.cae`) file from the dialog box that appears, then select the model you want to import from the **Import Model from Model Database** dialog box. Model databases from previous releases of Abaqus must be upgraded to the current release before you can import their model data.

When you import a model from a model database, Abaqus/CAE creates a copy of that model that includes all model data in the original version. However, this import excludes the following data present in the original model database file but not directly related to the model:

- Job data
- Custom user data created using the Abaqus Scripting Interface

You can also prompt Abaqus/CAE to display the **Copy Objects** dialog box immediately after the model is imported. This dialog box enables you to copy individual objects such as parts, sketches, or materials between models in the current model database. For more information about copying objects, see *Copying objects between models*.

Models imported from different model databases are not linked to their original models; any changes to the original model that occur after the import are not propagated to the copy.

Importing a model from an Abaqus input file

You can use an Abaqus input file to import a model into Abaqus/CAE by selecting **File->Import->Model** from the main menu bar.
Abaqus keywords that are imported from the input file are incorporated into a new model; for example, if the Young's modulus was imported from the **ELASTIC* keyword, it will be available in the Property module. Keywords that are not supported are ignored during import. Model component labels that differ only in case are not imported into Abaqus/CAE consistently; therefore, you should not use case to distinguish between model components of the same type. The input file does not have to be complete; for example, it may not contain any history data. Because an input file is unable to store all of the data from an Abaqus/CAE model database, you should not use input files to archive model data.

The following functionality can be imported into a model from an Abaqus input file:

- Nodes and elements
- Surfaces, node and element sets, and contact node sets
- Adaptive mesh controls
- Material, section, and orientation definitions
- Model and material descriptions
- Interactions and interaction properties
- Discrete fasteners (previously defined in Abaqus/CAE)
- Loads and boundary conditions (in the global coordinate system)
- Amplitudes
- Procedures, output requests, and monitor variables



Note:

Importing models with a very large number of attributes, such as steps or predefined fields, (on the order of 8000 or more) may take a significant amount of time.

See *Keyword support from the input file reader* for a complete list of the keywords that are supported by the input file reader. Not all keyword data lines are supported for import; Abaqus/CAE issues warning messages when unsupported entries are encountered. You can use the Keywords Editor to include options that the input file reader does not support; for detailed instructions on using the Keywords Editor, see *Adding unsupported keywords to your Abaqus/CAE model*.

Importing parts

Parts are imported from an input file in the form of meshes; a mesh consists of node and element definitions along with the type of element assigned. The input file reader can import a mesh containing most of the commonly used element types. However, the input file reader cannot import a mesh containing the following element types:

- Tube support elements (ITS*)
- User-defined elements (U*)

The following element types can be imported, but their section properties are not yet supported:

- Distributing coupling elements (DCOUP*)
- Drag chain elements (DRAG*)
- Frame elements (FRAME*)
- Gap contact stress/displacement elements (GAPCYL, GAPSPHER, and GAPUNI)

- Interface elements (INTER*, ISL*, IRS*, ISP*, ITT*, and DINTER*)
- Joint elements (JOINT*)
- Line spring elements (LS*)

Pore pressure cohesive elements can be imported, but you cannot view them in the Abaqus/CAE model. However, you can view pore pressure cohesive elements in the Visualization module after an analysis is complete.

You can use the Mesh module to change the element type assigned to elements imported from an input file.

Importing sets and surfaces

The import capability creates sets based on any **ELSET* or **NSET* keywords, as well as any ELSET or NSET parameters on other supported keywords. If the input file was written in terms of an assembly of part instances, Abaqus/CAE preserves your intent when creating part and assembly sets. If a set was defined within a part (deformable or rigid), Abaqus/CAE creates a part set. When you instance the part in the Assembly module, you can refer to the part set; however, the assembly-related modules provide only read-only access to part sets. If a set was defined within the assembly, Abaqus/CAE creates an assembly set. For more information, see *How do part sets and assembly sets differ?*.

In contrast, if the input file was not written in terms of an assembly of part instances, Abaqus/CAE tries to minimize the number of sets created. In most cases, sets in the input file are imported as only assembly sets. However, if a section assignment refers to a set, Abaqus/CAE imports the set as only a part set.

Element-based surfaces can be imported; however, Abaqus/CAE imports a node-based surface as a set of nodes, not as a surface. As a result, an imported node-based surface appears in the Set manager and not in the Surface manager.

Importing descriptions

Model descriptions and material descriptions that appear as comment lines in input files are imported into Abaqus/CAE. The comment lines that immediately precede the header of the input file store the model description, which is available in the **Edit Model Attributes** dialog box upon import. The comment lines that immediately precede the material and material behavior definitions store the material descriptions, which are available in the **Edit Material** dialog box upon import.

The job description from the **Edit Job** dialog box appears as the data line for the header of the input file. If desired, you can modify this data line to include more details about the job; however, this data line is not imported into Abaqus/CAE.

Importing interactions, constraints, and fasteners

Abaqus/CAE imports reactivated contact pairs only if the contact pairs were deactivated in the first analysis step.

Abaqus/CAE imports multi-point constraints as MPC connector sections and wire features or as MPC constraints, depending on how the input file identifies the nodes involved.

A fully defined discrete fastener contains both an attachment line and a connector section assignment. In order to recreate these discrete fasteners as originally modeled in Abaqus/CAE upon import, Abaqus/CAE writes special comment lines to the input file when you submit a job for analysis. These special comment lines immediately precede the Abaqus keyword used to define the discrete fasteners, begin with `**@ABQCAE`, and are ignored by the Abaqus solvers.

Assembled fasteners and point-based (mesh-independent) fasteners cannot be imported from an input file.

Importing distributed loads

In certain cases element-based distributed loads defined in the input file using the **DLOAD* keyword are imported as surface-based distributed loads. For identical element-based distributed loads applied over a region, Abaqus/CAE combines the associated regions, creates internal surfaces, and creates surface-based distributed loads applied to the internal surfaces.

Analyzing a submodel

When you analyze a submodel using Abaqus/Standard or Abaqus/Explicit, you provide the name of the output database or results file containing the global solution in the Abaqus execution procedure; the file name does not appear in the input file. As a result, when you import an input file that analyzes a submodel, you must specify the name of the output database or results file containing the global solution that will drive the submodel. Select **Model->Edit Attributes->*Model Name*** and enter the name of the file containing the global solution on the **Submodel** tabbed page.

Importing a model from an output database

You can use an output database to import a model into Abaqus/CAE by selecting **File**->**Import**->**Model** from the main menu bar.

The following functionality can be imported into a model from an output database:

- Nodes and elements
- Surfaces, node and element sets, and contact node sets
- Materials and section definitions
- Beam profiles

Although sections are imported, the assignment of a section to the appropriate element set is not imported. Similarly, the assignment of a beam profile to the appropriate beam section is not imported. When you submit an input file for analysis, Abaqus does not write some material definitions to the output database. As a result, if you import the output database into Abaqus/CAE, these materials will be missing from the model. If this occurs, you can import the model from the input file instead of from the output database.

Abaqus/CAE imports a part from an output database by reading the nodes and elements that define each part instance in the assembly. If the input file that created the output database was structured using parts and assemblies, each part instance imported from the output database appears as a separate part in Abaqus/CAE. In addition, the assembly appears in Abaqus/CAE along with each part instance. As a result, the model imported into Abaqus/CAE contains a part and a part instance corresponding to each part instance in the output database.

The nodes and elements that define a part instance in an output database have been translated and rotated to their position in the assembly. The resulting part that is imported into Abaqus/CAE has the same name as the original part instance in the output database, and its orientation reflects the orientation of the part instance in the output database. As a result, the orientation of the imported part may be different from the orientation of the part in the input file that created the output database.

If the input file that created the output database was not structured using parts and assemblies, Abaqus/CAE writes the mesh definition to the output database as a single part and a single part instance. If the single part in the output database includes an analytical rigid surface, the part that Abaqus/CAE tries to import will contain an invalid combination of deformable parts and analytical rigid parts. As a result the output database cannot be imported.

Importing a model from a Nastran input file

You can use a Nastran input file to import a model into Abaqus/CAE by selecting **File->Import->Model** from the main menu bar. Abaqus/CAE imports many common entities in the Nastran bulk data into the model you create; for more information on supported entities, see *Translating Nastran Bulk Data in Text Files to Partial Abaqus Input Files*.

When you import a Nastran input file into Abaqus/CAE, you can control several aspects of the Nastran-to-Abaqus conversion. These options include:

- Mapping of elements from Nastran to Abaqus element types.
- Selecting a method for converting Nastran section or membrane data into Abaqus sections. You can create Abaqus sections corresponding to each Nastran PSHELL or PCOMP property ID, create sections for all homogeneous elements referencing the same material, or create a separate shell or membrane section for each combination of orientation, offset, and/or thickness in the Nastran input file.
- Using pre-integrated shell sections.
- Applying a scaling multiplier for all density, mass, and rotary inertia values created in the Abaqus/CAE model.
- Converting statements that use the Nastran libraries SOL 101, SOL 108, or SOL 111 into Abaqus perturbation steps with load cases.
- Coupling of beam element offsets by creating new nodes at the offset locations, changing the beam connectivity to the new nodes, and rigidly coupling the new and original nodes.

Importing a model from an ANSYS input file

You can use an ANSYS input file to import a model into Abaqus/CAE by selecting **File->Import->Model** from the main menu bar. Abaqus/CAE imports many common entities in the ANSYS input file into the model you create; for more information on supported entities, see *Translating ANSYS Input Files to Partial Abaqus Input Files*.

A logical approach to successful import of IGES files

The following list describes the steps that you should follow if you experience problems importing a part from an IGES file into Abaqus/CAE. The same approach can be applied to import other file types, but the import options will vary based on the file type.

Defeature the part

Before you export a part from a CAD system, you should know what type of part you want to analyze with Abaqus and what you expect from the analysis. For example, you may want to remove excessive detail that will influence the mesh and dominate the time taken to perform the analysis. For more information, see *Know what you want as an end product*.

Modify the import options in Abaqus/CAE

If you are importing a part from an IGES file, you can choose the following options:

- **Trim Curve Preference**
- **Scale**
- **MSBO**

You can also specify the levels in the IGES file from which to import. For more information on the effects of setting these options, see *What are the IGES options in Abaqus/CAE?*.

Try to repair the part

If Abaqus/CAE requires a precise part to proceed (for example, if you need to partition the part), you can use the Geometry Edit toolset in the Part module to edit the part. For more information, see *Editing techniques*.

Diagnose and locate problems

If you are not sure why a part is still unusable by Abaqus/CAE, you can try the following:

- Look at the **IGES Entity Filter** list in the **IGES Options** dialog box for unsupported entities.
- Grey or lighter unconnected lines on a face are silhouette lines indicating undulations in the surface that Abaqus/CAE introduced while attempting to make the surface more precise. You should manually delete the face or import the part again without using the **Convert to precise representation** automated repair option.
- Try to mesh the part. Faces that cannot be meshed by Abaqus/CAE are an indication of poor geometry.

Try to reimport the part

If the repair tools fail to produce a valid part, the file describing the part may be invalid or contain illegal statements. You should try to import the part back into the CAD system that created it. If the part cannot be imported, you should check the export settings and regenerate the file. Entities in the file that are not related to the part definition will cause the import to fail; for example, a border around the part or a title block. Ideally, you should not try to import a part into Abaqus/CAE until you know that it can be imported back into the CAD system that generated the part.

Importing sketches and parts

This section describes how you use the main menu bar's **File** menu to import sketches and parts.

In this section:

- *Importing sketches*
- *Importing parts*
- *Importing an assembly*
- *Importing parts from an ACIS-format file*
- *Importing a part from a V5-format file*
- *Importing a part from an Elysium Neutral file*
- *Importing a part from an IGES-format file*
- *IGES entities recognized by Abaqus/CAE when importing a part or a sketch*
- *Importing a part from a Parasolid-format file*
- *Importing a part from a STEP-format file*
- *Importing a part from a VDA-FS-format file*
- *Importing a part from a SOLIDWORKS-format file*
- *Importing a part from an output database*
- *Importing a substructure into a model database as a part*
- *Importing a part from an Abaqus input file*
- *Importing an assembly from an Elysium assembly file*
- *Importing an assembly from an Elysium Neutral file*
- *Importing an assembly from a Parasolid-format file*

Importing sketches

Select **File->Import->Sketch** from the main menu bar to import a sketch from either:

- An IGES-format file (`.igs` files)
- An AutoCAD-format file (`.dxf` files)
- An ACIS-format file (`.sat` files)
- A STEP-format file (`.stp` files)

If you are importing a sketch, the file must contain a two-dimensional planar profile that can be mapped directly to the sketch plane. If the file contains three-dimensional geometry, Abaqus/CAE does the following:

- If you are importing an AutoCAD file, Abaqus/CAE creates the sketch using the geometry in the *X*–*Y* plane only.
- If you are importing an IGES, ACIS, or STEP file, Abaqus/CAE displays an error message and cancels the import procedure.

You can import sketches from files that contain only simple geometry because Abaqus/CAE must be able to translate the geometry to a corresponding Sketcher entity, such as a line, circle, arc, or spline. If Abaqus/CAE finds geometry it cannot translate, it ignores that geometry. For a list of the IGES and AutoCAD entities supported by Abaqus/CAE, see *Imported sketches*.

1. From the main menu bar, select **File->Import->Sketch**.

   Abaqus/CAE displays the **Import Sketch** dialog box.

2. From the **File Filter** menu at the bottom of the **Import Sketch** dialog box, select one of the following:

   - **ACIS SAT (*.sat*)**
   - **IGES (*.igs*)**
   - **STEP (*.stp*)**
   - **AutoCAD DXF (*.dxf*)**

3. Select the file containing the sketch to import, and click **OK**.

   Abaqus/CAE converts the profile in the file to a sketch. The sketch now appears in the list of sketches in the model and can be retrieved when you enter the Sketch module. For information on how to use the imported sketch, see *Stand-alone sketches*.

Additional information

- *Understanding the contents of an IGES file*
- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *The Sketch module*
- *Using the File menu*

Importing parts

You can import a part from a file that uses an external format, from an output database, from a substructure, or from an Abaqus input file.

Select **File->Import->Part** from the main menu to import a part from either:

An external-format file

You can import geometry information from the following formats:

- ACIS
- Elysium Neutral file generated by Pro/ENGINEER, NX, CATIA V5, or CATIA V6
- CATIA V5
- IGES
- Parasolid
- STEP
- VDA-FS
- SOLIDWORKS

For information about the versions of the CAD software supported by Abaqus/CAE, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

You can import all of the parts in an assembly, or you can import only a specified part. Abaqus/CAE creates a separate part for each part that is imported. As a result, if a part is instanced many times in the assembly, Abaqus/CAE creates separate parts corresponding to each instance. Abaqus/CAE does not retain the names of the parts or any color coding.

An output database

You can import the part instances stored in an output database in the form of mesh parts. A mesh part contains no feature information and is extracted from the output database as a collection of nodes, elements, surfaces, and sets. You can import either the undeformed or the deformed shape. If you import the deformed shape, you can specify the step and the increment from which to import. For more information, see *Importing a part from an output database*.

A substructure

You can import a substructure into a model database as a new part definition. For more information, see *Importing a substructure into a model database as a part*.

An Abaqus input file

You can import parts defined in an Abaqus input file in the form of orphan mesh parts. For more information, see *Importing a part from an Abaqus input file*.

1. From the main menu, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select one of the following:

   - **ACIS SAT (*.sat*)**; for more information, see *Importing parts from an ACIS-format file*.
   - **IGES (*.igs*, *.iges*)**; for more information, see *Importing a part from an IGES-format file*.

- **VDA (*.vda*)**; for more information, see *Importing a part from a VDA-FS-format file*.
- **STEP (*.stp*, *.step*)**; for more information, see *Importing a part from a STEP-format file*.
- **CATIA V5 (*.CATPart*, *.CATProduct*)**; for more information, see *Importing a part from a V5-format file*.
- **Parasolid (*.x_t*,*.x_b*, *.xmt*)**; for more information, see *Importing a part from a Parasolid-format file*.
- **ProE/NX/CATIA V5 Elysium Neutral (*.enf*)**; for more information, see *Importing a part from an Elysium Neutral file*.
- **Output Database (*.odb)**; for more information, see *Importing a part from an output database*.
- **Substructure (*.sim*)**; for more information, see *Importing a substructure into a model database as a part*.
- **Abaqus Input File (*.inp)**; for more information, see *Importing a part from an Abaqus input file*.
- **SOLIDWORKS (*.sldprt*, *.sldasm*)**; for more information, see *Importing a part from a SOLIDWORKS-format file*.

3. Abaqus/CAE lists all the files in the selected directory with the appropriate file extension. Select the file containing the part to import, and click **OK**.

   Abaqus/CAE enters the Part module and either displays a **Create** dialog box pertaining to your selection or creates the parts.

4. If applicable, select the desired repair options and part attributes and click **OK**. For more information, see *Controlling the import process* and *What are the part attributes?*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing an assembly

Select **File->Import->Assembly** from the main menu bar to import an assembly from the following formats:

- Assembly file
- Elysium Neutral file generated by Pro/ENGINEER, CATIA V5, or CATIA V6
- Parasolid

For information about the versions of the CAD software supported by Abaqus/CAE, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*. Abaqus/CAE does not support subassemblies. As a result, assemblies and subassemblies in the imported file appear as a single assembly in Abaqus/CAE.

You can import the entire assembly, or you can import only selected part instances. Abaqus/CAE appends your selection to the existing assembly and retains the original positioning of the instances.

1. From the main menu bar, select **File->Import->Assembly**.

   The **Import Assembly** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Assembly** dialog box, select one of the following:

   - **Assembly File (*.eaf*)**; for more information, see *Importing an assembly from an Elysium assembly file*.
   - **Parasolid (*.x_t*,*.x_b*, *.xmt*)**; for more information, see *Importing an assembly from a Parasolid-format file*.
   - **ProE/NX/CATIA V5 Elysium Neutral (*.enf*)**; for more information, see *Importing an assembly from an Elysium Neutral file*.

3. Abaqus/CAE lists all the files in the selected directory with the appropriate file extension. Select the file containing the assembly to import, and click **OK**.

   Abaqus/CAE enters the Assembly module and displays the **Import Assembly from *format* File** dialog box.

4. Do either of the following:

   - Click on the assembly name to import all of the part instances in the assembly.
   - Click on the arrow to the left of the assembly name to display all of the part instances in the assembly. Click on the names of the part instances that you want to import.

5. Click **OK** to import the assembly or selected part instances and to exit the **Import Assembly from *format* File** dialog box.

   Abaqus/CAE creates the part instances and the corresponding parts.

   Abaqus/CAE displays a message in the message area indicating if any of the part instances are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing parts from an ACIS-format file

Many computer-aided modeling and drafting applications can read and write ACIS-format files; you can use these files to exchange geometry information between these applications and Abaqus/CAE. Select **File->Import->Part** from the main menu bar to import a part from an ACIS-format file. You can import multiple parts stored in an ACIS-format file. You can export parts and the assembly from Abaqus/CAE into an ACIS-format file.

You cannot import parts of mixed modeling space from an ACIS-format file; for example, solids and axisymmetric surfaces. In addition, you cannot import parts of mixed type; for example, deformable bodies and discrete rigid surfaces.

An imported ACIS part forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **ACIS SAT (*.sat*)**.

   Abaqus/CAE lists all the files in the selected directory with a `.sat*` file extension.

3. Select the ACIS file containing the part or parts to import, and click **OK**.

   The **Create Part from ACIS File** dialog box appears.

4. Click the **Name-Repair** tab to set the following:

Name

The name of the new part. For more information, see *Using basic dialog box components*.

Part Filter

An ACIS file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the ACIS file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part.

In addition, if you combine these parts, Abaqus/CAE enables you to control the following aspects of the geometry combination:

- You can retain the intersecting boundaries between the imported parts, which enables you to eliminate invalid geometry.
- You can stitch these imported parts together using a user-specified stitching tolerance value.

If you want to import an individual part from the ACIS file, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   Again, in most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

- Choose **No scale** to maintain the dimensions stored in the file.
- Choose **Use transform from file, including scale** to read the scale factor, the rotation matrix, and the translation matrix from the ACIS file.
- Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the ACIS part.

   Abaqus/CAE enters the Part module, the imported part replaces the contents of the current viewport, and the part appears in the model's list of parts in the context bar.

   When you import a single part, Abaqus/CAE displays a message in the message area indicating if the part contains any validity or precision problems. If you import multiple parts from a single ACIS file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from a V5-format file

Select **File->Import->Part** from the main menu bar to import a part from a V5-format file directly into Abaqus/CAE. You can import CATIA-format parts, but you cannot export a CATIA-format part. For information about the versions of CATIA V5 supported by Abaqus/CAE, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.



Note:

You can directly import CATIA V5-format files only if you have a license for the optional add-on CATIA V5 Associative Interface. For more information, see *What can I do with the associative interfaces?*.

An imported CATIA part forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **CATIA V5 (*.CATPart*, *.CATProduct*)**.

   Abaqus/CAE lists all the files in the selected directory with acceptable file extensions.

3. Select the CATIA file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from CATIA File** dialog box.

4. Click the **Name-Repair** tab to set the following:

Name

The name of the new part. For more information, see *Using basic dialog box components*.

Repair Options

Abaqus/CAE automatically repairs a CATIA part during the import process.

Part Filter

A CATIA file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the CATIA file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value.

If you want to import an individual part from the CATIA file, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

Again, in most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the part and to exit the **Create Part from CATIA File** dialog box.

   Abaqus/CAE scans the CATIA file and starts the repair process, depending on the options selected. If you want to cancel the import process, click **Stop** in the prompt area.

8. When you import a single part, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If you import multiple parts from a single CATIA file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from an Elysium Neutral file

Select **File->Import->Part** from the main menu bar to import a part from an Elysium Neutral file. You can import an Elysium Neutral file that was generated by the following:

- NX using the Abaqus/CAE Associative Interface for NX.
- Pro/ENGINEER using the Pro/ENGINEER Associative Interface.

For information about the versions of NX and Pro/ENGINEER supported by Abaqus/CAE, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

During the part import process, Abaqus/CAE automatically determines whether the selected file was generated by NX or Pro/ENGINEER and tailors the import process accordingly. An imported part from an Elysium Neutral file forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **ProE/NX Elysium Neutral (*.enf*)**.

   Abaqus/CAE lists all the files in the current directory with a `.enf*` file extension.

3. Select the Elysium Neutral file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from ENF File** dialog box.

4. Click the **Name-Repair** tab to set the following:

Name

The name of the new part. For more information, see *Using basic dialog box components*.

Repair Options

Abaqus/CAE automatically repairs an Elysium Neutral file part during the import process.

Part Filter

An Elysium Neutral file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the Elysium Neutral file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value.

Alternatively, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

In most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.
   - Choose **Do not change units** to maintain the units and dimensions stored in the file.
   - Choose **Convert units to mm** to convert the units stored in the Elysium Neutral file to millimeters.
   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.
7. Click **OK** to import the part and to exit the **Create Part from ENF File** dialog box.

   Abaqus/CAE scans the Elysium Neutral file and starts the repair process, depending on the options selected. If you wish to cancel the import process, click **Stop** in the prompt area.
8. When you import a single part, Abaqus/CAE displays a message in the message area indicating if the part contains any validity or precision problems. If you import multiple parts from a single Elysium Neutral file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from an IGES-format file

Many computer-aided modeling and drafting applications can read and write IGES-format files; you can use these files to exchange geometry information between these applications and Abaqus/CAE. Select **File->Import->Part** from the main menu bar to import a part from an IGES-format file. You can export sketches and parts from Abaqus/CAE into an IGES-format file, but you cannot export the assembly to an IGES-format file.

If the IGES-format file contains multiple parts, Abaqus/CAE imports them as a single part. An imported IGES part forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **IGES (*.igs*. *.iges*)**.

   Abaqus/CAE lists all the files in the current directory with a `.igs*` or `.iges*` file extension.

3. Select the IGES file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from IGES File** dialog box.

4. Click the **Name-Repair** tab to set the following:

   - The name of the new part.
   - The repair options that Abaqus/CAE will apply during the import.

   In most cases you should accept the default settings; for more information, see *Controlling the import process*.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   Again, in most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

   - Choose **No scale** to maintain the dimensions stored in the file.
   - Choose **Use transform from file, including scale** to read the scale factor, the rotation matrix, and the translation matrix from the IGES file.
   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click the **IGES Options** tab to set the following:

   - How Abaqus/CAE converts surface and trim curves into an internal representation of the part.
   - Whether Abaqus/CAE recognizes a Manifold Solid B-rep Object (MSBO, entity type 186). In most cases Abaqus/CAE sets the MSBO option automatically after scanning an IGES-format file and finding the entity. However, if the IGES file is very large, you must set this option manually.

- The levels in the IGES file that Abaqus/CAE will try to import. For example, if the **Levels** field contains `0:2,4`, Abaqus/CAE will try to import levels 0, 1, 2, and 4. If a level contains something other than geometry, Abaqus/CAE will ignore the level during the import. You can remove levels from the list, but you cannot add levels. This field is not available if the IGES file contains only one level.

For more information, see *What are the IGES options in Abaqus/CAE?*, and *What is an IGES entity?*. You can also use the **IGES Options** dialog box to view the header information in the IGES file and to view a list of the entities found in the file. For more information, see *What is an IGES entity?*.

8. Click **OK** to import the IGES part and to close the **Create Part from IGES File** dialog box.

   Abaqus/CAE scans the IGES file and starts the repair process, depending on the options selected. If you wish to cancel the import process, click **Stop** in the prompt area.

9. When the part is imported, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *What are the IGES options in Abaqus/CAE?*
- *Editing techniques*
- *Using the File menu*

IGES entities recognized by Abaqus/CAE when importing a part or a sketch

During the import process, Abaqus/CAE converts the entities stored in the IGES file to an internal representation recognized by Abaqus/CAE. The IGES file can contain entities that are not recognized by Abaqus/CAE; however, these entities are ignored during the conversion.

Table 1 lists the IGES entities that Abaqus/CAE recognizes.

Table 1: IGES entities recognized by Abaqus/CAE when importing a part or a sketch.

Entity	Form	IGES Entity Name
100	0	Circular arc
104	1	Conic arc: general
104	2	Conic arc: ellipse
104	3	Conic arc: parabola
106	11	Copious data: 2D path
106	12	Copious data: 3D path
106	63	Copious data: Closed 2D curve
108	1	Plane entity: bounded
110	0	Line
112	0	Parametric spline curve
114	0	Parametric spline surface
116	0	Point
118	1	Ruled surface
120	0	Surface of revolution
122	0	Tabulated cylinder
123	0	Direction
124	0	Transformation
126	0	Rational B-spline curve
128	0	Rational B-spline surface
130	0	Offset curve
140	0	Offset surface
141	0	Boundary entity
142	0	Curve on parametric surface
143	0	Bounded surface
144	0	Trimmed surface
186	0	MSBO
190	0	Plane surface
192	0	Right circular cylindrical surface
194	0	Right circular conical surface
196	0	Spherical surface

Entity	Form	IGES Entity Name
198	0	Toroidal surface
502	1	Vertex list
504	1	Edge list
508	1	Loop
510	1	Face
514	1	Shell

You cannot import composite curves (IGES entity ID 102) into Abaqus/CAE.

Additional information

- *Importing a part from an IGES-format file*
- *Understanding the contents of an IGES file*
- *Editing techniques*

Importing a part from a Parasolid-format file

Select **File->Import->Part** from the main menu bar to import a part from a Parasolid-format file. Parasolid is a library of solid modeling functions developed by Unigraphics Solutions and marketed by EDS. You can use a translator from Elysium, Inc., to import Parasolid-format parts directly into Abaqus/CAE. A variety of CAD products can generate Parasolid-format parts, such as Unigraphics, SOLIDWORKS, Solid Edge, FEMAP, and MSC.Patran. For information about the versions of the CAD software supported by the translators from Elysium, Inc., see the *https://www.3ds.com/products-services/simulia/products/abaqus/add-ons/geometry-translators/* page at *www.3ds.com/simulia*.

An imported Parasolid part forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **Parasolid (*.x_t*,*.x_b*, *.xmt*)**.

   Abaqus/CAE lists all the files in the current directory with a `.x_t*`, `.x_b*`, or `.xmt*` file extension.

3. Select the Parasolid file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from PARASOLID File** dialog box.

4. Click the **Name-Repair** tab to set the following:

Name

The name of the new part. For more information, see *Using basic dialog box components*.

Repair Options

Abaqus/CAE automatically repairs a Parasolid part during the import process.

Part Filter

A Parasolid file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the Parasolid file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value.

If you want to import an individual part from the Parasolid file, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   In most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

- Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the Parasolid part and to close the **Create Part from PARASOLID File** dialog box.

   Abaqus/CAE scans the Parasolid file and repairs the part during the import process. If you wish to cancel the import process, click **Stop** in the prompt area.

8. When you import a single part, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If you import multiple parts from a single Parasolid file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from a STEP-format file

Select **File->Import->Part** from the main menu bar to import a part from a STEP-format file. You can export sketches and parts from Abaqus/CAE into a STEP-format file, but you cannot export the assembly to a STEP-format file.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select STEP **(*.stp, *.step)**.

   Abaqus/CAE lists all the files in the selected directory with a `.stp` or `.step` file extension.

3. Select the STEP file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from STEP File** dialog box.

4. Click the **Name-Repair** tab to set the following:

Name

When you import a part or parts into Abaqus/CAE, you can use the part names specified in the STEP-format file or you can provide a name for the imported part or parts. If you import multiple parts and provide a part name, Abaqus/CAE uses your name and appends a number to it; for example, if you specify the part name `ImportedPart` and create three individual parts, the newly imported Abaqus/CAE parts will be named `ImportedPart-1`, `ImportedPart-2`, and `ImportedPart-3`.

Part Filter

A STEP file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the STEP file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value.

If you want to import an individual part from the STEP file, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   Again, in most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

   - Choose **No scale** to maintain the dimensions stored in the file.
   - Choose **Use transform from file, including scale** to read the scale factor, the rotation matrix, and the translation matrix from the STEP file.
   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the STEP part and to close the **Create Part from STEP File** dialog box.

   Abaqus/CAE scans the STEP file and starts the repair process, depending on the options selected. If you wish to cancel the import process, click **Stop** in the prompt area.

   When you import a single part, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If you import multiple parts from a single STEP file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from a VDA-FS-format file

Many computer-aided modeling and drafting applications can read and write VDA-FS-format files; you can use these files to exchange geometry information between these applications and Abaqus/CAE. Select **File->Import->Part** from the main menu bar to import a part from a VDA-FS-format file. You can export parts from Abaqus/CAE into a VDA-FS-format file, but you cannot export the assembly to a VDA-FS-format file.

If the VDA-FS-format file contains multiple parts, Abaqus/CAE imports them as a single part. An imported VDA-FS part forms the base feature of a new part in Abaqus/CAE; you cannot modify this base feature directly, but you can add additional features to it, such as a solid extrusion or a blind cut.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **VDA-FS (*.vda*)**.

   Abaqus/CAE lists all the files in the selected directory with a `.vda*` file extension.

3. Select the VDA-FS file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from VDA File** dialog box.

4. Click the **Name-Repair** tab to set the following:

   - The name of the new part.
   - The repair options that Abaqus/CAE will apply during the import.

   For more information, see *Controlling the import process*.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   In most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

   - Choose **No scale** to maintain the dimensions stored in the file.
   - Choose **Use transform from file, including scale** to read the scale factor, the rotation matrix, and the translation matrix from the VDA-FS file.
   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the VDA-FS part and to close the **Create Part from VDA File** dialog box.

   Abaqus/CAE scans the VDA-FS file and starts the repair process, depending on the options selected. If you wish to cancel the import process, click **Stop** in the prompt area.

When the part is imported, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from a SOLIDWORKS-format file

Select **File->Import->Part** from the main menu to import a part from a SOLIDWORKS-format file.

1. From the main menu, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **SOLIDWORKS (*.sldprt*, *.sldasm*)**.

   Abaqus/CAE lists all the files in the selected directory with a `.sldprt` or `.sldasm` file extension.

3. Select the SOLIDWORKS file containing the part to import, and click **OK**.

   Abaqus/CAE displays the **Create Part from SOLIDWORKS File** dialog box.

4. Click the **Name-Repair** tab to set the following:

Name

When you import a part or parts into Abaqus/CAE, you can use the part names specified in the SOLIDWORKS-format file or you can provide a name for the imported part or parts. If you import multiple parts and provide a part name, Abaqus/CAE uses your name and appends a number to it; for example, if you specify the part name `ImportedPart` and create three individual parts, the newly imported Abaqus/CAE parts will be named `ImportedPart-1`, `ImportedPart-2`, and `ImportedPart-3`.

Part Filter

A SOLIDWORKS file can contain more than one part, and Abaqus/CAE imports all of the parts in the file by default. If you import all the parts in the SOLIDWORKS file, you can create individual Abaqus parts for each of them or you can combine them together into a single Abaqus part; in addition, if you combine these parts, Abaqus/CAE enables you to stitch these imported parts together using a user-specified stitching tolerance value.

If you want to import an individual part from the SOLIDWORKS file, you can toggle on **Import part number** and enter the number of a single part to import from the file.

5. Click the **Part Attributes** tab to set the following:

   - The modeling space of the part being imported. When you import a part, Abaqus/CAE scans the file and tries to determine the modeling space.
   - The type of the part being imported. Abaqus/CAE always assumes that the part type is deformable. Alternatively, you can select **Discrete rigid** to import a discrete rigid part or select **Eulerian** to import an Eulerian part. For more information, see *Part types*.

   Again, in most cases you should accept the default settings; for more information, see *What are the part attributes?*.

6. Click the **Scale** tab to scale the part during the import process.

   - Choose **No scale** to maintain the dimensions stored in the file.
   - Choose **Use transform from file, including scale** to read the scale factor, the rotation matrix, and the translation matrix from the SOLIDWORKS file.
   - Choose **Multiply all lengths by**, and enter a scale factor. Abaqus/CAE applies the scale factor to all of the coordinates in the file. As a consequence, any offset from the origin will be scaled accordingly.

7. Click **OK** to import the SOLIDWORKS part and to close the **Create Part from SOLIDWORKS File** dialog box.

   Abaqus/CAE scans the SOLIDWORKS file and starts the repair process, depending on the options selected. If you wish to cancel the import process, click **Stop** in the prompt area.

   When you import a single part, Abaqus/CAE displays a message indicating if the part contains any validity or precision problems. If you import multiple parts from a single SOLIDWORKS file, you can use the **Part Manager** to check if any of the parts are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a part from an output database

Select **File->Import->Part** from the main menu bar to import part instances stored in an output database in the form of mesh parts. If the output database contains multiple part instances, you can select the part instances to import. Abaqus/CAE imports each part instance as a separate part. You can import either the undeformed or the deformed shape. If you import the deformed shape, you can specify the step and the frame from which to import.

To verify the quality of the mesh, you can display the part in the Mesh module and select **Mesh->Verify** from the main menu bar. In addition, you can use the Mesh module to change the element type assigned to the mesh and to edit the original mesh definition. For more information, see *What can I do with the Edit Mesh toolset?*, and *Assigning Abaqus element types*.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **Output Database (*.odb)**.

   Abaqus/CAE lists all the files in the selected directory with an `.odb` file extension.

3. Select the output database containing the part to import, and click **OK**.

   The **Create Part from Output Database** dialog box appears. The dialog box lists each part instance in the output database along with its type (deformable body or discrete rigid surface).

4. From the dialog box, select the instances to import.
5. If you selected only a single part instance, Abaqus/CAE uses the name of the instance to name the resulting part, although you can change the name if desired. In contrast, if you selected more than one part instance to import, Abaqus/CAE uses the name of each instance to name each part and you cannot change their names.

   Abaqus/CAE determines the modeling space (three-dimensional, two-dimensional, or axisymmetric) of the part instances. You cannot change the modeling space or the type.

6. By default, Abaqus/CAE imports the undeformed configuration of the parts. To import the deformed parts, click **Import deformed configuration**. Select the step and frame containing the deformed shape from the available steps and frames in the output database.

   When importing deformed parts, the deformations are read from the field output variable U, if available; otherwise, the deformations are read from the field output variable UT.

7. Click **OK** to import the orphan mesh from the output database and to close the dialog box.
8. If the name that you entered is the same as the name of an existing part in the model, Abaqus/CAE asks if you want to overwrite the existing part or replace the mesh.

   If you choose to replace the mesh, Abaqus/CAE replaces the nodes and elements of the existing part with the nodes and elements of the imported orphan mesh. Sets and section assignments that referred to the original part are maintained. However, because the sets and section assignments refer to node and element numbers, the mesh of the imported part should be similar to the mesh of the original part. For example, you could replace the undeformed mesh with the deformed mesh.

   Abaqus/CAE enters the Part module, the imported part replaces the contents of the current viewport, and the part appears in the model's list of parts in the context bar.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*
- *Editing techniques*

Importing a substructure into a model database as a part

Select **File->Import->Part** from the main menu bar to import a substructure definition in a `.sim` file in the form of a new part definition. The `.sim` file you import must reside in the same directory as the supporting Abaqus files to which the SIM database refers; these supporting files might include data in the formats `.prt`, `.mdl`, or `.stt`. In addition, the release of the `.sim` file you import must be consistent with the release of Abaqus/CAE that you are using.

Substructure import also requires an output database (`.odb`) file for mesh display.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.
2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **Substructure (*.sim*)**.

   Abaqus/CAE lists all the files in the selected directory with a `.sim*` file extension.
3. Select the SIM database containing the substructure to import, and click **OK**.

   The **Create Substructure Part** dialog box appears, displaying the substructure's name and its unique identifier (which begins with the letter Z).
4. If desired, customize the **Part name** for the substructure you are importing.
5. In the **Use mesh from ODB file** field, specify the name of the output database file containing the mesh that you want to associate with this substructure definition.
6. Click **OK**.

   Abaqus/CAE imports the substructure into your model as a new part definition.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*
- *Editing techniques*

Importing a part from an Abaqus input file

You can import parts defined in an Abaqus input file in the form of orphan mesh parts.
Select **File->Import->Part** from the main menu bar to import parts defined in an Abaqus input file.

If the input file contains multiple parts, Abaqus/CAE imports each part as a separate orphan mesh part. Only the orphan mesh part information is imported; properties that are assigned to the parts, such as sections and materials, are not imported.

1. From the main menu bar, select **File->Import->Part**.

   The **Import Part** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Part** dialog box, select **Abaqus Input File (*.inp)**.

   Abaqus/CAE lists all the files in the selected directory with an `.inp` file extension.

3. Select the input file containing the parts to import, and click **OK**.

   Abaqus/CAE enters the Part module and imports the parts. The parts appear in the Model Tree and in the model's list of parts in the context bar.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*

Importing an assembly from an Elysium assembly file

Select **File->Import->Assembly** from the main menu bar to import an assembly from an Elysium assembly file. Elysium assembly files are created by all of the associative interfaces for Abaqus/CAE (see *What can I do with the associative interfaces?*).

The geometry files for the individual parts in the assembly must be located in the same directory as the assembly file. The Elysium assembly file always has the file extension `.eaf`; however, the file type of the part geometry files depends on which associative interface was used to write the model:

CATIA V6 Associative Interface

CATIA V5 part files (`.CATPart`). (The CATIA V6 parts and assemblies are converted to CATIA V5 format before they are exported.)

CATIA V5 Associative Interface

CATIA V5 part files (`.CATPart`)

SOLIDWORKS Associative Interface

ACIS files (`.sat`)

Pro/ENGINEER Associative Interface

Pro/ENGINEER Elysium Neutral files (`.enf_abq`)

Abaqus/CAE Associative Interface for NX

NX Elysium Neutral files (`.enf_abq`)

The following procedure is necessary only when performing a manual associative import of a model assembly. For complete details about creating assembly files and various methods for importing models using the associative interfaces, refer to the user guides for each associative interface. The user guides are available from the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

1. From the main menu bar, select **File->Import->Assembly**.

   The **Import Assembly** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Assembly** dialog box, select **Assembly File (*.eaf*)**.

   Abaqus/CAE lists the files in the selected directory with a `.eaf` file extension.

3. Select the file containing the assembly to import, and click **OK**.

   Abaqus/CAE displays the **Import Assembly from EAF File** dialog box.

4. Do either of the following:

   - Click on the assembly name to import all of the part instances in the assembly.
   - Click on the arrow to the left of the assembly name to display all of the part instances in the assembly. Click on the names of the part instances that you want to import.

5. Click **OK** to import the assembly or selected part instances and to exit the **Import Assembly from EAF File** dialog box.

Abaqus/CAE displays a message indicating if any of the part files were not found or if any part instances are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Importing an assembly from an Elysium Neutral file

Select **File->Import->Assembly** from the main menu bar to import an assembly from an Elysium Neutral file. You can import an Elysium Neutral file that was generated by the following:

- CATIA V6 using the CATIA V6 Associative Interface.
- NX using the Pro/ENGINEER Associative Interface.
- Pro/ENGINEER using the Pro/ENGINEER Associative Interface.

For details about the versions of CATIA V6, NX, and Pro/ENGINEER supported by Abaqus/CAE, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

1. From the main menu bar, select **File->Import->Assembly**.

   The **Import Assembly** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Assembly** dialog box, select **ProE/NX Elysium Neutral (*.enf*)**.

   Abaqus/CAE lists all the files in the current directory with the appropriate file extension.

3. Select the file containing the assembly to import, and click **OK**.

   Abaqus/CAE displays the **Import Assembly from ENF File** dialog box.

4. Do either of the following:

   - Click on the assembly name to import all of the part instances in the assembly.
   - Click on the arrow to the left of the assembly name to display all of the part instances in the assembly. Click on the names of the part instances that you want to import.

5. Click **OK** to import the assembly or selected part instances and to exit the **Import Assembly from ENF File** dialog box.

   Abaqus/CAE displays a message indicating if any of the part instances are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing an assembly from a Parasolid-format file

Select **File->Import->Assembly** from the main menu bar to import an assembly from a Parasolid-format file. Parasolid is a library of solid modeling functions developed by Unigraphics Solutions and marketed by EDS. You can use a translator from Elysium, Inc., to import Parasolid-format assemblies directly into Abaqus/CAE. A variety of CAD products can generate Parasolid-format parts, such as Unigraphics, SOLIDWORKS, Solid Edge, FEMAP, and MSC.Patran. For information about the versions of the CAD software supported by the translators from Elysium, Inc., see the *https://www.3ds.com/products-services/simulia/products/abaqus/add-ons/geometry-translators/* page at *www.3ds.com/simulia*.

1. From the main menu bar, select **File->Import->Assembly**.

   The **Import Assembly** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Import Assembly** dialog box, select **Parasolid (*.x_t*,*.x_b*, *.xmt*)**.

   Abaqus/CAE lists all the files in the current directory with a `.x_t*`, `x_b*`, and `.xmt*` file extension.

3. Select the Parasolid file containing the assembly to import, and click **OK**.

   Abaqus/CAE displays the **Import Assembly from PARASOLID File** dialog box.

4. Do either of the following:

   - Click on the assembly name to import all of the part instances in the assembly.
   - Click on the arrow to the left of the assembly name to display all of the part instances in the assembly. Click on the names of the part instances that you want to import.

5. Click **OK** to import the assembly or selected part instances and to exit the **Import Assembly from PARASOLID File** dialog box.

   Abaqus/CAE displays a message indicating if any of the part instances are invalid or imprecise. If necessary, use the Query toolset to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

Additional information

- *Editing techniques*
- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*

Importing a model

Select **File->Import->Model** from the main menu bar to import a model. You can import a model from the following:

Abaqus/CAE model database (*.cae)

Abaqus/CAE creates a copy of the model that you select in the current model database. Data other than model data, such as analysis jobs or custom user data that were created using the Abaqus Scripting Interface, are not copied to the current model database.

Abaqus input file (*.inp, *.pes)

Options and parameters in the input file are translated into objects recognized by the import capability, and Abaqus/CAE creates a new model. For a detailed list of the keywords supported by the import model capability, see *Keyword support from the input file reader*.

Most of the commonly used element types can be assigned to orphan elements. However, some element types cannot be imported from an input file. For a description of the input file reader and a full list of unsupported elements, see *Importing a model from an Abaqus input file*.

The input file reader is not able to import models with a very large number (roughly 8000 or more) of steps.

Abaqus output database (*.odb)

Abaqus/CAE imports the nodes and elements defined in the output database along with the materials, sets, surfaces, section definitions, and beam profiles and creates a new model.

Nastran input file (*.bdf, *.dat, *.nas, *.nastran, *.blk, *.bulk)

Abaqus/CAE imports entities in the Nastran input file as a new model in the current model database. The Nastran-to-Abaqus conversion enables you to select the Abaqus elements to which you map Nastran elements, select a method for converting Nastran section data into Abaqus sections, use pre-integrated shell sections, scale mass/inertia values, convert Nastran subcases to Abaqus load cases in a linear perturbation step, and couple beam element offsets.

ANSYS input file (*.cdb)

Abaqus/CAE imports entities in the ANSYS input file as a new model in the current model database. The ANSYS-to-Abaqus conversion converts selected ANSYS entities to their equivalent entities in Abaqus and notes in a log file the entities that could not be translated.

1. From the main menu bar, select **File->Import->Model**.

   The **Import Model** dialog box appears.

2. Select the type of file from which to import the model.

   - Select **Abaqus/CAE Database** (file extension `.cae`) to import a model from an Abaqus/CAE database. For more information, see *Importing a model from an Abaqus/CAE model database*.
   - Select **Abaqus Input File** (file extensions `.inp` and `.pes`) to import a model from an Abaqus input file. Abaqus/CAE imports the input file and creates a model using information from the supported options. Unsupported options and parameters are ignored. For more information, see *Importing a model from an Abaqus input file*.

- Select **Abaqus Output Database** (file extension `.odb`) to import a model from an output database. Abaqus/CAE imports the output database and creates a new model. For more information, see *Importing a model from an output database*.
- Select **Nastran Input File** (file extensions `.bdf`, `.dat`, `.nas`, `.nastran`, `.blk`, and `.bulk`) to import a model from a Nastran input file. Abaqus/CAE imports the Nastran data and creates a new model. For more information, see *Importing a model from a Nastran input file*.
- Select **Ansys Input File** (file extension `.cdb`) to import a model from an ANSYS input file. Abaqus/CAE imports the ANSYS data and creates a new Abaqus model. For more information, see *Importing a model from an ANSYS input file*.

Abaqus/CAE filters the list of available files based on your file type selection.

3. Select the file from which you want to import data, and click **OK**. For more information on specifying the file to open, see *Using file selection dialog boxes*.
4. If you are importing a model from a model database file, the **Import Model from Model Database** dialog box appears. Complete the following additional steps:
   a. Select the model that you want to copy from the selected model database.
   b. If desired, rename the model as you import it into the current model database. By default, the copied model retains its original name.
   c. If you want to copy objects between models immediately after you import the model, toggle on **After export, show "Model->Copy Objects" dialog**.
   d. Click **OK** to close the **Import Model from Model Database** dialog box.

      Abaqus/CAE imports the selected model into the current model database and, if you selected **After export, show "Model->Copy Objects" dialog**, Abaqus/CAE opens the **Copy Objects** dialog box. You can use this dialog box to select individual objects to copy between models in the current model database. For more information about selecting items in this dialog box, see *Copying objects between models*.

5. If you are importing a model from an Abaqus input file or an output database file, Abaqus/CAE completes the import process by performing the following steps:
   - Creating the new model
   - Naming the model with the same name as the original input file or output database file
   - Making the imported model the current model
   - Displaying the imported model in the model list in the context bar

6. If you are importing a model from a Nastran input file, the **Import Nastran Input File** dialog box appears. Complete the following additional steps:
   a. If desired, rename the model as you import it into Abaqus/CAE. By default, the imported model inherits the name of the Nastran input file.
   b. From the **Element Type Mapping** options, choose the Abaqus element types to which you want to map the Nastran element types **CBAR**, **CQUAD4**, **CHEXA(8)**, and **CTETRA(10)**. For each Nastran element type, click , then select the Abaqus element type by toggling element controls from the dialog box that appears.
   c. From the **Section consolidation** options, select the method you want to use to convert section or membrane data in the Nastran input file:
      - Select **Preserve section IDs** to create an Abaqus section corresponding to each Nastran PSHELL or PCOMP property ID and to create Abaqus discrete fields for shell or membrane thickness and material orientation data in Nastran.

- Select **Group by material ID** to create a single Abaqus section for all elements referencing the same material and to create Abaqus discrete fields for shell thickness and material orientation data in Nastran.
- Select **None** to create a separate shell or membrane section for each combination of orientation, offset, and/or thickness.

d. Toggle on **Use pre-integrated shell sections** to provide the section property data before the analysis. If this option is not selected, Abaqus calculates (integrates) the cross-sectional behavior from section integration points during the analysis.

e. Toggle on **Apply mass/inertia scaling from PARAM, WTMASS** to use the value on the Nastran data line `PARAM, WTMASS, value` as a multiplier for all density, mass, and rotary inertia values created in the Abaqus model.

f. Toggle on **Convert subcases to load cases in a linear perturbation step** to translate Nastran data lines that use the Nastran libraries SOL 101, SOL 108, or SOL 111 into Abaqus perturbation steps with load cases.

g. Toggle on **Couple beam element offsets** to translate beam element offsets in the Nastran data by creating new nodes at the offset locations, changing the beam connectivity to the new nodes, and rigidly coupling the new and original nodes. If this option is not selected, Abaqus translates beam element offsets to the centroid and shear center for the beam general section definition.

h. If you want to retain the Abaqus input file created during the translation process, toggle on **Keep generated Abaqus input file**.

i. Click **OK** to close the **Import Nastran Input File** dialog box.

Abaqus/CAE translates the Nastran data according to your specifications and imports the model into the current model database. If any errors arise during translation, you can display the translation log file in the **Import Nastran Log** dialog box, which enables you to browse the log file and search its contents for error messages.

Additional information

- *Using file selection dialog boxes*
- *Understanding the files generated by creating and analyzing a model*
- *Using the File menu*
- *Importing a model from a Nastran input file*
- *Importing a model from an ANSYS input file*

Exporting geometry, model, and mesh data

This section describes how you can export sketches, parts, the assembly, or the model.

In this section:

- *Exporting a sketch to an ACIS-, IGES-, or STEP-format file*
- *Exporting a part to an ACIS-, IGES-, STEP-, or VDA-format file*
- *Exporting the assembly to an ACIS-format file*
- *Exporting viewport data to a VRML-format file*
- *Exporting viewport data to a 3D XML-format file*
- *Exporting viewport data to an OBJ-format file*
- *Exporting model data to Nastran bulk data file format*

Exporting a sketch to an ACIS-, IGES-, or STEP-format file

Select **File->Export->Sketch** from the main menu bar to export the current sketch to an ACIS, IGES, or STEP file. Many computer-aided modeling and drafting applications can read and write ACIS, IGES, and STEP files; therefore, you can transfer sketches between Abaqus/CAE and these applications.

After you save a sketch in the Sketcher, you can export the sketch at any time during an Abaqus/CAE session

1. From the main menu bar, select **File->Export->Sketch**.

   The **Export Sketch** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Export Sketch** dialog box, select one of the following:

   - **ACIS SAT (*.sat)**
   - **IGES (*.igs)**
   - **STEP (*.stp)**

   Abaqus/CAE lists all the files in the selected directory with the appropriate file extension.

3. Select the file to which you want to export the sketch, or type the name of a new file in the **File Name** text field.
4. From the **Export Sketch** dialog box, click **OK**.

   Abaqus/CAE closes the dialog box and displays the **Select Sketch** dialog box with a list of all the saved sketches.

5. From the **Select Sketch** dialog box, select the sketch to export and click **OK**.
6. If you are exporting a sketch to an IGES file, Abaqus/CAE displays the **IGES Flavors** dialog box. Do the following:

   a. Choose one of the following flavors with which to write the IGES file:

   - **Standard**
   - **AutoCAD**
   - **Solid Works**
   - **JAMA**

   b. Click **OK**.

   Abaqus/CAE tailors the internal representation of the IGES file to match the format expected by the selected application. By default, Abaqus/CAE exports a sketch in a **Standard** format. Abaqus/CAE writes all of the geometry data to a single layer in the IGES file. The IGES file contains geometry data only, Abaqus/CAE does not export construction lines and dimensions from the sketch.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*

Exporting a part to an ACIS-, IGES-, STEP-, or VDA-format file

Select **File->Export->Part** from the main menu bar to export the current part to a file using one of the following formats:

- **ACIS SAT (*.sat)**
- **IGES (*.igs)**
- **STEP (*.stp)**
- **VDA (*.vda)**

Before you export a part, you must display the part that you want to export in the current viewport. You can export only the geometry of a part; orphan elements, if they exist, are not included in the exported part.

1. In the current viewport, display the part that you wish to export.
2. From the main menu bar, select **File->Export->Part**.

   The **Export Part** dialog box appears.
3. From the **File Filter** menu at the bottom of the **Export Part** dialog box, select the file format.

   Abaqus/CAE lists all the files in the selected directory with the appropriate file extension.
4. Select the file to which you want to export the part, or type the name of a new file in the **File Name** text field, and click **OK**.
5. If you are exporting a part to an IGES file, Abaqus/CAE displays the **IGES Flavors** dialog box. Do the following:

   a. Choose one of the following flavors with which to write the IGES file:

      - **Standard**
      - **AutoCAD**
      - **Solid Works**
      - **JAMA**

   b. Click **OK**.

   Abaqus/CAE tailors the internal representation of the IGES file to match the format expected by the selected application. By default, Abaqus/CAE exports a part in a **Standard** format. Abaqus/CAE writes all of the geometry data to a single layer in the IGES file.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*

Exporting the assembly to an ACIS-format file

Select **File->Export->Assembly** from the main menu bar to export the assembly to an ACIS-format file.
You can export the assembly to the ACIS part (SAT) format or to the ACIS assembly (ASAT) format.

Many computer-aided modeling and drafting applications can read and write ACIS files; therefore, you can transfer parts between Abaqus/CAE and these applications. You can export the assembly at any time during an Abaqus/CAE session. You can export the geometry of parts; orphan elements, if they exist, are not included in the exported parts. You can export the complete assembly to a single file and maintain the part and model instances. The assembly is exported with all applied transformations. Assembly-level constraints (such as the parallel face or coaxial position constraint) are not exported.

When you export the assembly to the ACIS part (SAT) format, Abaqus/CAE exports a copy of each part transformed to its position in the assembly. This file is a flat part file with no assembly structure. If you subsequently import the SAT file, Abaqus/CAE creates a part corresponding to each ACIS part in the SAT file for later use when you instance the part in the Assembly module. When you create instances of the imported ACIS parts, Abaqus/CAE uses the position information to recreate the original assembly. You cannot import the assembly directly.

When you export the assembly to the ACIS assembly (ASAT) format, Abaqus/CAE exports "true" assembly part instance information; no duplication of part information, separate position information, and model instance support. The import of assemblies in an ACIS assembly (ASAT) format is not supported.

1. From the main menu bar, select **File->Export->Assembly**.

   The **Export to ACIS File** dialog box appears.

2. From the **File Filter** menu at the bottom of the **Export to ACIS File** dialog box, select the file format.

   Abaqus/CAE lists all the files in the selected directory with the appropriate file extension.

3. Select the file to which you want to export the assembly, or enter the name of a new file in the **File Name** text field.
4. Click **OK** to export the assembly and to close the **Export to ACIS File** dialog box.

Additional information

- *Using file selection dialog boxes*
- *What is an Abaqus/CAE model?*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*

Exporting viewport data to a VRML-format file

Select **File->Export->VRML** from the main menu bar to export the model in the current viewport to a VRML-format file. VRML-format files can be used to display three-dimensional images in a web browser or a stand-alone VRML client. VRML files use meters for all lengths and distances. A special plug-in viewer, such as Cortona or Cosmo, is required to view VRML files.

You can export your model at any time during an Abaqus/CAE session to a standard VRML file or to a compressed VRML file. A compressed VRML file will generally be significantly smaller than the corresponding standard VRML file, but the compression ratio depends on the contents of the model.

VRML files are stored in an ASCII format that can be viewed in a standard text editor. Compressed VRML files must be uncompressed to convert them to ASCII format.

Animations and two-dimensional images such as sketches or layers will not be exported. (For information on exporting animations, see *Saving an animation file*.) In addition, view orientation or coordinate system triads and viewport arrow annotations will not be exported. Text fonts in the viewport image may appear smaller in the VRML file; you can increase the font size in the viewport to ensure that the text will be legible in the VRML file (see *Customizing fonts*, for more information).

When you export an image of a shell or membrane with face labels displayed, the face labels will appear in the VRML file only with respect to the view in which the image was displayed in Abaqus/CAE. If you rotate the image or otherwise change the view in a VRML viewer, the face labels shown may not be appropriate for the displayed faces. For example, if you export an image of a shell model with the positive (`SPOS`) face showing, the `SPOS` label will appear in a VRML viewer even if you rotate the model to show the negative face.

1. From the main menu bar, select **File->Export->VRML**.

   The **Export to VRML File** dialog box appears.
2. From the **File Filter** menu at the bottom of the dialog box, select one of the following:

   ### VRML (*.wrl)

   Abaqus/CAE lists all the files in the selected directory with the file extension `.wrl`.

   ### Compressed VRML (*.wrz)

   Abaqus/CAE lists all the files in the selected directory with the file extension `.wrz`.
3. Select the file to which you want to export the image, or type the name of a new file in the **File Name** text field.
4. Click **OK** to export the image and to close the **Export to VRML File** dialog box.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*

Exporting viewport data to a 3D XML-format file

Select **File->Export->3DXML** from the main menu bar to export the model in the current viewport to a 3D XML-format file. 3D XML files can be used to integrate three-dimensional images into common business applications, such as e-mail messages, web pages, word processing documents, and presentations. The 3D XML Player (available from Dassault Systèmes at no cost) can be used as a stand-alone 3D XML client, and it is required to view 3D XML files in other applications. You can also view 3D XML files in CATIA V5 without the 3D XML Player. This export capability cannot be used to transfer geometry or models to **3D**EXPERIENCE platformapps.

You can export your model at any time during an Abaqus/CAE session to a standard 3D XML file.

3D XML files are stored in an XML format based on the universal standard XML language.

To view certain line-based features (such as element edges, orientation triads, and datum planes) in an exported 3D XML file, set the view mode in the 3D XML Player to **Shading with Edges**. One- and two-dimensional features—including beam or truss elements, wireframe plots, reference points, sketches, and layers—and coordinate system triads are not exported. In addition, text is not exported; therefore, annotations, plot legends, and labels do not appear in the 3D XML file.

For visualization purposes, the 3D XML Player models a "floor" in the *X*–*Y* plane, and the default isometric view orients the *Z*-axis in the up direction. In Abaqus/CAE the default isometric view orients the *Y*-axis in the up direction. Models that appear upright in the default Abaqus/CAE view, therefore, normally appear to lie sideways on the floor in the default 3D XML Player view. To avoid this problem, Abaqus automatically rotates the model 90° about the *X*-axis and 90° about the *Z*-axis in the exported 3D XML file. These rotations make the default views of the model equivalent in Abaqus/CAE and the 3D XML Player.

The 3D XML format displays still images only. For information on exporting animations, see *Saving an animation file*.



Note:

Some 3D XML viewers do not support texture mapping in the 3D XML files. Therefore, when exporting a contour plot, you may need to switch to the tessellated rendering method to visualize colors in some 3D XML viewers; select **Tessellated** as the **Contour Method** in the **Contour Plot Options** dialog box, as described in *Choosing the contour method*.

1. From the main menu bar, select **File->Export->3DXML**.

   The **Export to 3DXML File** dialog box appears.
2. From the **File Filter** menu at the bottom of the dialog box, select **3DXML (*.3dxml)**.

   Abaqus/CAE lists all the files in the selected directory with the file extension `.3dxml`.
3. Select the file to which you want to export the image, or type the name of a new file in the **File Name** text field.
4. Click **OK** to export the image and to close the **Export to 3DXML File** dialog box.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*

Exporting viewport data to an OBJ-format file

Select **File->Export->OBJ** from the main menu bar to export the model in the current viewport to an OBJ-format file. When you export data from the Visualization module, Abaqus/CAE exports only mesh data to the OBJ-format file. If you export data from any other module, Abaqus/CAE exports mesh data if the mesh is displayed in the current viewport; otherwise, Abaqus/CAE exports geometry data.

Export from the Visualization module is sensitive to the current step and frame. If you display an undeformed plot in the viewport, Abaqus/CAE exports undeformed data to the OBJ-format file; if you display a deformed plot, Abaqus/CAE exports deformed data for the current step and frame to the OBJ-format file.

Abaqus/CAE performs the following conversions to geometry and mesh data during the export process:

- Solid elements in the model database are converted to shell elements in the OBJ-format file.
- Quadrilateral elements in the model database are converted to triangular elements in the OBJ-format file.
- Second-order elements in the model database are converted to first-order elements in the OBJ-format file.

  If your model is meshed with a large number of second-order elements, the export process can create a large OBJ file as Abaqus/CAE splits the second-order elements into the resulting first-order elements.

The quality of the geometry or mesh data you export and the resulting file size both depend on the curve refinement settings. For more information on changing curve refinement quality, see *Controlling curve refinement*.

1. If you are exporting data from the Visualization module, select the plot state for which you want to export data. For export of data from a deformed plot state, select the step and frame for which you want to export data. For more information on selecting the current step and frame, see *Selecting the results step and frame*.
2. From the main menu bar, select **File->Export->OBJ**.

   The **Export to OBJ File** dialog box appears.
3. From the **File Filter** menu at the bottom of the dialog box, select **OBJ (*.obj)**.

   Abaqus/CAE lists all the files in the selected directory with the file extension `.obj`.
4. Select the file to which you want to export the geometry and mesh data, or type the name of a new file in the **File Name** text field.
5. Click **OK** to export the data and to close the **Export to OBJ File** dialog box.

Additional information

- *Using file selection dialog boxes*
- *Using the File menu*
- *What kinds of files can be imported and exported from Abaqus/CAE?*
- *Managing display groups*
- *Managing view cuts*

Exporting model data to Nastran bulk data file format

Select **Job->Export->Nastran Input File->*jobname*** from the main menu bar in the Job module to export the model data and job data associated with an Abaqus job to Nastran bulk data file (`.bdf`) format.



Tip: You can also initiate this export from the Model Tree. Expand the **Jobs** container, click mouse button 3 on the job, and select **Export->Nastran Input File**.

The Nastran bulk data file you create includes selected data from the Abaqus model and job; see *Translating an Abaqus Input File to a Partial Nastran Bulk Data Text File*, for more information about the supported Abaqus keywords and the Nastran entities to which they map. The bulk data file includes selected history data in the Abaqus model, but it does not include parts and assemblies, which are not supported in Nastran.

Additional information

- *Using file selection dialog boxes*
- *Importing a model*
- *Translating an Abaqus Input File to a Partial Nastran Bulk Data Text File*

Creating and analyzing a model using the Abaqus/CAE modules

This part describes how to use the modules in Abaqus/CAE to define a model's geometry and other physical properties and then submit the model for analysis.

In this section:

- *The Part module*
- *The Property module*
- *The Assembly module*
- *The Step module*
- *The Interaction module*
- *The Load module*
- *The Mesh module*
- *The Optimization module*
- *The Job module*
- *The Sketch module*

The Part module

Parts are the building blocks of an Abaqus/CAE model. You use the Part module to create each part, and you use the Assembly module to assemble instances of the parts.

The tutorial in *Using Additional Techniques to Create and Analyze a Model in Abaqus/CAE* contains examples of how you create, modify, and manipulate parts. This chapter explains how you use the tools within the Part module to work with parts.

In this section:

- *Understanding the role of the Part module*
- *Entering and exiting the Part module*
- *What is feature-based modeling?*
- *How is a part defined in Abaqus/CAE?*
- *Copying a part*
- *What are orphan nodes and elements?*
- *Modeling rigid bodies and display bodies*
- *The reference point and point parts*
- *What types of features can you create?*
- *Using feature-based modeling effectively*
- *Capturing your design and analysis intent*
- *What is part and assembly locking?*
- *What are extruding, revolving, and sweeping?*
- *What is lofting?*
- *Using the Sketcher in conjunction with the Part module*
- *Understanding toolsets in the Part module*
- *Using the Part module toolbox*
- *Managing parts*
- *Using the Create Part dialog box*
- *Adding a feature to a part*
- *Adding a solid feature*
- *Adding a shell feature*
- *Adding a wire feature*
- *Adding a cut feature*
- *Using the Edit Feature dialog box*
- *Using the Edit Loft dialog box*
- *Blending edges*
- *Mirroring a part*

Understanding the role of the Part module

There are several ways to create a part in Abaqus/CAE:

- Create the part using the tools available in the Part module.
- Import the part from a file containing geometry stored in a third-party format.
- Import the part mesh from an output database.
- Import a meshed part from an Abaqus input file.
- Merge or cut part instances in the Assembly module.
- Create a meshed part in the Mesh module.

A part created using the Part module tools is called a native part and has a feature-based representation. A feature captures your design intent and contains geometry information as well as a set of rules that govern the behavior of the geometry. For example, a circular through cut is a feature, and Abaqus/CAE stores the diameter of the cut along with the information that it should pass all the way through the part. If you increase the size of the part, Abaqus/CAE recognizes that the depth of the cut should increase so that it continues to pass through the part.

You use the Part module to create, edit, and manage the parts in the current model. Abaqus/CAE stores each part in the form of an ordered list of features. The parameters that define each feature—extruded depth, hole diameter, sweep path, etc.—combine to define the geometry of the part.

The Part module allows you to do the following:

- Create deformable, discrete rigid, analytical rigid, or Eulerian parts. The part tools also allow you to edit and manipulate the existing parts defined in the current model.
- Create the features—solids, shells, wires, cuts, and rounds—that define the geometry of the part.
- Use the Feature Manipulation toolset to edit, delete, suppress, resume, and regenerate a part's features.
- Assign the reference point to a rigid part.
- Use the Sketcher to create, edit, and manage the two-dimensional sketches that form the profile of a part's features. These profiles can be extruded, revolved, or swept to create part geometry; or they can be used directly to form a planar or axisymmetric part.
- Use the Set toolset, the Partition toolset, and the Datum toolset. These toolsets operate on the part in the current viewport and allow you to create sets, partitions, and datum geometry, respectively.

Entering and exiting the Part module

You can enter the Part module at any time during an Abaqus/CAE session by clicking **Part** in the **Module** list located in the context bar. The **Part**, **Shape**, **Feature**, and **Tools** menus appear on the main menu bar, and the title bar of the current viewport displays the name of the current part, if one exists.

To exit the Part module, select any other module from the **Module** list. You need not take any specific action to save your parts before exiting the module; they are saved automatically when you save the entire model by selecting **File**->**Save** or **File**->**Save As** from the main menu bar.

What is feature-based modeling?

This section describes the feature-based modeling approach that Abaqus/CAE uses to define a part.

In this section:

- *The relationship between parts and features*
- *The base feature*
- *Simplifying a part's feature list*
- *What is a part instance?*

The relationship between parts and features

A part created in Abaqus/CAE has a feature-based representation. A feature is a meaningful piece of the design and provides the engineer with a convenient and natural way to build and modify a part. Parts created in Abaqus/CAE are constructed from an ordered list of features and the parameters that define the geometry of each feature. You select from the following shape features to build a part in the Part module:

- Solids
- Shells
- Wires
- Cuts
- Blends

Using the tools in the Part module, you create and edit all the features necessary to describe each of the parts in your model. Abaqus/CAE stores each feature and uses this information to define the entire part, to regenerate the part if you modify it, and to generate an instance of the part in the Assembly module. For more information on how parts are related to part instances, see *What is a part instance?*.

The following sequence illustrates how the three-dimensional part in *Figure 1* would be constructed using features available in Abaqus/CAE:

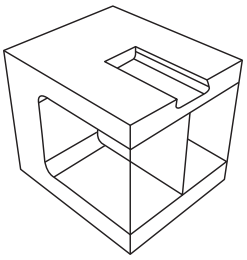


Figure 1: Part constructed using solid, shell, wire, cut, and blend features.

1. The first feature you create while building a part is called the base feature; you construct the remainder of the part by adding more features that either modify or add detail to the base feature. In this example the base feature is a U-shaped part; the user sketched a two-dimensional profile and extruded it to form the base feature, as shown in *Figure 2*.

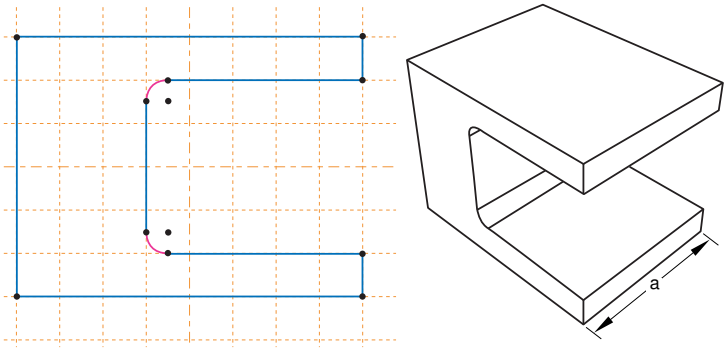


Figure 2: The base feature.

The sketch and the extrusion depth (a) are the modifiable parameters that define the base feature. You can revisit the base feature and change its size or shape by using the Feature Manipulation toolset to modify either the section sketch or the extrusion distance. If desired, you can delete the base feature and sketch a new shape.

2. A stiffening web is added as a shell feature. The user sketched a line on one of the internal faces and extruded the sketch to the opposite face, as shown in *Figure 3*. The sketch is the only modifiable parameter that defines the shell feature.

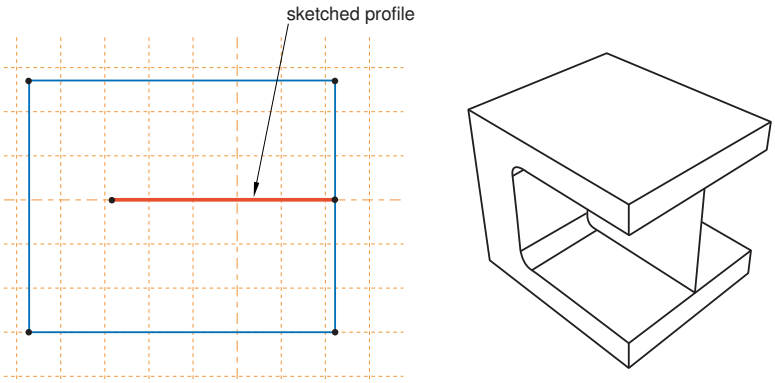


Figure 3: A shell feature.

3. Rods are added to the corners as wire features. The wire was created by connecting two points that the user selected, as shown in *Figure 4*. Wires created in this way have no modifiable parameters; they must be deleted and recreated if you need to change them.

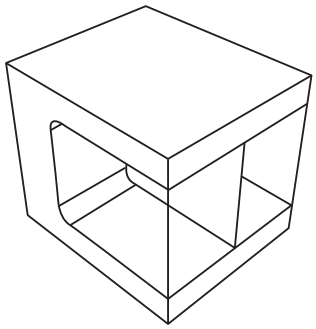


Figure 4: Wire features.

4. A blind cut is cut into the top of the clamp. The user sketched a two-dimensional profile, and the profile was extruded into the clamp through a specified distance, as shown in *Figure 5*. The sketch and the depth of the slot are the modifiable parameters that define the blind cut feature.

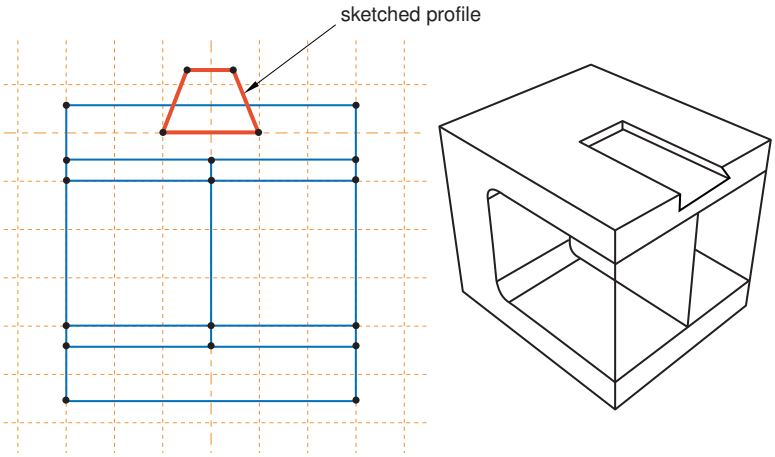


Figure 5: A cut feature.

5. The edges of the cut are rounded. The user selected the edges to round and provided the radius of the round, as shown in *Figure 6*. The radius is the modifiable parameter that defines the round feature.

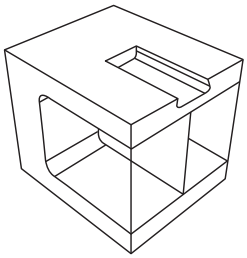


Figure 6: Round features.

If the geometry of a new feature depends on an existing feature, Abaqus/CAE creates a parent-child relationship between the features. The new feature is the child, and the feature it depends on is the parent. For example, in the part described above the round feature is a child of the cut feature. If you change the position or size of the cut, the edges remain rounded. Similarly, if you delete the cut, Abaqus/CAE also deletes the rounds.

If you modify a parent feature, the modification might invalidate children of the parent feature. For example, in the part described above if you were to increase the depth of the cut so that it became a through cut, you would lose the fillets along its edges; that is, the fillets would fail to regenerate after the modification. Abaqus/CAE offers you the following two choices:

- Keep the changes to the parent feature but suppress the features that failed to regenerate. Children of the suppressed features will also be suppressed.
- Exit the modification of the parent feature and return to the state of the last successful regeneration.

Additional information

- *What is feature-based modeling?*
- *Modifying and manipulating features*

The base feature

The first feature you create while building a part is called the base feature; you construct the remainder of the part by adding more features that either modify or add detail to the base feature. This process of building an Abaqus/CAE native part using the tools in the Part module follows a sequence of operations analogous to building a part in a machine shop. For example, you start with a piece of billet stock (the base feature) and then you do the following:

- Attach additional pieces to the billet (apply a solid extrusion, a revolved shell, or a sketched wire).
- Cut away the billet (apply an extruded cut, a revolved cut, or a circular hole; or round or chamfer an edge).

When you create a new part, you must describe the base feature. You do this by specifying two properties of the base feature: its shape and type. The shape indicates the basic topology of the feature; that is, whether it is a solid, shell, wire, or point. The type indicates which of the following methods will be used to generate the base feature:

Planar

You sketch the feature on a two-dimensional sketch plane.

Extrusion

You sketch the feature profile and then extrude it through a specified distance.

Revolution

You sketch the feature profile and then revolve it by a specified angle about an axis.

Sweep

You sketch two shapes: a sweep path and a sweep profile. The profile is then swept along the path to create the feature.

Coordinates

You enter the coordinates of a single point in the prompt area.

Before you create a part and choose the shape and the type of the base feature, you should know the sequence you will use to construct the desired part. Choosing the correct type and shape of the base feature is important.

Table 1 shows the base features that you can select based on the part's modeling space and type.

Table 1: Choosing the base feature.

	Modeling Space	
Part Type	**Three-dimensional**	**Two-dimensional or Axisymmetric**
Deformable	Any	Planar shell, planar wire, or point
Discrete rigid	Any (you must convert a 3D solid discrete rigid part to a shell before you instance it)	Planar wire or point
Analytical rigid	Extruded or revolved shell	Planar wire
Eulerian	Extruded, revolved, or swept solid	Not applicable
Fluid	Extruded, revolved, or swept solid	Not applicable
Electromagnetic	Extruded, revolved, or swept solid	Planar shell

A part imported from a file containing third-party format geometry consists of a single feature that you import into Abaqus/CAE as the base feature of a new part. You cannot modify this base feature, but you can add additional features to it. Similarly, a mesh part is created in the Mesh module or imported from an output database as the base feature of a new part. You can use the mesh editing tools to add and delete nodes and elements from a mesh, or you can use the tools in the Part module to add geometric features to the mesh.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Using the Create Part dialog box*
- *Adding a solid feature*
- *Adding a shell feature*
- *Adding a wire feature*

Simplifying a part's feature list

When you copy a part to a new part, you can reduce all the feature and parameter information to a simple definition. If you reduce the feature list, Abaqus/CAE will regenerate the part faster if you subsequently modify it; however, you will no longer be able to modify any parameters of the part. You copy a part by selecting **Part->Copy->*part name*** from the main menu bar.

Simplifying a part's feature list is especially useful if you have spent a lot of time creating a part and have iterated many times over the design. For example, if you created a slot and redimensioned the slot before arriving at the final design, the original part contains features that define each variation of the slot. If you copy the part and simplify the feature list, the new part will contain only one feature that defines the final version of the slot.

What is a part instance?

A part instance can be thought of as a representation of the original part. You create a part in the Part module and define its properties in the Property module. However, when you assemble the model using the Assembly module, you work only with instances of the part, not the part itself. The Interaction and Load modules also operate on the assembly and, therefore, on part instances. In contrast, the Mesh module enables you to operate on either the assembly or one or more of its component parts.

You create part instances in the Assembly module. You then position those instances relative to each other in a global coordinate system to form the assembly. You can create and position multiple instances of a single part. In addition, you can assemble instances of deformable, analytical rigid, and discrete rigid parts when you are solving contact problems. For more information on the types of parts you can create in Abaqus/CAE, see *Part types*.

The following example illustrates the relationship between parts and part instances. A pump housing is composed of three parts: the housing cover, a gasket, and a mounting bolt. In the Part module you create each of the three parts shown in *Figure 1*:

- One housing cover
- One gasket
- One bolt

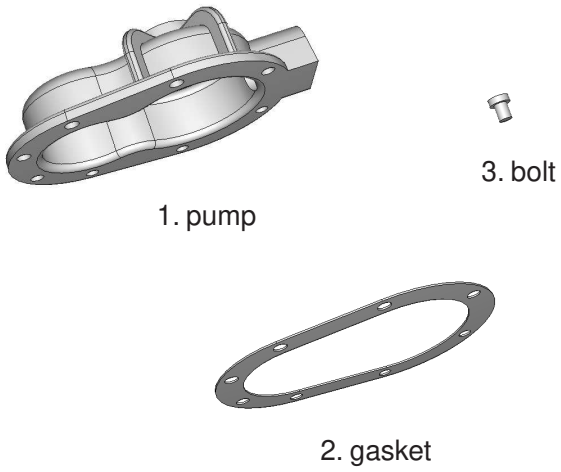


Figure 1: The original parts.

In the Assembly module you assemble instances of each part:

- One instance of the body
- One instance of the gasket
- Eight instances of the bolt

You then position the instances relative to a common coordinate system, thereby creating the model of the pump housing, as shown in *Figure 2*.



Figure 2: The model is assembled from instances of the parts.

Now, suppose you want to change the length of the bolts. You return to the Part module and modify the length of the bolt by editing the original part. When you return to the Assembly module, Abaqus/CAE recognizes that the part was modified and automatically regenerates the eight instances of the bolt to reflect the change in the length.

You cannot modify the geometry of a part instance directly; you can modify the part itself only within the Part module. When you modify a part, Abaqus/CAE automatically regenerates all instances of the modified part in the assembly. Part instances are discussed in more detail in the context of the Assembly module in *Working with part instances*.

How is a part defined in Abaqus/CAE?

This section describes the parts you can create in the Part module—deformable, rigid, and Eulerian—and the electromagnetic parts available in electromagnetic models.

In this section:

- *Part modeling space*
- *Part types*
- *Part size*

Part modeling space

When you create a new part, you must specify the modeling space in which the part will reside. You can assign the following three types of modeling space:

Three-dimensional

Abaqus/CAE embeds the part in the *X*, *Y*, *Z* coordinate system. A three-dimensional part can contain any combination of solid, shell, wire, cut, round, and chamfer features. You model a three-dimensional part using three-dimensional solid, shell, beam, truss, or membrane elements.

Two-dimensional planar

Abaqus/CAE embeds the part in the *X*–*Y* plane. A two-dimensional planar part can contain a combination of only planar shell and wire features, and all cut features are defined as planar through cuts. You model a two-dimensional planar part using two-dimensional solid continuum elements, as well as truss or beam elements.

Axisymmetric

Abaqus/CAE embeds the part in the *X*–*Y* plane with the *Y*-axis indicating the axis of revolution. An axisymmetric part can contain a combination of only planar shell and wire features, and all cut features are defined as planar through cuts. You model an axisymmetric part using axisymmetric solid continuum elements or axisymmetric shell elements.

Modeling space refers to the space in which the part is embedded rather than to the topology of the part itself. Thus, you can create a three-dimensional part using a topologically two-dimensional shell feature or a one-dimensional wire feature. You can change the modeling space of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears.

Abaqus/CAE uses the following methods to determine the modeling space of an imported part:

- When you import a part from a file containing geometry stored in a third-party format, you can specify the part's modeling space, provided that Abaqus/CAE does not determine it must be three-dimensional.
- When you import a mesh from an output database, Abaqus/CAE determines the modeling space of the new part from the information stored in the output database.
- When you import a mesh from an input file, Abaqus/CAE determines the modeling space of the new part from the element type.
- When you create a mesh part in the Mesh module, the modeling space of the mesh part is the same as the modeling space of the original part.

Detailed instructions on how to specify modeling space when creating and importing a part can be found in *Choosing the modeling space of a new part*, and *Importing sketches and parts*.

Additional information

• *How is a part defined in Abaqus/CAE?*

Part types

When you create a new part or import a part from a file containing geometry stored in a third-party format, you must choose the part's type.
The possible types for Abaqus/Standard and Abaqus/Explicit are:

Deformable

Any arbitrarily shaped axisymmetric, two-dimensional, or three-dimensional part that you can create or import can be specified as a deformable part. A deformable part represents a part that can deform under load; the load can be mechanical, thermal, or electrical. By default, Abaqus/CAE creates parts that are deformable.

Discrete rigid

A discrete rigid part is similar to a deformable part in that it can be any arbitrary shape. However, a discrete rigid part is assumed to be rigid and is used in contact analyses to model bodies that cannot deform.

Analytical rigid

An analytical rigid part is similar to a discrete rigid part in that it is used to represent a rigid surface in a contact analysis. However, the shape of an analytical rigid part is not arbitrary and must be formed from a set of sketched lines, arcs, and parabolas.

Eulerian

Eulerian parts are used to define a domain in which material can flow for an Eulerian analysis. Eulerian parts do not deform during an analysis; instead, the material within the part deforms under load and can flow across the rigid element boundaries. For more information about Eulerian analyses, see *Eulerian analyses*.

Electromagnetic

The electromagnetic part type is used only in an electromagnetic model. For more information, see *Eddy Current Analysis*.

You can assemble deformable bodies, discrete rigid parts, analytical rigid parts, Eulerian parts, and electromagnetic parts in the Assembly module. If allowed, you can change the type of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears.

Abaqus/CAE uses the following methods to determine the type of an imported part:

- When you import a part from a file containing geometry stored in a third-party format, you can specify the part's type to be either deformable, discrete rigid, or Eulerian.
- When you import a mesh from an output database, Abaqus/CAE determines the type of the new part from the information stored in the output database.
- When you import a mesh from an input file, Abaqus/CAE determines the type of the new part from the element type.
- When you create a mesh part in the Mesh module, the type of the mesh part is the same as the type of the original part.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Modeling rigid bodies and display bodies*

Part size

When you create a new part, you must choose the part's approximate size. The size that you enter is used by Abaqus/CAE to calculate the size of the Sketcher sheet and the spacing of its grid. You should set the approximate size of the part to match the largest dimension of the finished part. If you find subsequently that the part exceeds the size of the Sketcher sheet, use the Sketch customization options to increase the sheet size. You cannot change a part's approximate size after you have created it. However, you can copy the part to a new part and scale the part during the copy operation. For more information, see *Copying a part*.

Abaqus/CAE uses a geometry engine to model parts and features. The recommended approximate size limits are between 0.001 (10^{-3}) and 10000 (10^{4}) units. This size range should prevent your model from exceeding the limits of the geometry engine. For example, the minimum size supported by the geometry engine is 10^{-6}, so maintaining geometry on the order of 10^{-3} will normally allow node and element dimensions to remain above the minimum size. Parts that exceed the recommended limits may exhibit geometric defects. If you find that you need to specify an approximate size that is outside the suggested range, you should consider adopting a different set of units.

Additional information

- *Setting the approximate size of the new part*
- *Copying a part*

Copying a part

Select **Part->Copy->*part name*** from the main menu bar to copy a part to a new part.

You can create an identical copy of the original part, or you can do the following during the copy operation:

Compress features

Abaqus/CAE reduces all the feature and parameter information to a simple definition of the part. As a result, Abaqus/CAE will regenerate the part faster if you subsequently modify it; however, you will no longer be able to modify any parameters of the part. For more information, see *Simplifying a part's feature list*.

Scale part by

Abaqus/CAE scales the new part by the scale factor that you enter. If you choose to scale a part, Abaqus/CAE also compresses its features. You can use scaling to correct imported parts. If the scale of the imported part is incorrect, you can copy the part to a new part and scale it to the correct dimensions in the process. In some cases you can produce a valid part by scaling the part down, repairing the part, and then scaling the part back to its original dimensions. You can also scale an imported part during the import process. For more information, see *Importing sketches and parts*.

Mirror part about plane

Abaqus/CAE mirrors the part about the selected plane (*X*–*Y*, *Y*–*Z*, or *X*–*Z*). If you select the **Mirror part about plane** option, Abaqus/CAE selects the **Compress features** option.

To mirror a part about a plane other than one of the principal planes and without compressing the features, use **Shape->Transform->Mirror**. For more information, see *Mirroring a part*.

Separate disconnected regions into parts

In some cases when you import an IGES- or VDA-FS-format part and select the **Stitch edges** repair option, Abaqus/CAE imports separate parts as a single part. If you toggle on the **Separate disconnected regions into parts** option and copy the imported part to a new part, Abaqus/CAE will separate disconnected regions into separate parts. For more information, see *Controlling the import process*.

You can copy a mesh part and separate it into disconnected parts based on nodal connectivity. Abaqus/CAE assumes that all connected nodes belong to a single part and does not take element type into consideration. However, Abaqus/CAE ignores connectivity between axisymmetric solid elements with nonlinear, asymmetric deformation (CAXA) and some line elements (connectors, springs, dashpots, gap, and joint).

You can copy parts containing both geometry and orphan mesh features. You can use the **Compress features**, **Scale part by**, **Mirror part about plane**, and **Separate disconnected regions into parts** options on any part.

However, note the following:

1. When you compress or mirror a geometry part during the copy operation, reference points, attachment points, sets, surfaces, point parts, and datums are not copied .
2. When you compress an orphan mesh part during the copy operation, reference points, attachment points, sets, surfaces, point parts, and datums are copied.
3. When you mirror an orphan mesh part during the copy operation, reference points, attachment points, point parts, and datums are not copied. Only sets and surfaces are copied and mirrored.

What are orphan nodes and elements?

Orphan nodes and elements are components of a finite element mesh that exist without any associated geometry. In effect, the mesh information has been orphaned from its parent geometry. Orphan nodes and elements can be created in several ways; they can be:

- Imported from an output database (for more information, see *Importing a part from an output database*)
- Imported from an Abaqus input file (for more information, see *Importing a model*)
- Created as a mesh part (for more information, see *Creating a mesh part*)
- Created in a bottom-up meshing procedure (for more information, see *Bottom-up meshing*)
- Created by certain mesh editing operations, such as create element and offset (for more information, see *Using the mesh editing tools*)
- Created by deleting the associativity with their parent geometry (for more information, see *Deleting mesh-geometry associativity*)

The first three methods above create an orphan mesh feature as the base feature of a new part. The remaining methods are part of the Edit Mesh toolset; these methods edit the existing mesh and, therefore, do not exist as features of the part. For more information, see *The Edit Mesh toolset*.

You can select the face of an orphan element as the sketch plane to add geometric features. In addition, in the Mesh module you can change the element type assigned to orphan elements, and you can verify and edit the mesh.

Modeling rigid bodies and display bodies

This section describes rigid bodies and display bodies.

In this section:

- *Rigid parts*
- *Sketching the profile of an analytical rigid part*
- *What is the difference between a rigid part and a rigid body constraint?*
- *What is a display body?*

Rigid parts

When your model contains parts that contact each other, you can specify that one or more of the parts is rigid. A rigid part represents a part that is so much stiffer than the rest of the model that its deformation can be considered negligible.

In contrast to a part that you define as rigid, a part that you define as deformable can deform during contact with either a rigid part or another deformable part. For example, a model of a metal stamping process might use a deformable part to model the blank and rigid parts to model the mold and die, as shown in *Figure 1*.

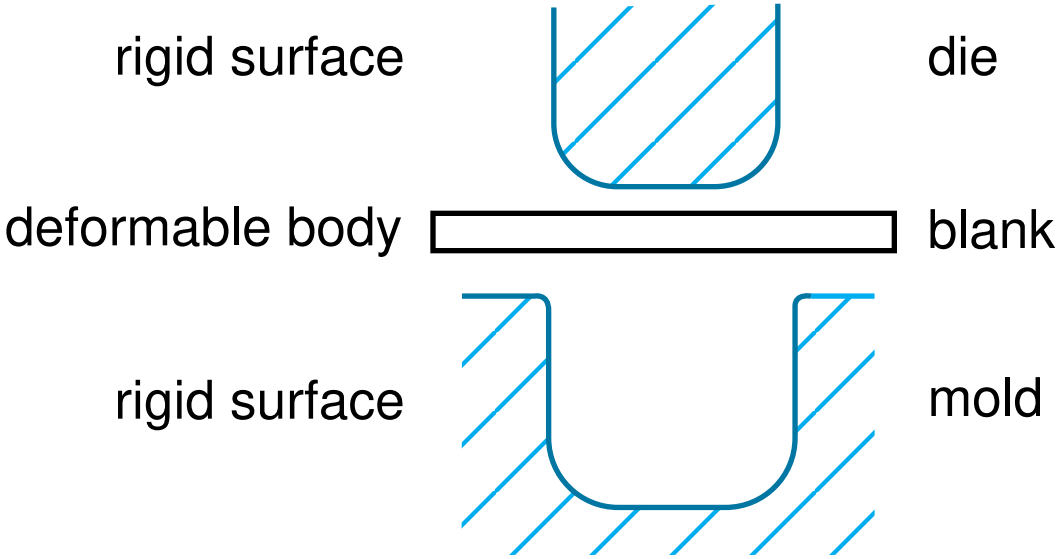


Figure 1: Rigid and deformable parts.

In this example the mold is constrained to have no motion, and the die moves through a prescribed path during the stamping process. You control the motion of rigid parts by selecting a rigid body reference point and constraining or prescribing its motion. For more information, see *The reference point*.

Computational efficiency is the principal advantage of rigid parts over deformable parts. During the analysis element-level calculations are not performed for rigid parts. Although some computational effort is required to update the motion of the rigid body and to assemble concentrated and distributed loads, the motion of the rigid body is determined completely by the reference point. To change the type of a part from deformable to rigid and vice versa, you can click mouse button 3 on the part in the Model Tree and select **Edit** from the menu that appears. For more information, see *What is the difference between a rigid part and a rigid body constraint?* and *Display bodies*.

You can choose between two kinds of rigid parts:

Discrete rigid parts

A part that you declared to be a discrete rigid part can be any arbitrary three-dimensional, two-dimensional, or axisymmetric shape. Therefore, you can use all the Part module feature tools—solids, shells, wires, cuts, and blends—to create a discrete rigid part. However, only discrete rigid parts containing shells and wires can be meshed with rigid elements in the Mesh module. If you try to create an instance of a solid discrete rigid part in the Assembly module, Abaqus/CAE displays an error message; you must return to the Part module and convert the faces of the solid to shells.

Analytical rigid parts

An analytical rigid part is similar to a discrete rigid part in that it is used to represent a rigid part in a contact analysis. If possible, you should use an analytical rigid part when describing a rigid part because it is

computationally less expensive than a discrete rigid part. The shape of an analytical rigid part is not arbitrary, and the profile must be smooth. You can use only the following methods to create an analytical rigid part:

- You can sketch the two-dimensional profile of the part and revolve the profile around an axis of symmetry to form a three-dimensional revolved analytical rigid part, as shown in *Figure 2*.

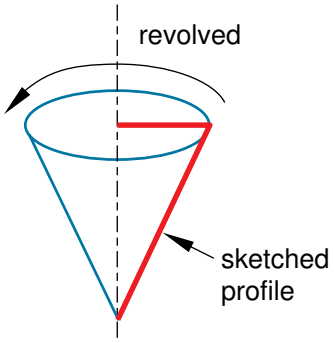


Figure 2: A revolved analytical rigid part.

- You can sketch the two-dimensional profile of the part and extrude the profile infinitely to form a three-dimensional extruded analytical rigid part. Although Abaqus/CAE considers that the extrusion extends to infinity, the Part module displays a three-dimensional extruded analytical rigid part with a depth that you specify, as shown in *Figure 3*.

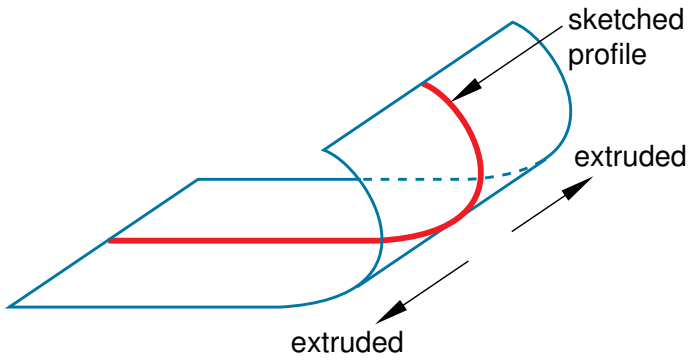


Figure 3: An extruded analytical rigid part.

- You can sketch the profile of a planar two-dimensional analytical rigid part, as shown in *Figure 4*.

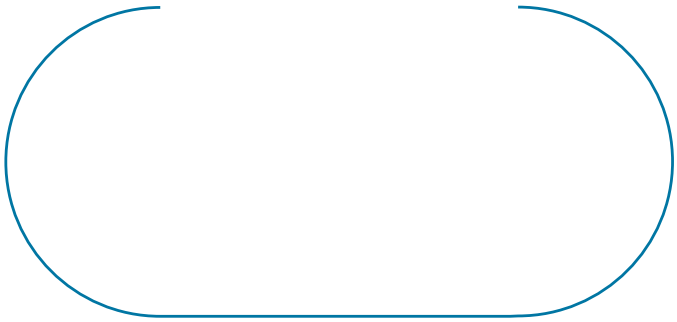


Figure 4: A planar analytical rigid part.

- You can sketch the profile of an axisymmetric two-dimensional analytical rigid part, as shown in *Figure 5*.

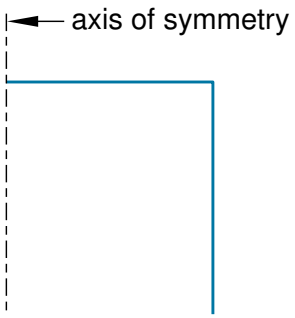


Figure 5: An axisymmetric analytical rigid part.

You can import a part from a file containing geometry stored in a third-party format and define it to be either a deformable or a discrete rigid part; however, you cannot define an imported part to be an analytical rigid part. As an alternative, you can import the geometry of the analytical rigid part into a sketch. You can then create a new analytical rigid part and copy the imported sketch into the Sketcher toolset.

A rigid part in Abaqus/CAE is equivalent to a rigid surface in an Abaqus/Standard or Abaqus/Explicit analysis. For more information, see the following:

- *Analytical Rigid Surface Definition*
- *Rigid Body Definition*
- *Rigid Elements*
- *About Contact Interactions*

Additional information

- *The reference point*
- *Part types*
- *How is a part defined in Abaqus/CAE?*
- *Sketching simple objects*

Sketching the profile of an analytical rigid part

Several tools are available in the Sketcher to help you construct each portion of the rigid part profile:

Lines

You use the Sketcher's **Line** tool to sketch straight lines.

Arcs and fillets

You use the Sketcher's **Arc** and **Fillet** tools to sketch circular arcs or to fillet two lines. Any resulting arcs must subtend an angle less than 180°; if you want to construct an arc subtending an angle greater than 180°, you should create two adjacent arcs. Abaqus/CAE displays an error message if you create an arc subtending an angle greater than 180° while sketching the profile of an analytical rigid surface.

Splines

You use the Sketcher's **Spline** tool to sketch parabolic splines. You create a parabolic spline by defining a three-point spline, where the three points are the start of the spline, a point anywhere along the spline, and the end of the spline. Only splines composed of exactly three points generate the profiles required by the analytical rigid part definition; consequently, Abaqus/CAE displays an error message if you create a spline using more than three points while sketching the profile of an analytical rigid part.

You can construct an analytical rigid part from any combination of lines, arcs, and parabolic splines; however, the resulting profile must be a single connected (but not necessarily closed) curve. In addition, the curve must be smooth to obtain a converged solution with Abaqus/Standard or Abaqus/Explicit. You may want to apply a sequence of small lines, arcs, or parabolic splines to eliminate any surface discontinuities (Abaqus/CAE does not have an equivalent to the FILLET RADIUS parameter on the Abaqus/Standard and Abaqus/Explicit **SURFACE* option). For more information on creating parabolic splines and maintaining tangency, see *Sketching splines*. For more information on the rules governing analytical rigid surfaces, see *Analytical Rigid Surface Definition*.

A sketch of an analytical rigid part that includes a line, an arc, and a fillet is illustrated in *Figure 1*.

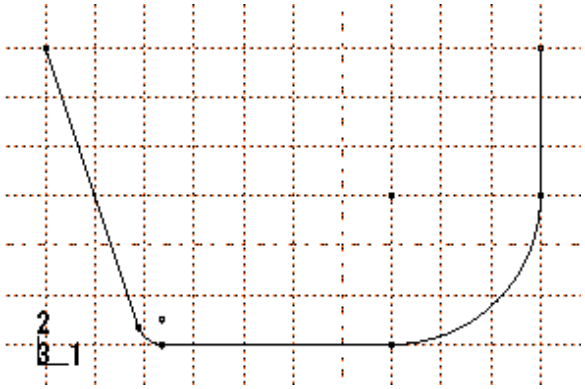


Figure 1: A sketch of an analytical rigid part.

An analytical rigid part is defined completely by the two-dimensional profile of the base feature that you create with the Sketcher; consequently, the Part module tools cannot be used to add features when you return to the Part module from the Sketcher. You can modify the part only by editing the original sketch.

After you create an analytical rigid surface, you must assign a rigid body reference point to it. You control the motion of the analytical rigid surface by constraining or prescribing the motion of the reference point. For more information, see *The reference point*.

Additional information

• *Sketching simple objects*

What is the difference between a rigid part and a rigid body constraint?

You can create a rigid part in the Part module by creating a part and declaring its type to be discrete or analytical rigid. You can create a reference point and assign it to the rigid body reference point. Motion or constraints that you apply to the reference point are then applied to the entire rigid part.

Similarly, you can create a rigid body constraint in the Interaction module. Rigid body constraints allow you to constrain the motion of regions of the assembly to the motion of a reference point. The relative positions of the regions that are part of the rigid body remain constant throughout the analysis. In addition, you can select regions from a part instance and use a rigid body constraint to specify an isothermal rigid body for a fully coupled thermal-stress analysis. For detailed instructions on defining rigid body constraints and assigning a rigid body reference point, see *Defining rigid body constraints*.

You do not have to create a reference point for a part, even if the part type is discrete or analytical rigid. However, if you do not create a reference point for a rigid part, every instance of the part in the assembly must be included in a rigid body constraint.

Rigid parts are associated with parts; rigid body constraints are associated with regions of the assembly. For example, if you define a part to be rigid, every instance of the part in the assembly is rigid. In contrast, if you define a part to be deformable, you can use rigid body constraints to make only some of the instances rigid. If you do not create a reference point in the Part module, you cannot create a rigid body reference point by associating an instance of the rigid part with a reference point created in the Assembly module. However, you can associate the instance with a rigid body constraint and a reference point created in the Assembly module.

If you define a part to be rigid, you can use the Model Tree to change the part type to be deformable. To check that your basic model is correct, you might run a quick analysis with a part defined as rigid and then change the type to deformable. Similarly, if you define a part to be deformable and apply a rigid body constraint to an instance of the part in the assembly, you can easily remove the constraint at a later time. You can run your quick analysis with a rigid body constraint applied to the part instance and then remove the constraint and run a full analysis with the part instance acting as a deformable body. The two approaches are very similar.

What is a display body?

A display body is a part instance that will be used for display only. You do not have to mesh the part, and the part is not included in the analysis; however, when you view the results of the analysis, the Visualization module displays the part along with the rest of your model. If Abaqus/CAE reports that an imported part is invalid, you can still include the part in your model as a display body. For more information, see *What is a valid and precise part?*.

You create a display body by applying a display body constraint in the Interaction module. You can apply a display body constraint to both deformable and rigid parts, and you can apply a display body constraint to parts containing both geometry and orphan elements. You can constrain the part instance to be fixed in space, or you can constrain it to follow selected points. For more information, see *Understanding constraints*. For an example of a model that uses display body constraints, see *Display bodies*.

The reference point and point parts

This section describes how you can create a reference point that is associated with a part and how you can create a part containing just a single point that is also the reference point.

In this section:

- *The reference point*
- *Point parts*

The reference point

You can use the Reference Point toolset to create a reference point that is associated with a part by selecting **Tools->Reference Point** from the main menu bar. A part can include only one reference point, and Abaqus/CAE labels it **RP**. Abaqus/CAE asks you if you want to delete the original point if you try to create a second point. A reference point on a part appears on all instances of the part in the assembly. The assembly can include more than one reference point, and Abaqus/CAE labels them **RP-1**, **RP-2**, **RP-3**, etc. For more information about the reference point, see *The Reference Point toolset*.

Abaqus/CAE displays the reference point at the desired location along with its label. You can change the reference point label by selecting **Rename** from the Model Tree. If desired, you can turn off the display of the reference point symbol and the reference point label; for more information, see *Controlling reference point display*.

If the part is a discrete or analytical rigid part, you use the reference point to indicate the rigid body reference point. When you create the assembly, the reference point appears on each instance of the part. You use the Interaction module to apply constraints to the reference point or the Load module to define the motion of the reference point using loads or boundary conditions. Motion or constraints that you apply to the reference point are then applied to the entire rigid part.

Point parts

When you create a part, you can choose the shape of its base feature to be a solid, a shell, a wire, or a point. If you select a point, you must specify the coordinates of the point and Abaqus/CAE creates a part for which the base feature is a point at that location. In addition, the point is the reference point for the part. The modeling space of a point part can be three-dimensional, two-dimensional, or axisymmetric. The type of a point part can be either deformable or rigid.

You can continue to add features to a point part, such as datums and wires. More typically, you will use a point part to simplify your model by replacing a rigid part with a point part that has mass and inertia. You can add mass to a rigid point part; see *Inertia*. In addition, you can attach a display body to the point and use the display body to represent the original rigid part; see *Display bodies*. Finally, you can constrain the point part to your model by modeling a connector such as JOIN or REVOLUTE; see *Connectors*.

What types of features can you create?

After you select the type and shape of the part and sketch the two-dimensional profile of its base feature, you add additional features or modify existing features to create the finished part.

In this section:

- *Solid features*
- *Shell features*
- *Wire features*
- *Cut features*
- *Blend features*

Solid features

To create a solid feature, select **Shape->Solid** from the main menu bar or select one of the solid tools in the Part module toolbox. Once you have sketched the initial profile, you perform one of the following operations to create the feature:

- To create an extruded solid feature, you extrude the profile through a specified distance (*d*), as shown in *Figure 1*.

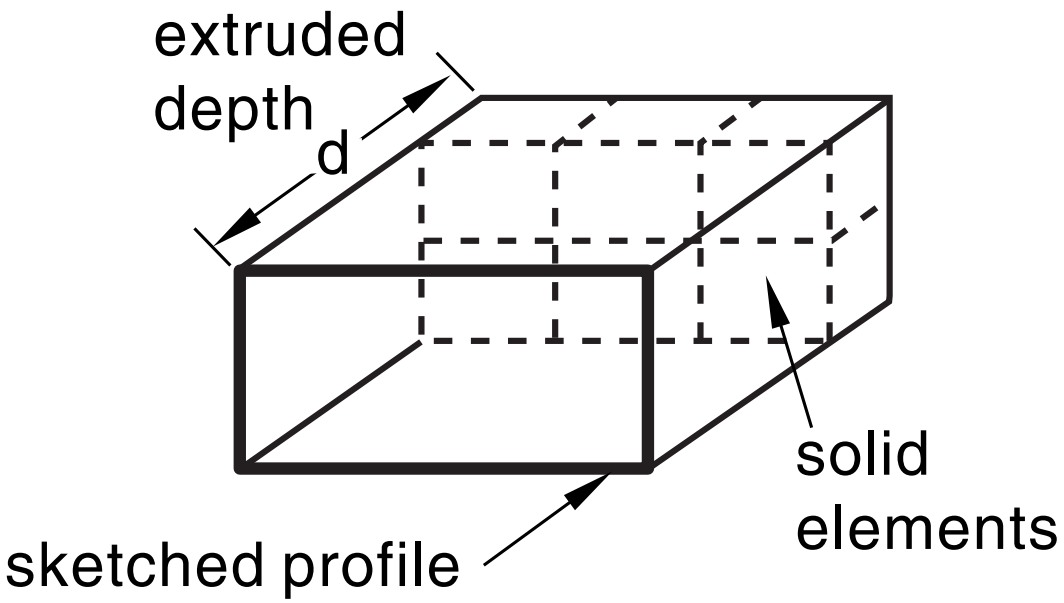


Figure 1: An extruded solid feature.

In addition, you can apply either draft or twist to the extrusion, as shown in *Figure 2*.

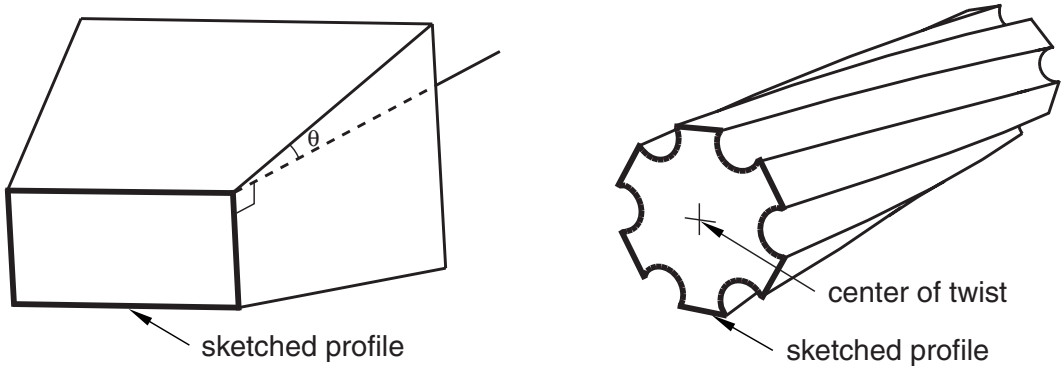


Figure 2: An extruded solid feature with draft and one with twist.

You define the draft angle for an extrusion with draft or the center of twist and the pitch (the extrusion distance in which a 360° twist occurs) for an extrusion with twist. Select **Shape->Solid->Extrude** from the main menu bar to create this type of feature.

- To create a solid loft feature, you transition the shape from the initial loft section to an end section of a different shape or orientation. Abaqus/CAE determines the shape between the start and end sections using tangency constraints, intermediate sections, and a path curve. A simple loft (with only two loft sections, no tangency constraints, and a straight path) is shown in *Figure 3*. Select **Shape->Solid->Loft** from the main menu bar to create this type of feature.

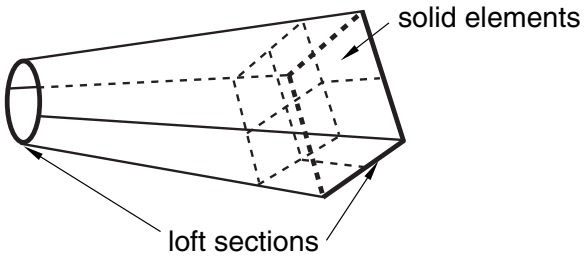


Figure 3: A solid loft feature.

- To create a revolved solid feature, you revolve the profile through a specified angle (α). A construction line serves as the axis of revolution, as shown in *Figure 4*.

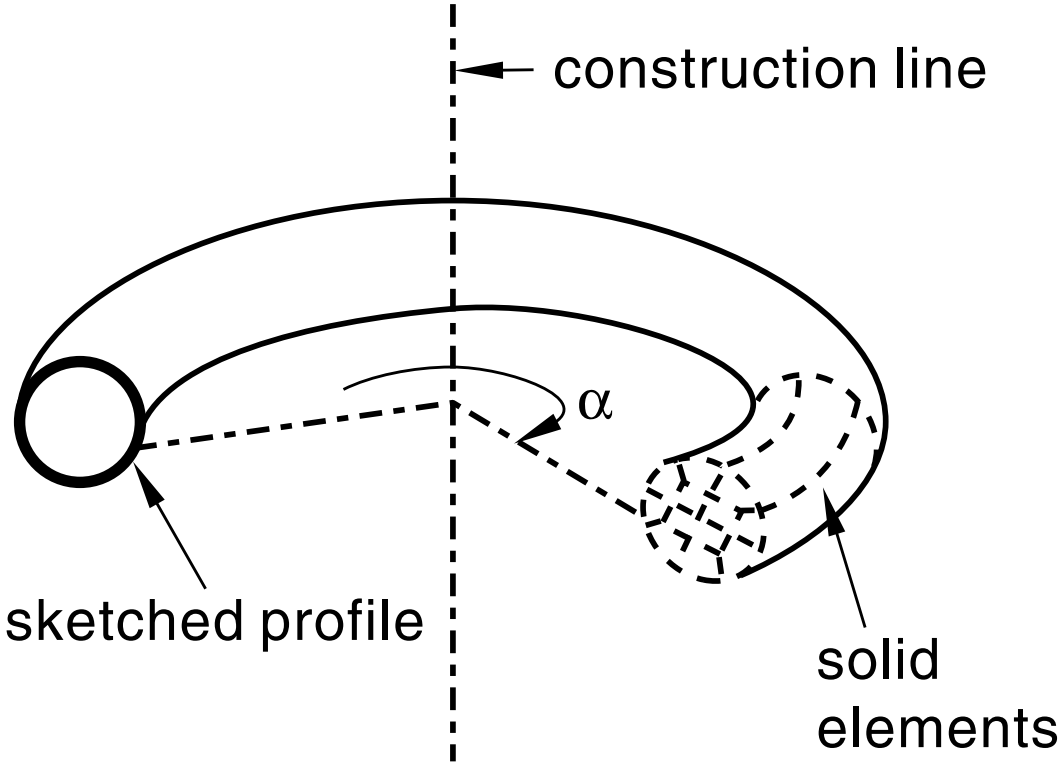


Figure 4: A revolved solid feature.

In addition, you can enter a pitch value (p) to translate the profile along the axis of revolution as it is revolved; *Figure 5* shows a solid revolved 360° with pitch. Select **Shape->Solid->Revolve** from the main menu bar to create this type of feature.

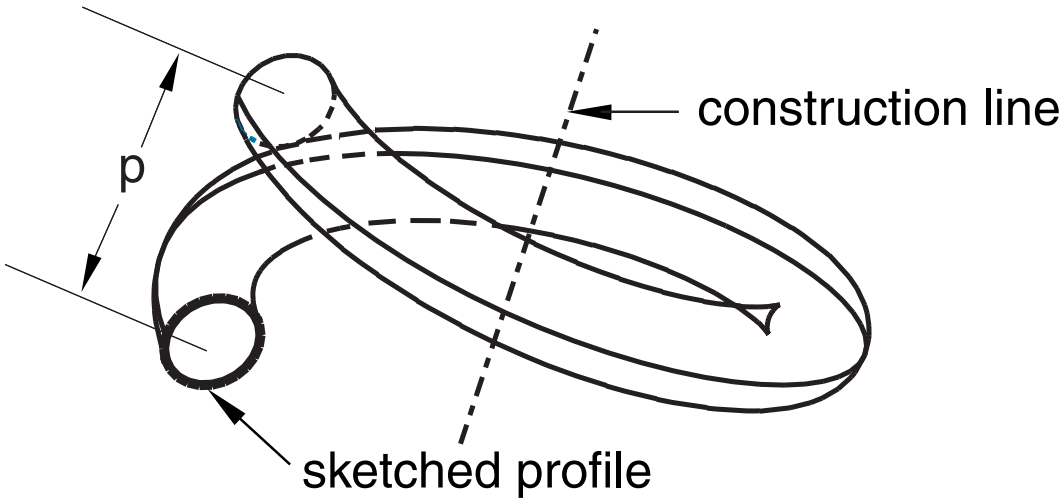


Figure 5: A 360° revolved solid feature with pitch (*p*).

- To create a swept solid feature, you sweep the profile along a specified path, as shown in *Figure 6*. Select **Shape->Solid->Sweep** from the main menu bar to create this type of feature. For more information, see *Defining the sweep path and the sweep profile*.

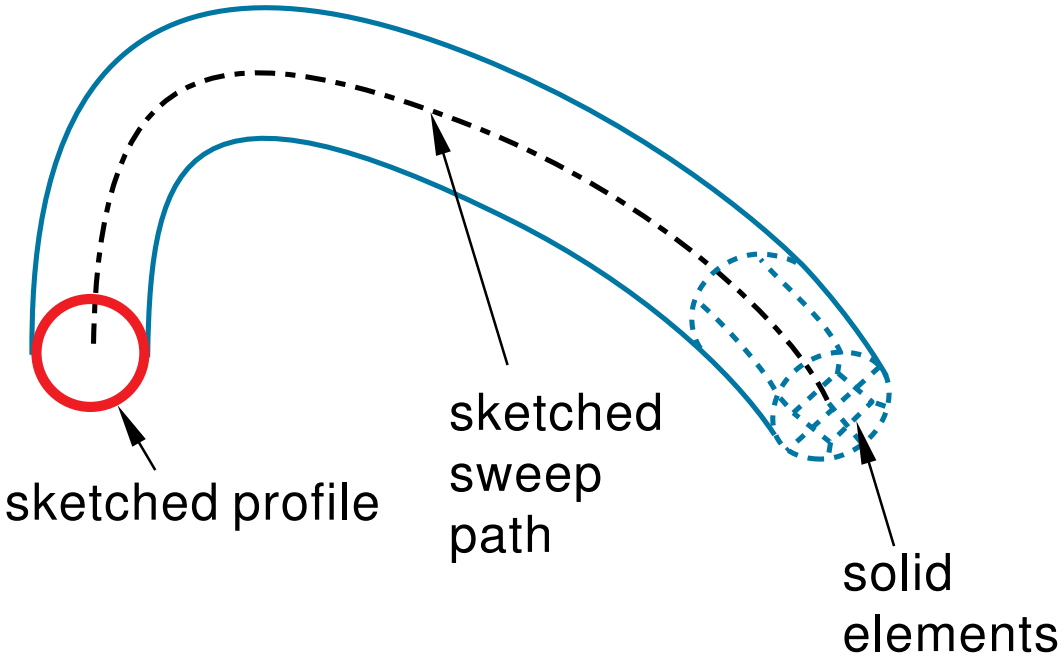


Figure 6: A swept solid feature.

You can use any of the solid tools to add a solid feature to a deformable or discrete part that you created in three-dimensional modeling space. You cannot add a solid feature to a two-dimensional or axisymmetric part.

Figure 1, *Figure 3*, *Figure 4*, and *Figure 6* illustrate how each feature might later be meshed. You can mesh a solid feature using any of the three-dimensional, solid continuum elements available in Abaqus/Standard or Abaqus/Explicit.

Additional information

- *Adding a solid feature*
- *What is feature-based modeling?*
- *Meshing complex solids with hexahedral elements*

Shell features

A shell feature is an idealization of a solid in which thickness is considered small compared to the width and depth. To create a shell feature, select **Shape->Shell** from the main menu bar or select one of the shell tools in the Part module toolbox. You create a shell feature by using the shell tools to do one of the following:

- To create an extruded shell feature, you sketch a profile and extrude it through a specified distance (*d*), as shown in *Figure 1*.

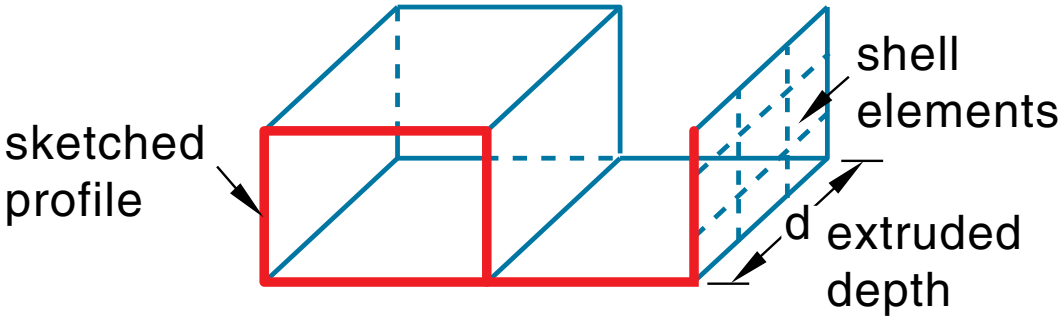


Figure 1: An extruded shell feature.

In addition, you can apply either draft or twist to the extrusion, as shown in *Figure 2*.

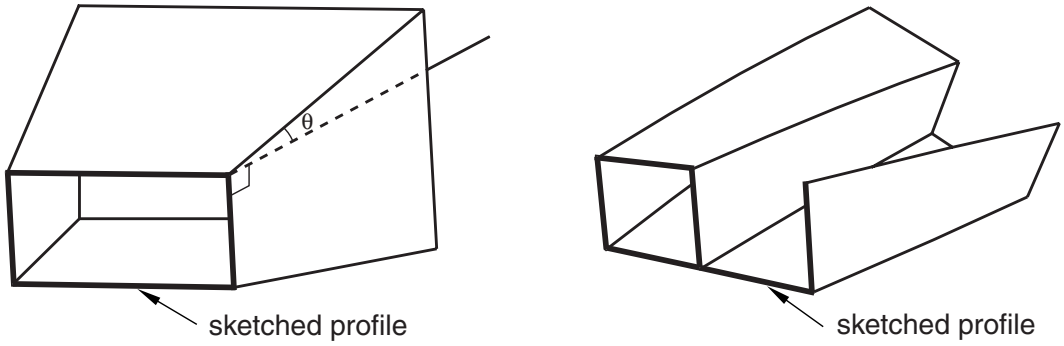


Figure 2: An extruded shell feature with draft and one with twist.

You define the draft angle for an extrusion with draft or the center of twist and the pitch (the extrusion distance in which a 360° twist occurs) for an extrusion with twist. Select **Shape->Shell->Extrude** from the main menu bar to create this type of feature.

- To create a shell loft feature, you transition the shape from the initial loft section to an end section of a different shape or orientation. Abaqus/CAE determines the shape between the start and end sections using tangency constraints, intermediate sections, and a path curve. A simple loft (with only two loft sections, no tangency constraints, and a straight path) is shown in *Figure 3*. Select **Shape->Shell->Loft** from the main menu bar to create this type of feature.

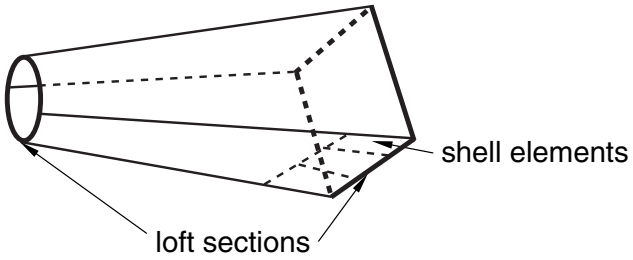


Figure 3: A shell loft feature.

- To create a revolved shell feature, you sketch a profile and revolve it through a specified angle ( ). A construction line serves as the axis of revolution, as shown in *Figure 4*.

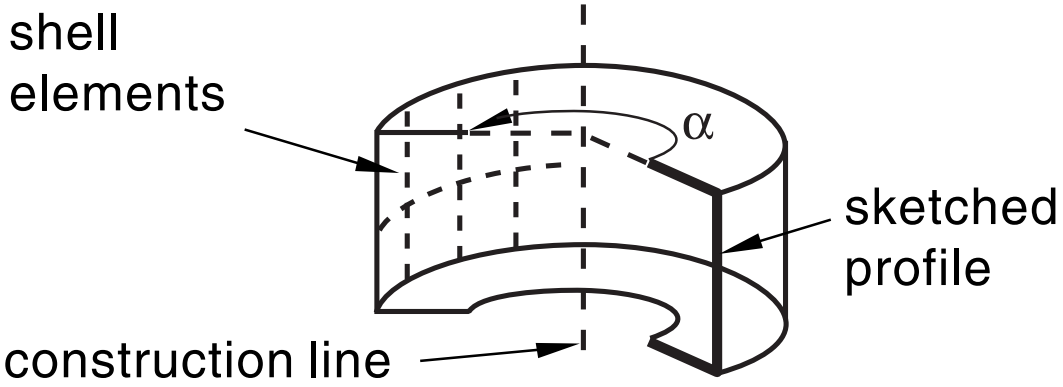


Figure 4: A revolved shell feature.

In addition, you can enter a pitch value to translate the profile along the axis of revolution as it is revolved; *Figure 5* shows a revolved shell with pitch.

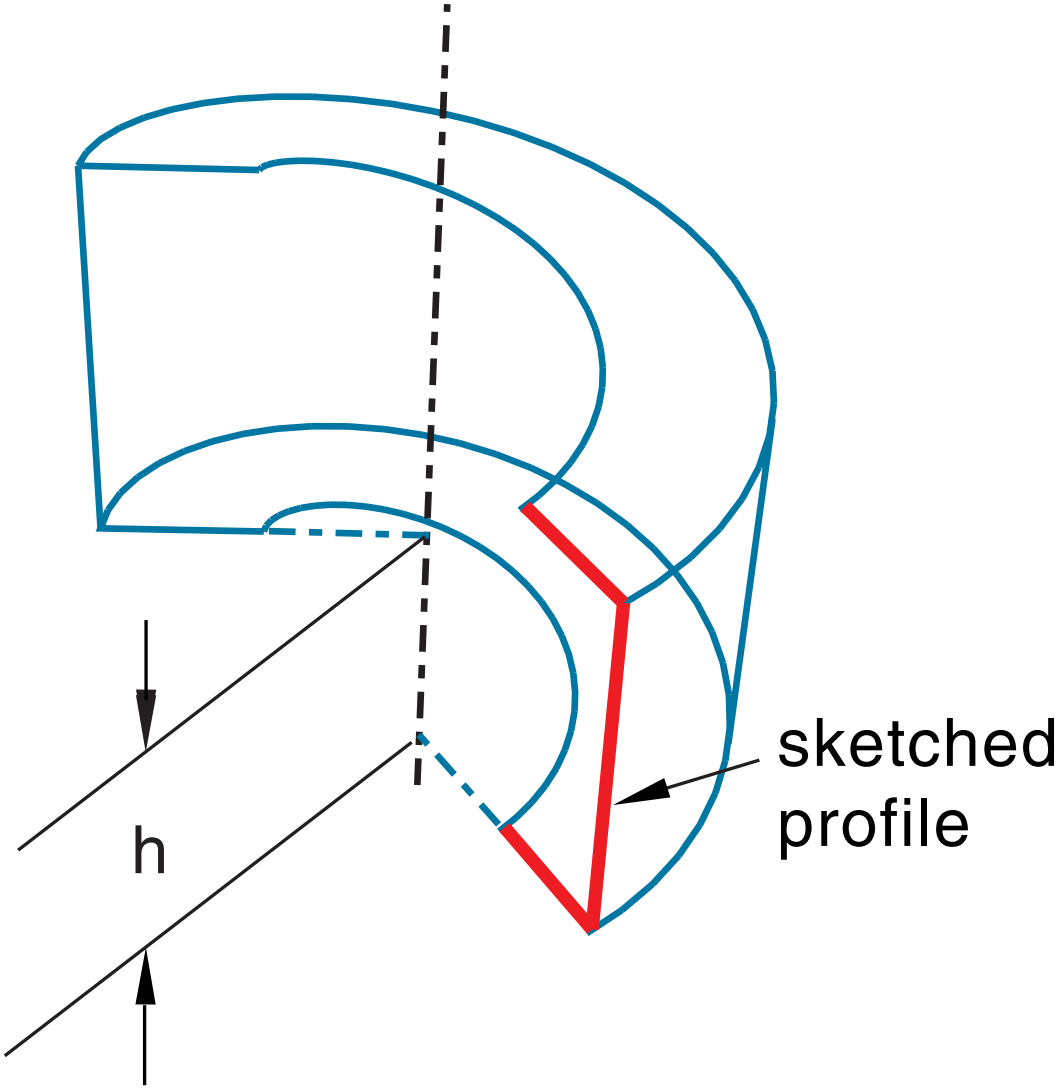


Figure 5: A revolved shell feature with pitch.

The dimension *h* represents the translation of the sketched profile due to pitch; *h* would be equal to the pitch if the part was revolved a full 360°. Select **Shape**->**Shell**->**Revolve** from the main menu bar to create this type of feature.

- To create a swept shell feature, you sketch a profile and sweep it along a specified path, as shown in *Figure 6*.

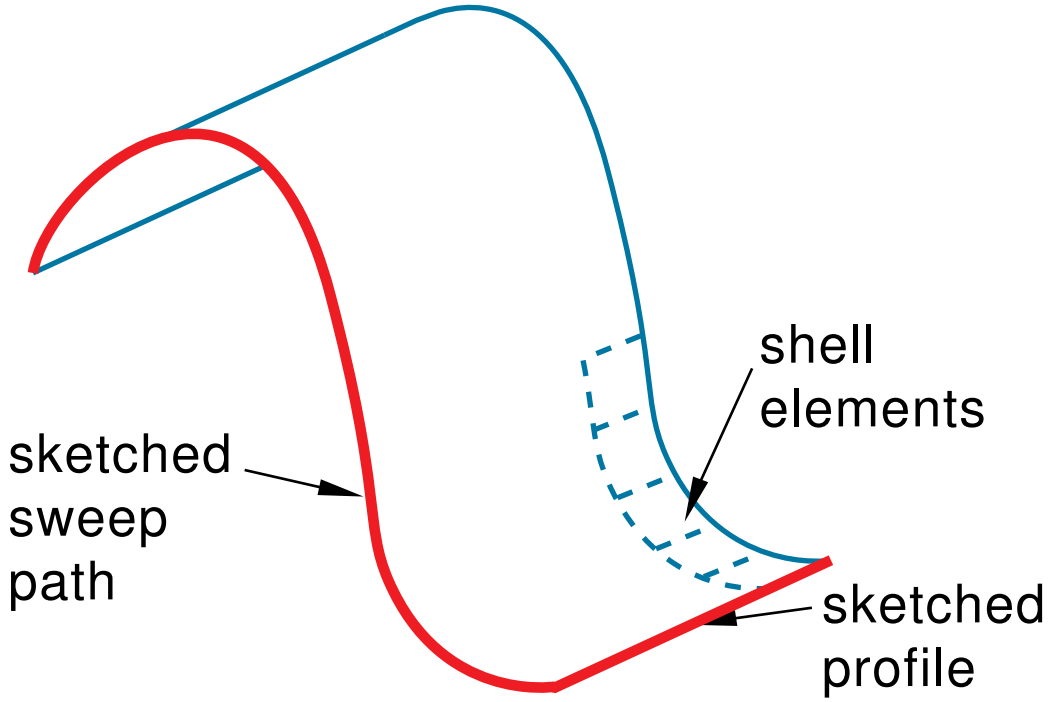


Figure 6: A swept shell feature.

Select **Shape->Shell->Sweep** from the main menu bar to create this type of feature. For more information, see *Defining the sweep path and the sweep profile*.

- To create a planar shell feature, you sketch the outline of the shell on a selected planar face or datum plane, as shown in *Figure 7*.

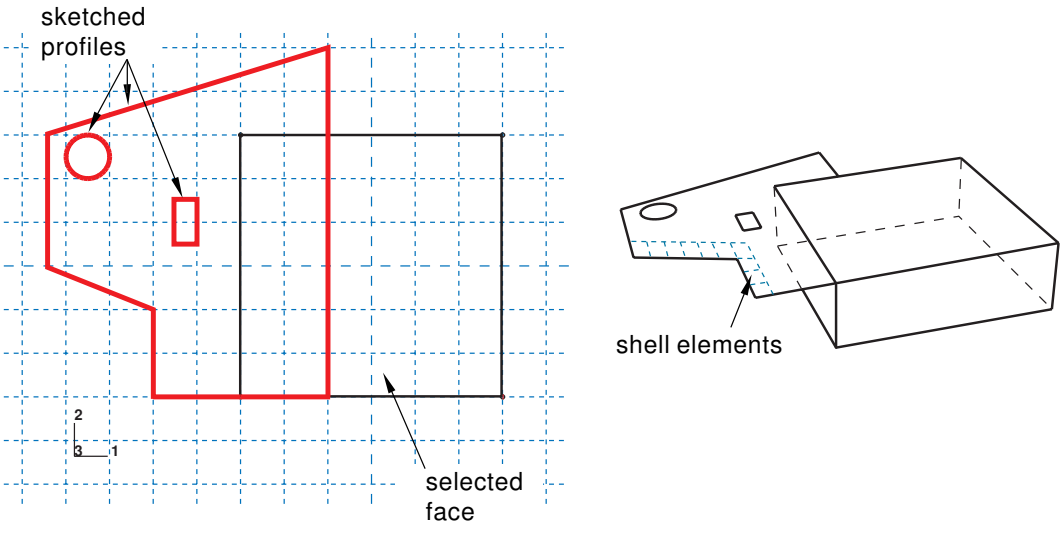


Figure 7: A sketched shell feature.

When you sketch on a planar face (for example, the side of a cube), the shell feature is created only where it extends beyond the face; a shell feature cannot overlap a face. A sketch on a planar face of a cube and the resulting shell feature are shown in *Figure 7*. In this example the shell feature is a fin extending beyond the selected face of the cube. Select **Shape->Shell->Planar** from the main menu bar to create this type of feature.

- To create a shell-from-solid feature, you convert the faces of a solid feature to shell features; in effect, hollow out a solid. A shell-from-solid feature is shown in *Figure 8*. Select **Shape->Shell->From Solid** from the main menu bar to create this type of feature.

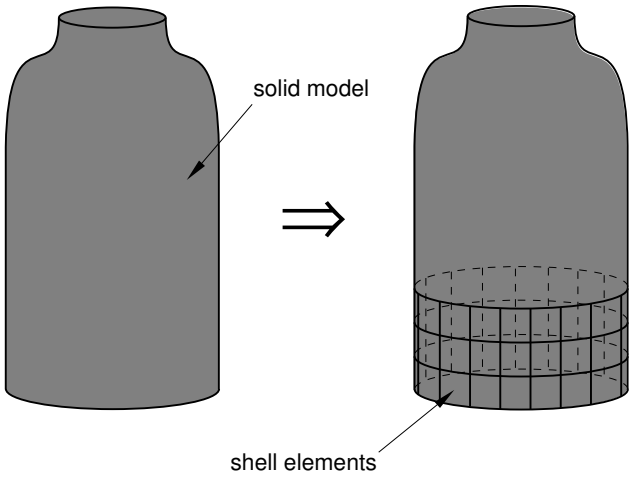


Figure 8: A shell-from-solid feature.

You can use any of the shell tools to add a shell feature to a part that you created in three-dimensional modeling space; however, when you are working on parts created in two-dimensional or axisymmetric modeling space, you can use only the planar shell tool to add a shell feature. You use the Property module to create a section prescribing the desired thickness and to assign the section to the shell feature. For more information, see *Defining sections*, and *Which properties can I assign to a part?*.

Many of the figures illustrate how the shell features might later be meshed. You can mesh a shell feature using:

- Two-dimensional or axisymmetric continuum elements (limited to planar shell features)
- Three-dimensional shell elements
- Membrane elements

Additional information

- *Adding a shell feature*
- *What is feature-based modeling?*

Wire features

A wire is depicted as a line in Abaqus/CAE and is used to idealize a solid in which both its thickness and depth are considered small compared to its length. To create a wire feature, select **Shape->Wire** from the main menu bar or select one of the wire tools in the Part module toolbox. You create a wire feature in the Part module using the wire tools to do one of the following:

- Sketch a wire on a selected planar face or datum plane to create a sketched wire feature, as shown in *Figure 1*. Select **Shape->Wire->Sketch** from the main menu bar to create this type of feature.

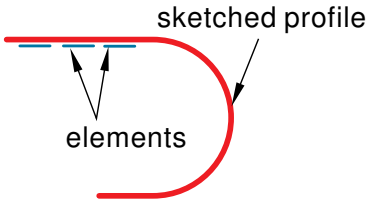


Figure 1: A sketched wire feature.

When you sketch on a planar face (for example, the side of a cube), the wire feature is created only where it extends beyond the face.

- Connect two or more points with straight lines, as shown in *Figure 2*, or with a spline curve, as shown in *Figure 3*. Select **Shape->Wire->Point to Point** from the main menu bar to create this type of feature. Select **Polyline** or **Spline** for the **Geometry Type** to create straight lines or a spline curve, respectively. You can choose to imprint the wire on the existing part by creating edges, merge the wire with the existing part, or create the wire separate from the existing part. The rectangular solid feature in *Figure 3* is shown for reference. The image on the left shows the full length of the spline wire using the **Imprint wire** or **Separate wire** options, while the image on the right shows a spline wire connecting the same set of points using the **Merge wire** option. You can create geometry sets that include the wires and vertices defined in the wire feature.

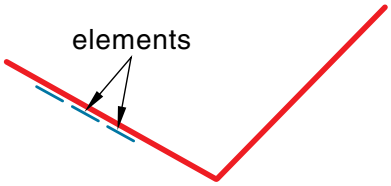


Figure 2: A wire feature connecting three points.

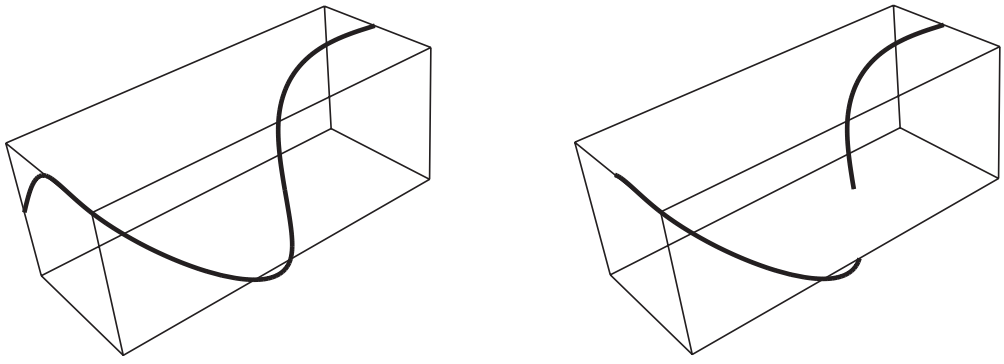


Figure 3: A wire feature connecting several points of a solid feature.

You can use the wire tools to add a wire feature to any deformable or discrete rigid part. You cannot add a wire feature to an analytical rigid part; you can only modify the original sketch that defined that part.

You use the Property module to create a section that prescribes the desired cross-sectional geometry and to assign that section to the wire feature. (For more information, see *Defining sections*, and *Which properties can I assign to a part?*.) You can model a wire feature using any of the beam, truss, or axisymmetric shell elements available in Abaqus/Standard or Abaqus/Explicit.



Note:

Although you can create a mesh of beam elements, the current release of Abaqus/CAE allows you to assign only the following sections to a wire:

- Beam section
- Truss section

Additional information

- *Adding a wire feature*
- *What is feature-based modeling?*

Cut features

A cut is a feature that removes material from a part. A cut can be a circular hole, or it can be any arbitrary shape. The sketched profile of any cut must be closed. In many cases the entire profile will affect the shape of the cut feature, even if it does not initially contact the surface being cut. To create a cut feature, select **Shape->Cut** from the main menu bar or select one of the cut tools in the Part module toolbox.



Note:

Most of the figures do not show a closed cut profile where it intersects with the part surface. These lines have been removed to show the shape of the cut feature.

Once you have sketched the initial profile, you perform one of the following operations to create a cut feature:

- To create an extruded cut, you extrude the profile through a specified distance (*d*), as shown in *Figure 1*.

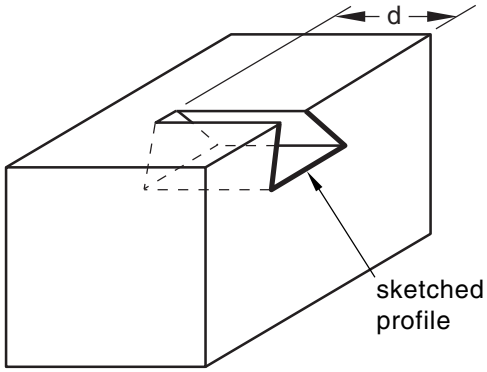


Figure 1: An extruded cut feature.

In addition, you can apply either draft or twist to the extruded cut, as shown in *Figure 2*. You define the draft angle for an extruded cut with draft or the center of twist and the pitch (the extrusion distance in which a 360° twist occurs) for an extruded cut with twist. Select **Shape->Cut->Extrude** from the main menu bar to create this type of feature.

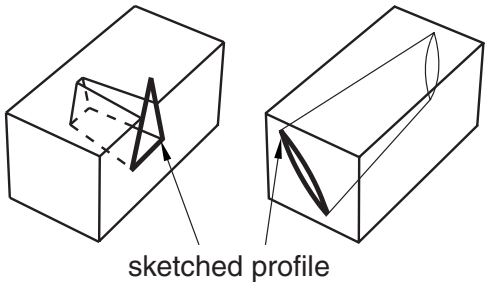


Figure 2: Extruded cut features with draft and twist.

- To create a cut loft feature, you transition the shape from the initial loft section to an end section of a different shape or orientation, as shown in *Figure 3*. Abaqus/CAE determines the shape between the start and end sections using tangency constraints, intermediate sections, and path curves. Select **Shape->Cut->Loft** from the main menu bar to create this type of feature.

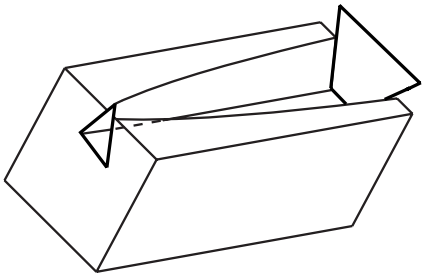


Figure 3: A cut loft feature.

- To create a revolved cut, you revolve the profile through a specified angle ( ). A construction line serves as the axis of revolution. In addition, you can enter a pitch value to translate the profile along the axis of revolution as it is revolved and to create part details such as screw threads. *Figure 4* shows a revolved cut and a revolved cut with pitch. Select **Shape->Cut->Revolve** from the main menu bar to create this type of feature.

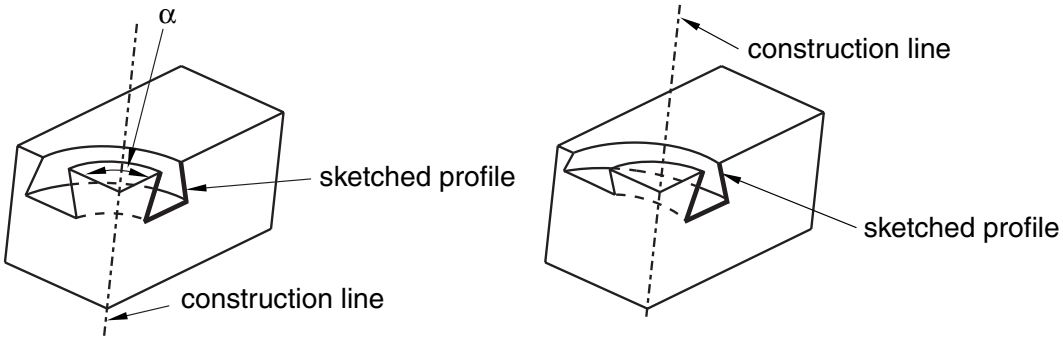


Figure 4: A revolved cut and a revolved cut with pitch.

- To create a swept cut, you sweep the profile along a specified path, as shown in *Figure 5*. Select **Shape->Cut->Sweep** from the main menu bar to create this type of feature. For more information, see *Defining the sweep path and the sweep profile*.

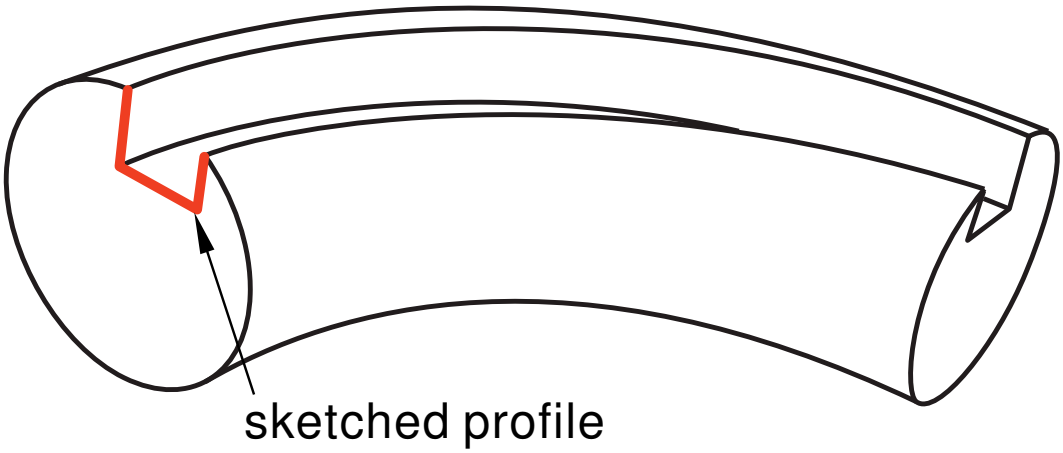


Figure 5: A swept cut feature.

- To create a circular hole, you enter the diameter of a hole and the distance of its center from two selected edges, as shown in *Figure 6*. Select **Shape->Cut->Circular Hole** from the main menu bar to create this type of feature.

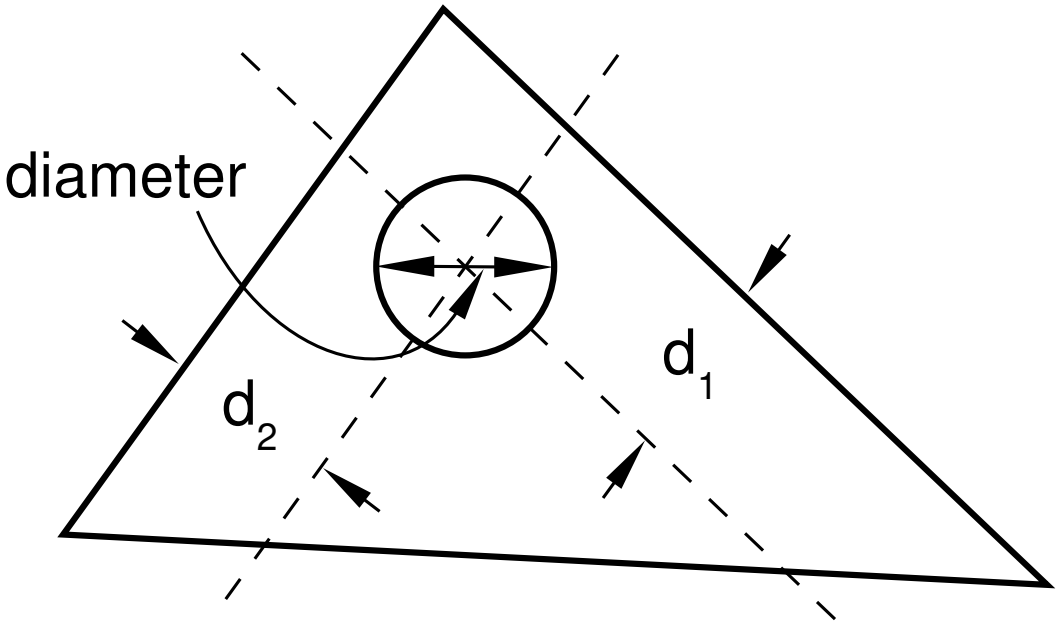


Figure 6: A circular hole feature.

When you are sketching the profile of an extruded, revolved, or swept cut, you can sketch multiple profiles in a single sketch. Abaqus/CAE extrudes each of the profiles when you exit the Sketcher and creates a cut corresponding to each profile as shown in *Figure 7*.

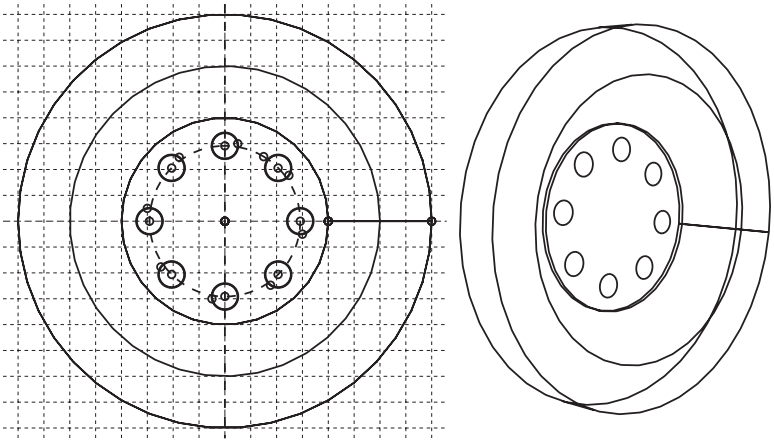


Figure 7: Multiple profiles extruded from a single sketch.

The sequence of cuts is stored as a single feature, and you can edit it only as a single feature. For example, if you change the extrusion depth, the depth will change for all the cuts in the feature.

You can use the cut tools to add a cut feature to any deformable or discrete rigid part. You cannot add a cut feature to an analytical rigid part; you can only modify the original sketch that defined that part.

Additional information

- *Adding a wire feature*
- *What is feature-based modeling?*

Blend features

A blend feature smooths an edge of a three-dimensional solid part. To create a blend feature, select **Shape->Blend** from the main menu bar or select one of the blend tools in the Part module toolbox. You create a blend feature in the Part module using the blend tools to do one of the following:

- Smooth an edge with a circular blend of a specified radius, as shown in *Figure 1*. Select **Shape->Blend->Round/Fillet** from the main menu bar to create this type of feature.

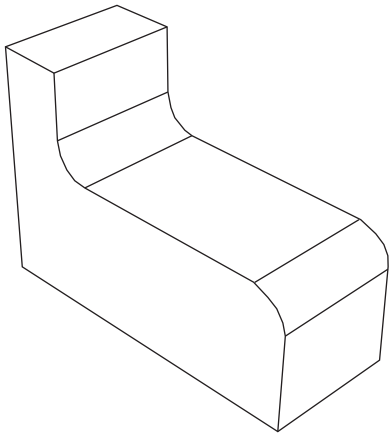


Figure 1: A round/fillet blend feature.

- Bevel an edge with a chamfered blend of a specified length, as shown in *Figure 2*. Select **Shape->Blend->Chamfer** from the main menu bar to create this type of feature.

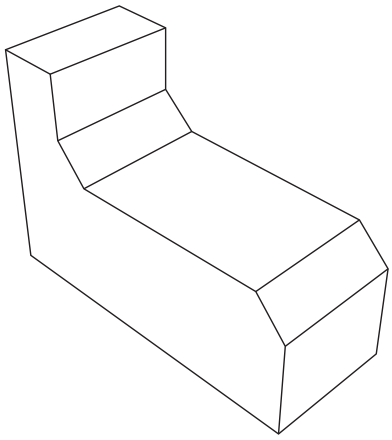


Figure 2: A chamfer blend feature.

You can use the blend tools to blend edges of a deformable or discrete rigid part that you created in three-dimensional modeling space. You cannot add a blend feature to a two-dimensional or axisymmetric part; however, you can blend its corners by editing the sketch of the part.

Additional information

- *Adding a wire feature*
- *What is feature-based modeling?*

Using feature-based modeling effectively

You can devise a more efficient approach to creating a part if you understand how Abaqus/CAE uses feature-based modeling and how the rules that define a feature are applied. The following techniques will help you create robust parts that can be modified easily:

Plan a strategy

Feature-based modeling provides flexibility, but it can also add overhead to your model. For example, you can suppress an extrusion using the suppress tool in the Feature Manipulation toolset. Alternatively, you could effectively suppress the extrusion by removing it with a cut feature. Although you could restore the extrusion subsequently by removing the cut feature, the resulting part contains additional feature-based information that can slow down regeneration. Regeneration speed can be improved by using the geometry cache to save parts in different states, but the cache uses system memory that may be needed for other operations (for more information, see *Tuning feature regeneration*). In addition, dependencies may cause feature regeneration to fail if you add more detail to the part; and, because the extrusion is no longer visible, the cause of the failure to regenerate may be hard to determine.

Before you decide how to create a part, you should always consider if you will ever need to modify the part in the future. If you decide that you might need to modify the part, you should consider the techniques that you will use to create the features that define the part. The simplest techniques may not provide the flexibility you need for modifying the features. You may find it cumbersome to edit or suppress individual items of geometry, such as an extrusion, a fillet, or a hole.

Alternatively, if you know that you will never change the final design, you may not need the flexibility provided by feature-based modeling and can use the simplest and most convenient techniques to define the part.

In general, you should try to finish creating your parts in the Part module before you start creating part instances and positioning them in the assembly. You should try to finish creating all your parts before you apply attributes to the assembly, such as sets, loads, and boundary conditions. If you apply attributes to the assembly and then return to the Part module to modify the original part, Abaqus/CAE may not be able to determine where the attribute should be applied. For example, if you apply a pressure load to a face and then return to the Part module to partition the face into two regions, Abaqus/CAE will apply the pressure to only one of the regions.

Use reference geometry

When you are adding a feature to a part, you should always use underlying reference geometry to define the new feature's location relative to existing features. While sketching a feature, you may be able to select reference geometry directly; for example, if you are sketching a circle, you may be able to select a vertex from the reference geometry to define its center. Alternatively, you may have to add a dimension between reference geometry and the new feature. If you do not use reference geometry to position the sketch of a new feature and you subsequently modify the part, the resulting changes to the feature can be unpredictable.

Use dimensions

Dimensions add clarity to the sketches that define features and document your design intent for future reference. Dimensions also add constraints to your sketches. You can modify dimensions in the Sketcher, and the part and assembly will regenerate accordingly.

Pay attention to the order in which you create features

A new feature of a part is aware of existing features. In addition, if the new feature depends on an existing feature for positioning information, Abaqus/CAE creates a parent-child relationship between the features. Parent-child relationships and the order in which you created features play an important role in feature regeneration.

A modeling scheme that is carefully ordered and follows the sequence below is less likely to run into unnecessary or ill-conditioned modeling problems:

1. Create the basic geometry of a part using extrusions, revolutions, cuts, and sweeps.
2. Add extruded, revolved, swept, and planar features.
3. Add round or fillet features.
4. Add partitions only when the rest of the geometry is complete.

Allow for some overlap

If possible, you should allow for overlap between an existing feature and a feature that fills a hole or cuts a hole. Allowing for overlap makes your part robust, and the features are more likely to regenerate successfully. For example, when you cut a slot, extend its sketched profile above the surface you are cutting, as shown in *Figure 1*.

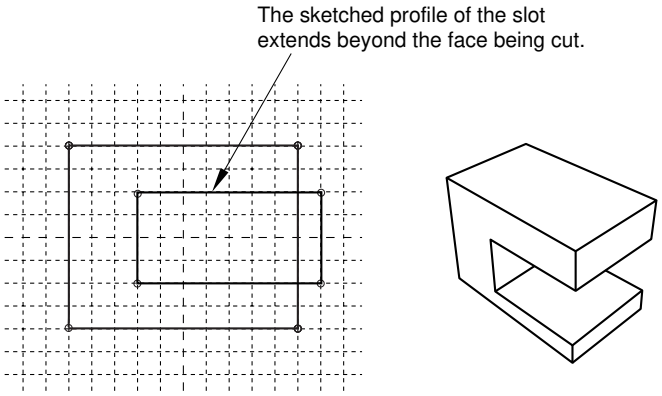


Figure 1: The sketched profile of a slot should extend beyond any surfaces that are cut.

Create solids where possible

Solid features are more robust than shell features. You may find it hard to position a group of shell features and match up the edges precisely. In contrast, sections of a solid can overlap and tolerance becomes less critical. Another advantage of using a solid is that you can use round and chamfer features to define the geometry. If you are modeling a shell, you should try to create solid features and convert the solids to shells when you have finished defining the shape. In addition, if you subsequently want to add additional shell features to a shell part, where the shell part was generated from a solid, you should do the following:

1. Delete the last solid-to-shell feature to convert the model back to a solid.
2. Add your new solid features.
3. Create a new solid-to-shell feature to convert the model back to a shell.

Additional information

- *What is feature-based modeling?*
- *Modifying and manipulating features*
- *Capturing your design and analysis intent*

Capturing your design and analysis intent

If used carefully, the feature-based modeling approach used by Abaqus/CAE allows you to capture both your design and analysis intent.

Design intent is the capability to make changes based on design considerations. For example, when you add a cut feature, you can select either a through cut or a blind cut. If the cut feature represents a bolt hole, you know that the hole must always pass completely through the part. As a consequence, you should select a through cut, and Abaqus/CAE recognizes that the hole remains through even when you change the thickness of the part.

Analysis intent is the capability to make changes based on analysis considerations. Although Abaqus/CAE allows you to create parts with complex, detailed geometry, your final goal is usually a finite element analysis of a meshed representation of the part. Excessive detail, such as fillets and small holes, can lead to regions with a very fine mesh that will, in turn, dominate the time taken by Abaqus/Standard or Abaqus/Explicit to reach a solution. The amount of detail you provide when you create a part in the Part module should be a reflection of your goals. Alternatively, you can create a part with detailed features but suppress them prior to meshing the assembly. For example, if a model takes several hours to analyze, you may wish to simplify it by suppressing features; you could then submit an analysis that runs faster and checks your basic modeling assumptions. If the simplified model behaves as expected, you can unsuppress the features and resubmit a full analysis.

For an example of different feature-based design approaches based on design and analysis intent, consider the cover plate shown in *Figure 1*.

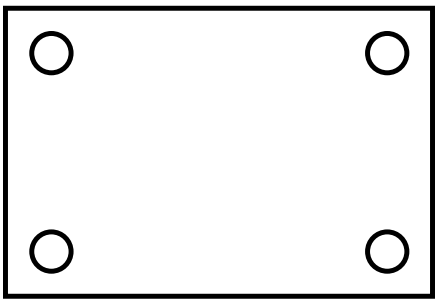


Figure 1: A model of a cover plate.

You could create the three-dimensional shell that models the plate in several ways:

1. Sketch a base feature that includes the four holes.
2. Sketch a rectangular base feature, and add four separate cut features.
3. Sketch a rectangular base feature, and add a single cut feature that cuts all four holes.

Either of the three approaches would generate the same part, but your design intent and your analysis intent govern the best approach. For example:

- Do you want to create and analyze plates of varying sizes with different sized holes for different applications? If the diameter of all four holes is always identical, you should create all four holes as a single cut feature. However, if the diameter of individual holes might differ, you should create four separate cut features.
- Do you want to suppress features before you finalize your design? For example, you could perform a series of analyses with the holes suppressed to determine the desired plate thickness. You could then unsuppress the holes and analyze the finished model. In addition, suppressing features may simplify the mesh that Abaqus/CAE generates, or suppressing features may make the assembly sweep meshable.

  If you want to suppress all four holes in the example of the rectangular cover plate, you should create all four holes as a single cut feature. However, if you want to suppress individual holes, you should create four separate cut

features. If the analysis is straightforward and you do not need to analyze a simplified model, you should sketch a base feature that includes the four holes.

Additional information

- *What is feature-based modeling?*
- *Modifying and manipulating features*
- *Using feature-based modeling effectively*

What is part and assembly locking?

Part and assembly locking is an Abaqus/CAE function that prevents any changes to the features of a part or to the features of the assembly. You can lock parts or the assembly to prevent accidental changes, such as when sharing a model with other Abaqus users or when working on a model that contains many similar parts. You must unlock a part or the assembly if you plan on modifying it.



Note:

Part and assembly locking is not a security feature; any user can unlock and modify parts and assemblies that were locked by another user.

You can click mouse button 3 on a part or on the assembly in the Model Tree and use the menu that appears to lock and unlock the feature. A padlock before the feature name in the Model Tree indicates that a part or the assembly has been locked by the user or by a database upgrade. For more information, see *Using the Model Tree to manage features*.

Alternatively, you can use the **Part Manager** to lock or unlock any part in a model. If the part is unlocked, the **Status** field is empty in the **Part Manager**. If the part is locked, the **Status** field indicates one of two conditions:

Locked (Database upgrade)

Abaqus/CAE locked the part automatically while upgrading the model from a previous release of Abaqus.

Locked

A user locked the part using the Model Tree or the **Part Manager**.

Abaqus/CAE automatically locks the assembly and all the parts in a model when it upgrades a database from an older release of Abaqus. Locking the assembly and the parts allows Abaqus/CAE to complete the upgrade faster than if the assembly and all the parts were also regenerated. If you unlock a part that was locked by a database upgrade, Abaqus/CAE regenerates that part. Similarly, if you unlock an assembly that was locked by a database upgrade, Abaqus/CAE regenerates the assembly.



Warning:

If a part is locked due to a database upgrade, you should unlock the part prior to making any changes to set or property definitions. If you unlock the part after making modifications, your changes can become invalid when Abaqus/CAE regenerates the part.

If you unlock a part that you locked with the Model Tree or the **Part Manager**, Abaqus/CAE does not regenerate the part because it could not be modified while it was locked. Similarly, Abaqus/CAE does not regenerate the assembly when you unlock it after locking it with the Model Tree. If a part that you unlock fails to regenerate, both the locked version and the unlocked version are retained. You can recreate missing features on the unlocked version of the part and use it to replace the locked part throughout the model.

You can instance a locked part and use it in the assembly. In addition, you can add or delete set or property definitions to a locked part or to a locked assembly. However, you must unlock a part or the assembly before you can add features to it or edit existing features.

What are extruding, revolving, and sweeping?

The following sections describe the techniques you can use to extrude, revolve, and sweep a two-dimensional sketch to create a three-dimensional part or feature.

In this section:

- *Defining the extrusion distance*
- *Controlling the direction of an extruded feature*
- *Including twist in an extrusion*
- *Including draft in an extrusion*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *Controlling the direction of a revolved feature*
- *Controlling the cross-section of a revolved feature with pitch*
- *Defining the sweep path and the sweep profile*

Defining the extrusion distance

You can sketch a two-dimensional profile and extrude it to create the following:

- A three-dimensional extruded solid feature.
- A three-dimensional extruded shell feature.
- A three-dimensional extruded cut feature.

Abaqus/CAE provides the following methods for defining the extrusion distance:

Blind

Specify the distance over which Abaqus/CAE extrudes the sketch. The sketch and the distance define the feature and can be edited using the Feature Manipulation toolset. You can use this method when creating extruded solid, shell, and cut features. *Figure 1* illustrates a blind extruded cut in a solid part.

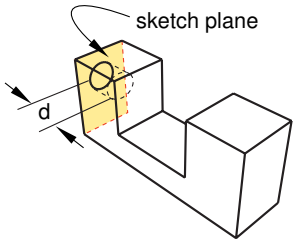


Figure 1: A blind extruded cut.

Up to Face

Select a single face to which Abaqus/CAE extrudes the sketch. The selected face does not have to be parallel to the sketch plane. The selected face can be a nonplanar face; however, it must completely contain the extruded section. If you select this method to define the extrusion distance, only the sketch can be modified using the Feature Manipulation toolset; if you wish to extrude to a different face, you must create a new extruded cut feature. You can use this method when creating extruded solid, shell, and cut features. *Figure 2* illustrates a sketch extruded to a nonplanar face.

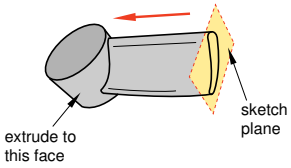


Figure 2: A solid feature extruded up to a nonplanar face.

Through All

This method is available only for extruded cut features. Abaqus/CAE extrudes the sketch defining the profile of the cut completely though the part. If you select this method to define the extrusion distance, only the sketch can be modified using the Feature Manipulation toolset. *Figure 3* illustrates a through all cut in a solid part.

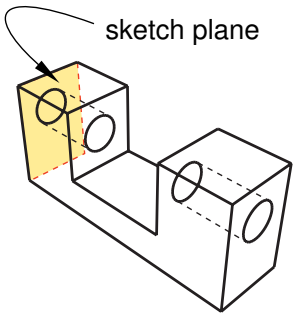


Figure 3: A through all extruded cut.

Controlling the direction of an extruded feature

When you add an extruded feature to a three-dimensional part, Abaqus/CAE chooses a default direction of extrusion from the sketched profile based on the type of feature you are creating. By default, a solid or shell feature is extruded outward such that material is added to the existing base feature. Conversely, a cut feature is extruded inward such that material is removed from the existing base feature.

You can control the direction of an extruded feature as follows:

Choosing the direction while adding an extruded feature

When you complete the sketch to add an extruded feature to an existing part, Abaqus/CAE displays the new sketched profile on the original part. The sketched profile includes an arrow that indicates the extrusion direction. Abaqus/CAE also displays the **Edit Extrusion** dialog box.

You can control the direction of extrusion by clicking  in the **Edit Extrusion** dialog box. The arrow in the viewport changes direction to show the new extrusion direction.

Editing the direction of an existing extruded feature

When you select an extruded feature to edit, Abaqus/CAE highlights the selected feature in the viewport and the **Edit Feature** dialog box appears.

You can reverse the direction of extrusion by toggling **Flip extrude direction** in the **Edit Feature** dialog box. Abaqus/CAE does not display an arrow that indicates the direction of extrusion; however, you can click **Apply** to view your changes. When the direction is acceptable, click **OK** to end the editing process.

You cannot change the direction of extrusion when you are creating a new part because the part would be identical regardless of the direction.

Additional information

- *Adding an extruded solid feature*
- *Adding an extruded shell feature*
- *Creating an extruded cut*

Including twist in an extrusion

You can choose to include twist during the creation of an extrusion. Twist can be used to create twisted cables, helical gears, and other complex shapes that can be formed by passing a constant cross-section through a sequence of parallel planes. Twist modifies an extrusion by rotating the sketched profile about an axis parallel to the direction of extrusion. The center of twist is an isolated point in the sketched profile; it is the point at which the axis used to twist the extrusion passes through the sketch plane. The pitch defines the extrusion distance in which the profile would be twisted by 360°. You can modify the extrusion profile, extrusion direction, center of twist, and pitch using the Feature Manipulation toolset.

You can add twist during the creation of extruded solid, shell, and cut features. *Figure 1* illustrates a twisted extrusion.

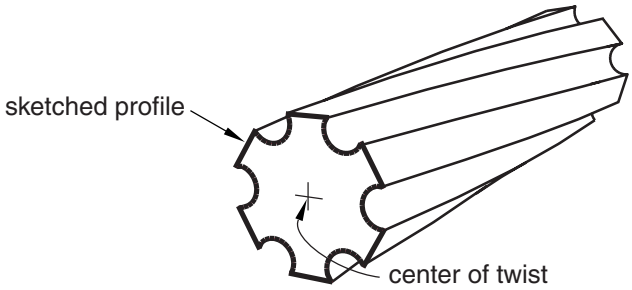


Figure 1: A solid feature extruded with twist.

If you want to create complex shapes in which the sketched profile is revolved rather than extruded, such as screw threads or coil springs, you can include pitch in a revolved solid, shell, or cut feature. See *What types of features can you create?*, for basic information about all the available feature types and *Defining the axis of revolution for axisymmetric parts and for revolved features*, for more information about revolved features.

Additional information

- *Adding an extruded solid feature*
- *Adding a revolved solid feature*
- *Adding an extruded shell feature*
- *Adding a revolved shell feature*
- *Creating an extruded cut*
- *Creating a revolved cut*
- *Meshing complex solids with hexahedral elements*

Including draft in an extrusion

You can choose to create an extrusion with draft. Draft can be used to accurately represent the small angle often applied to ease the removal of cast or molded parts from the tooling. Draft in an extrusion can also be used to create tapered parts.

In a straight extrusion the draft angle is 0°, so all extruded surfaces are perpendicular to the original profile. Draft modifies an extrusion by adjusting the angle between the extruded surfaces and the original sketch plane. Abaqus/CAE reverses the application of draft angle from internal to external features. If external loops in a sketched profile are expanding, internal ones are contracting; this behavior is expected for draft and is required for part removal from tooling (all surfaces taper in the same direction).

You can modify draft, along with the extrusion profile and direction, using the Feature Manipulation toolset. You can add draft during the creation of extruded solid, shell, and cut features. *Figure 1* illustrates an extruded cut with draft in a solid part.

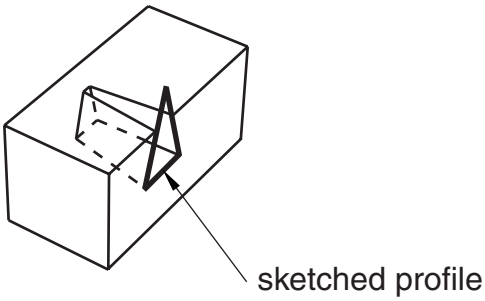


Figure 1: A cut feature extruded with draft.

Note:

The complete sketched profile for the cut in *Figure 1* is a triangle, as shown. If the profile were a trapezoid whose top edge coincided with the edge of the block, the cut would look very different. As the profile was extruded, the application of draft made it smaller. The top face of a trapezoid profile would immediately fall below the surface of the block instead of extending through the top surface.

Abaqus/CAE cannot mesh an extruded solid that includes draft with hexahedral elements unless you partition the solid into structured regions.

Defining the axis of revolution for axisymmetric parts and for revolved features

When you create an axisymmetric part and when you add a revolved feature to a part, the sketch of the profile must include a construction line that defines the axis of rotation. The following rules apply to the sketch and to the construction line:

Creating an axisymmetric part

You can create axisymmetric parts that are defined by either a shell or a wire along with an axis of symmetry by selecting **Part->Create** from the main menu bar. Abaqus/CAE allows you to include a twist degree of freedom in your model when you create an axisymmetric part.

When you sketch the part's base feature, Abaqus/CAE displays a vertical construction line on the *Y*-axis of the sketch representing the axis of symmetry. You must sketch only to the right of the line. Your sketch can touch this line but cannot cross it.

You can add only shell and wire features to an axisymmetric base feature. Abaqus/CAE displays the original sketch and construction line when you add a feature, and the same rules apply—you cannot delete this construction line, and you must sketch only to the right of it.

Creating revolved features

You can create three-dimensional parts with a revolved solid or a revolved shell base feature by selecting **Part->Create** from the main menu bar. Similarly, you can add revolved solids, shells, and cuts to three-dimensional solids and shells by selecting **Shape->Solid->Revolve**, **Shape->Shell->Revolve**, or **Shape->Cut->Revolve** from the main menu bar.

The sketch of a revolved feature must contain a construction line representing the axis of revolution. When you create a new revolved part, Abaqus/CAE creates a vertical construction line through the origin of the Sketcher grid. If desired, you can delete this construction line and redraw it at a new angle and position. In contrast, when you add a revolved feature to an existing part, you must sketch the construction line representing the axis of revolution. You can sketch to the right or to the left of the construction line. Your sketch can touch this line but cannot cross it. If the completed sketch contains more than one construction line, Abaqus/CAE prompts you to select the line that will represent the axis of revolution.

When you are sketching the construction line that represents the axis of revolution, you can position the construction line by selecting a datum axis from the underlying part if one exists. You cannot select the datum axis directly; you must select a point from either end of the datum axis. You can use the datum axis to create concentric features.

When you exit the Sketcher, Abaqus/CAE opens a dialog box to complete the definition of the revolved feature. You enter the angle through which the profile will be revolved and the direction of revolution, and you can choose whether to translate the profile along the axis of revolution by including pitch. You can also specify the direction of translation. The pitch value is the distance through which the profile would be translated during a rotation of 360°. Pitch allows the creation of parts such as coil springs and part details such as screw threads.

If you want to create complex shapes in which the sketched profile is extruded rather than revolved, such as twisted cables or helical gears, you can include twist in an extruded solid, shell, or cut feature. See *What types of features can you create?*, for basic information about all the available feature types and *Defining the extrusion distance*, for more information about extruded features.

Additional information

- *Adding an extruded solid feature*
- *Adding a revolved solid feature*
- *Adding an extruded shell feature*

- *Adding a revolved shell feature*
- *Creating an extruded cut*
- *Creating a revolved cut*

Controlling the direction of a revolved feature

When you create a part with a revolved base feature or add a revolved feature to a three-dimensional part, you can control the direction of revolution. If you include pitch in a revolved feature, you can also control the direction of translation. The descriptions that follow provide the details of controlling both the direction of revolution and, if applicable, the direction of translation for any revolved feature that you create in Abaqus/CAE:

Choosing the direction while creating a revolved feature

When you complete the sketch to create a revolved feature, Abaqus/CAE displays the sketched profile including an arrow that indicates the direction of revolution. If you are creating a part with a revolved base feature, the axis of revolution is also displayed. In addition, Abaqus/CAE displays the **Edit Revolution** dialog box.

If desired, rotate the view until you can discern the arrow direction that indicates the direction of revolution.

You can reverse the direction of revolution by clicking  for **Revolve direction** in the dialog box. The arrow in the viewport changes direction to show the new direction of revolution.

If you select **Include translation** for the revolved feature, a second arrow appears in the viewport to indicate the direction of translation along the axis of revolution. Click  for **Pitch direction** in the dialog box to reverse the direction of translation. Similar to the change for the direction of revolution, the corresponding arrow in the viewport changes direction to show the new translation direction.

Editing the direction of an existing revolved part or feature

When you select a revolved feature to edit, Abaqus/CAE highlights the selected feature in the viewport and the **Edit Feature** dialog box appears.

To reverse the direction of revolution, toggle **Flip revolve direction** in the **Edit Feature** dialog box. Click **Apply** to view your changes.

If the revolved feature you are editing includes pitch, you can edit the direction of translation. Toggle **Flip pitch direction** to reverse the direction of translation. Click **Apply** to view your changes.

Click **OK** to accept the changes.

Additional information

- *Adding a revolved solid feature*
- *Adding a revolved shell feature*
- *Creating a revolved cut*
- *Meshing complex solids with hexahedral elements*

Controlling the cross-section of a revolved feature with pitch

When you create a revolved feature without pitch, the sketched profile is swept about a circular path described by the radius between the axis of revolution and the sketch. The cross-section of the revolved feature is the sketch; the cross-section is both parallel to the axis of revolution and normal to the circular path at all times.

When you include pitch in a revolved feature, the path of the sketch becomes helical instead of circular. You can choose to keep the sketch parallel to the axis of revolution, or you can choose to rotate the sketch normal to the helical path.

To make your sketched profile normal to the helical path of the revolved feature with pitch, toggle on **Sweep sketch normal to path** in the **Edit Revolution** dialog box. When Abaqus/CAE creates the revolved feature, it rotates the sketched profile such that it is normal to the path of revolution at the starting point. The profile remains normal to the path throughout feature creation. Regardless of the pitch value, the cross-section will match the sketched profile. Using this option, you can create coil springs or other features where the cross-section to the path is your sketched profile.

If you do not choose **Sweep sketch normal to path**, the sketched profile remains parallel to the axis of revolution and the cross-section of the revolved feature will vary from the profile. The difference between the profile and the cross-section will increase as you increase the value of pitch. For example, if there is no pitch, a circular sketched profile creates a circular cross-section. If you increase the pitch, the cross-section will become increasingly elliptical. You can create screw threads or other features where the cross-section parallel to the axis of revolution is your sketched profile.

To change the cross-section behavior after the feature is created, you can toggle **Move sketch normal to path** in the **Edit Feature** dialog box.

Additional information

- *Adding a revolved solid feature*
- *Adding a revolved shell feature*
- *Creating a revolved cut*
- *Meshing complex solids with hexahedral elements*

Defining the sweep path and the sweep profile

To create a swept feature, select **Shape->Solid->Sweep**, **Shape->Shell->Sweep**, or **Shape->Cut->Sweep** from the main menu bar or select the equivalent tool from the Part module toolbox. Abaqus/CAE displays the **Create Solid Sweep**, **Create Shell Sweep**, or **Create Cut Sweep** dialog box.

Sweeping is a two-part operation: first you define the sweep path, and then you define the sweep profile. The profile is swept along the length of the path to form a three-dimensional solid, shell, or cut feature. The sweep path can be any continuous path you can create with the Sketcher or any series of connected edges or wires in your part. The latter option allows you to define a three-dimensional sweep path, such as a point-to-point spline wire; the sketch method provides greater flexibility but supports only two-dimensional paths. *Figure 1* shows an example of a sweep path and a sweep profile.

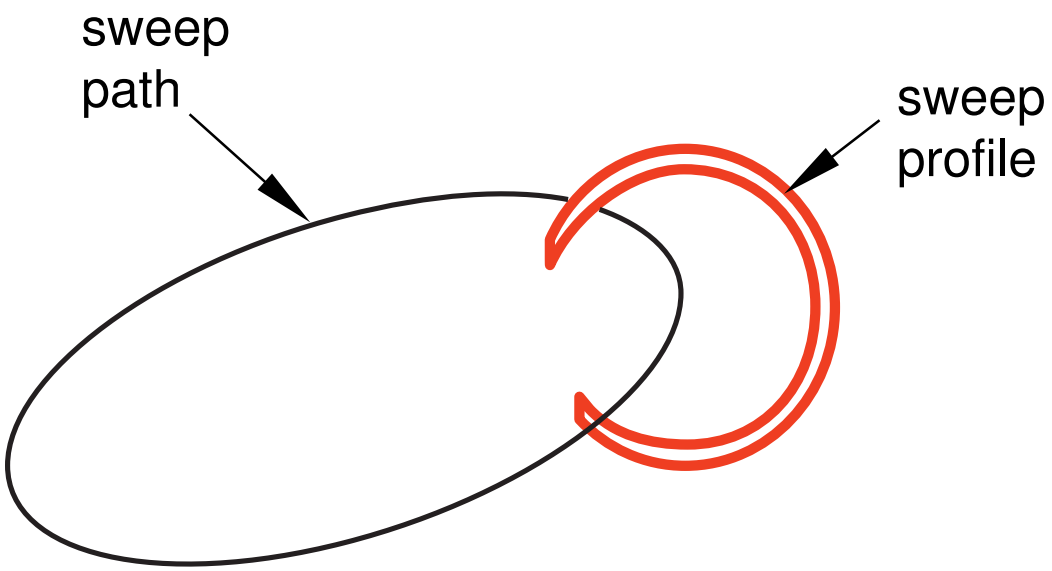


Figure 1: An example of a sweep path and profile.

The feature created by sweeping the sweep profile along the above path is shown in *Figure 2*.

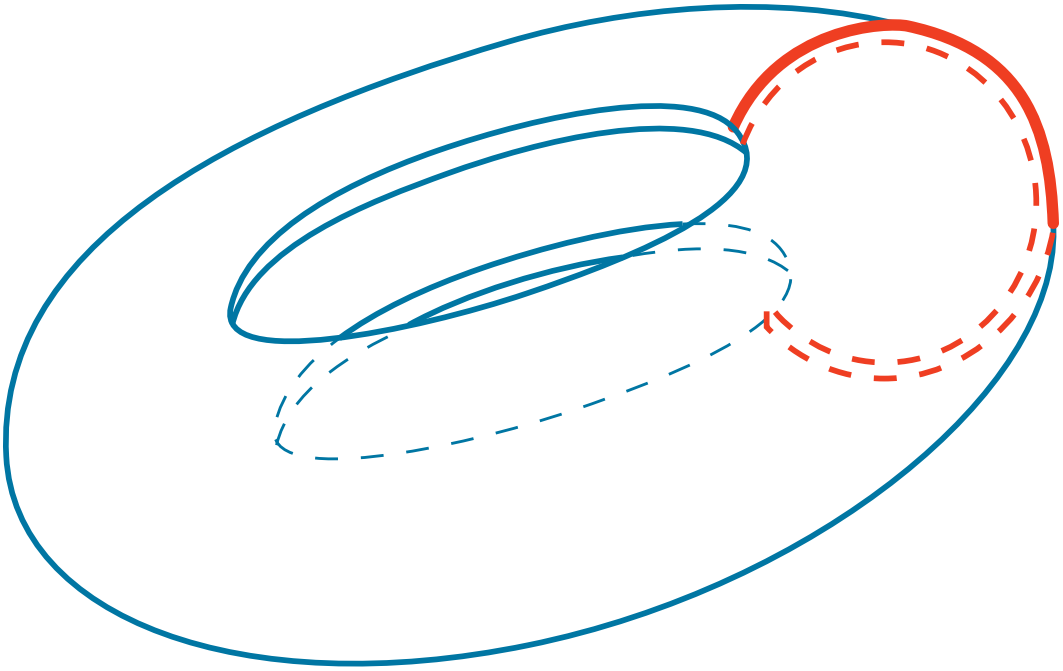


Figure 2: The resulting swept feature.

The sweep profile can be defined either in the Sketcher or by selecting components in the geometry. For solid or cut swept features you can select one of the faces in your part to use as the sweep profile; for shell swept features you can select one or more edges in your part to use as the sweep profile.

If you define your sweep path or sweep profile using the Sketcher, you can modify that feature using the Feature Manipulation toolset. The sweep tools are available only when you are working on a deformable or discrete part that you created in a three-dimensional modeling space.

You can define a swept solid, swept shell, or swept cut feature whose sweep profile is offset from the sweep path. In this case Abaqus/CAE moves the sweep path to a parallel location that passes through the sweep profile and creates the swept feature at that location.

You can control whether the orientation of the sweep profile changes as it travels along the sweep path. Applying a draft to a sweep feature works best when the sweep path is linear. If you toggle on **Keep profile normal constant**, Abaqus/CAE does not change the sweep profile orientation and the profile at the beginning of the sweep path will be parallel to the profile at the end of the sweep path. If you toggle off this option, Abaqus/CAE adjusts the orientation of the sweep profile so that the angle between the sweep path and the profile normal remains constant as the profile travels down the sweep path. The draft option and **Keep profile normal constant** option are mutually exclusive; Abaqus/CAE toggles off one of these options if you select the other.

The sweep profile must be closed when you are creating a swept solid or cut feature. However, unlike the sweep profile, the sweep path can be open or closed regardless of whether you are creating a swept solid, shell, or cut feature. If the sweep path is closed, the two ends of the path must meet tangentially. For example, the closed sweep paths labeled “Bad” in *Figure 3* are not allowed because the ends of the path meet at an angle.

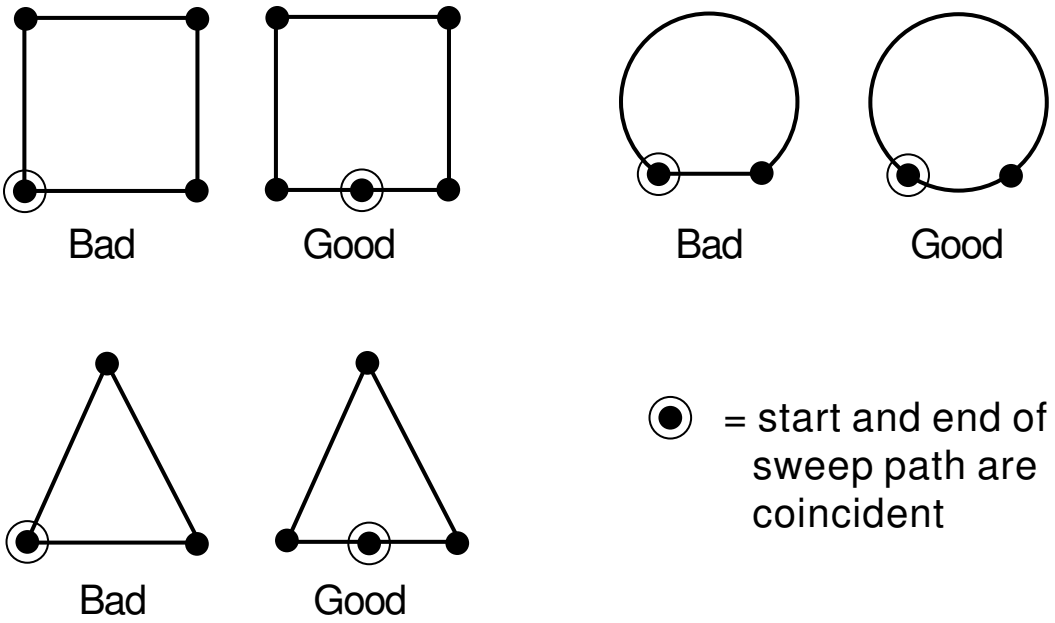


Figure 3: Valid and invalid sweep paths.

As you define your sweep feature, you can apply a twist or draft. For more information about these tools, see *What types of features can you create?*. You can also toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the swept solid feature and the existing part. The internal boundaries can create regions that can be structured or swept meshed without having to resort to partitioning.

Additional information

- *What is feature-based modeling?*
- *Adding a point-to-point wire feature*

What is lofting?

Lofting is a method that allows you to create complex three-dimensional features that cannot be created by extruding, revolving, or sweeping.
For example, you can use lofts to model an exhaust manifold that would be difficult to create by other means due to varying cross-sections. You can create solid, shell, or cut loft features in Abaqus/CAE. A loft feature transforms from a starting section shape and orientation to an ending shape and orientation. You first create sections that define the shape of the loft as it passes through an area in space. Then Abaqus/CAE can create the path between sections automatically, or you can define one or more continuous paths connecting one point on each loft section to a corresponding point on the next section. You can also choose from several tangency options to control the shape of the loft as it leaves the starting section or as it approaches the ending section. This section describes the options available for defining the loft sections, loft paths, and loft tangencies prior to creating a loft feature and explains self-intersection.

In this section:

- *Defining the loft sections*
- *Defining a loft path*
- *Defining loft tangency*
- *Self-intersection checks*

Defining the loft sections

Loft sections represent the shape that the loft feature will have at a particular point along a loft path. At least two sections are required to create a loft feature. You can create additional sections to control the shape of the loft between the starting and ending sections. In a solid or cut loft, each loft section must be a closed loop with no branches. In a shell loft, the loft sections can either all be open or all be closed. You can define planar or nonplanar loft sections to create a loft feature. Once the loft is created, the number of sections and their order within the loft cannot be changed.

You create loft sections by picking from existing edges on the part in the current viewport. Any edges can be selected; for example:

- Edges that define extruded, revolved, or swept features.
- Edges that define planar wire or shell features.
- Spline wire features.

You can use individual edges from several features to define a single section. However, planar wires sketched on datum planes are one of the simplest means you can use to define the loft sections. Using the simplest means to define your loft sections will give you more control and will result in a more robust loft feature.

You cannot modify loft sections directly. Once you create the loft feature, you can use the Feature Manipulation toolset to edit features that created the edges used in the loft sections. Moving a vertex or edge that is used in a loft section will change the shape of the section and the shape of the corresponding loft feature.

Defining a loft path

Each loft feature that you create requires at least one loft path. When you have defined the loft sections, you can choose to modify the loft path or paths. The paths of a loft feature connect a point on the starting section to a point on the ending section. If more than two loft sections are defined, each path also passes through a point on each intermediate section. You can define a loft path using the options on the **Transition** tabbed page of the **Edit Loft** dialog box.

When you create a loft feature, you can choose from the following methods to define a loft path:

Specify tangencies

Specify tangencies is the default loft path definition. If you select **Specify tangencies**, Abaqus/CAE creates a single smooth path that passes through the center of each loft section, as shown in *Figure 1*. You can apply tangency conditions that modify the shape of the loft near the starting and ending loft sections. For more information on loft tangency, see *Defining loft tangency*.

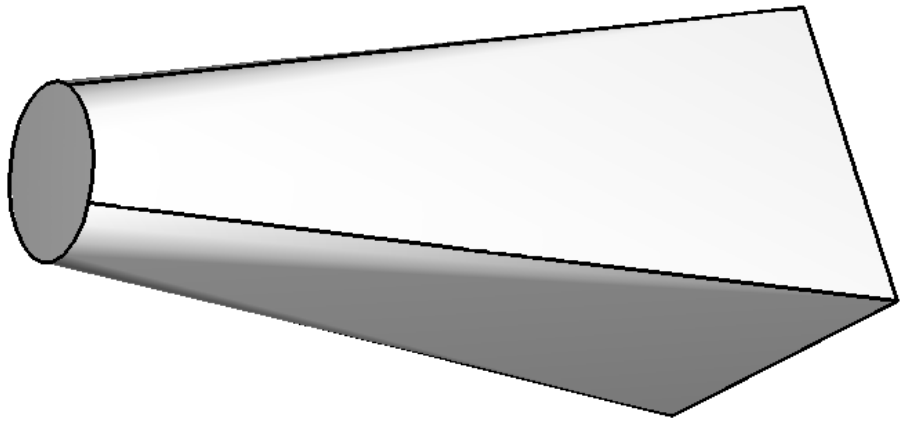


Figure 1: A loft feature with a path defined by Abaqus/CAE.

Select path

If you choose **Select path**, you can select from existing edges to define a loft path. This method also allows you to define multiple loft paths. The loft feature is created by following the loft paths as they connect one loft section to the next, as shown in *Figure 2*. A loft feature with a single selected path behaves similarly to a swept feature except that the cross-section of the loft is constantly changing to match the position and shape of the next loft section along the path.

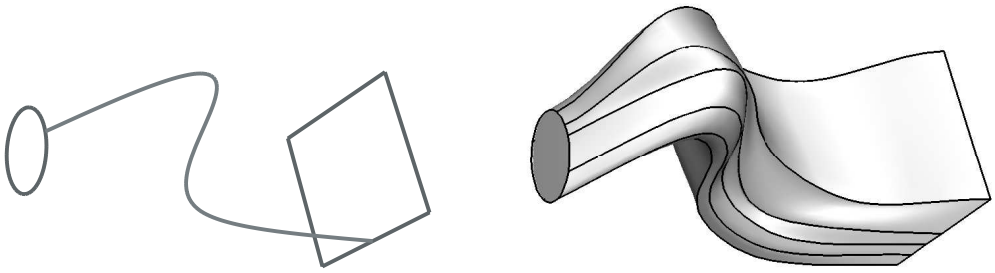


Figure 2: A loft feature with a single user-defined path.

You must pick from existing line segments in the viewport to create paths connecting all of the loft sections. Each path must be a smooth curve, and it must connect the sections in the same order that they will be connected when the loft is created. You can use the tool to create spline wires that define the three-dimensional paths. For more information, see *Adding a point-to-point wire feature*.

Once the loft feature is created, you cannot edit the paths directly, regardless of which path definition you chose. However, if you used **Select path**, you can edit the points that created each spline wire by using the Feature Manipulation toolset to edit the features that created the wire vertices.

Defining loft tangency

If you accept the default **Specify tangencies** method for a loft, you can choose from several loft tangency options. Loft tangency affects the angle at which the loft faces leave the first loft section and approach the last section. The effect of tangency settings is transient, diminishing in proportion to the distance from the start or end section. The shape of the loft feature between any intermediate sections is unaffected by the loft tangency.

You can set all of the loft tangency options except **None** independently for the starting and ending section. For example, you can set the start tangency to **Normal** and the end tangency to **Radial**. You can choose from the following options to define the loft tangency:

None

None is the default setting, and it is the only tangency setting that can be used with nonplanar sections. If you choose **None**, you must use it for both the start and the end tangency. **None** applies no conditions to the shape or direction of the loft. The edges of the loft feature will make a linear approach from the starting section to the second section and from the next-to-last section to the last section as shown in *Figure 1*.

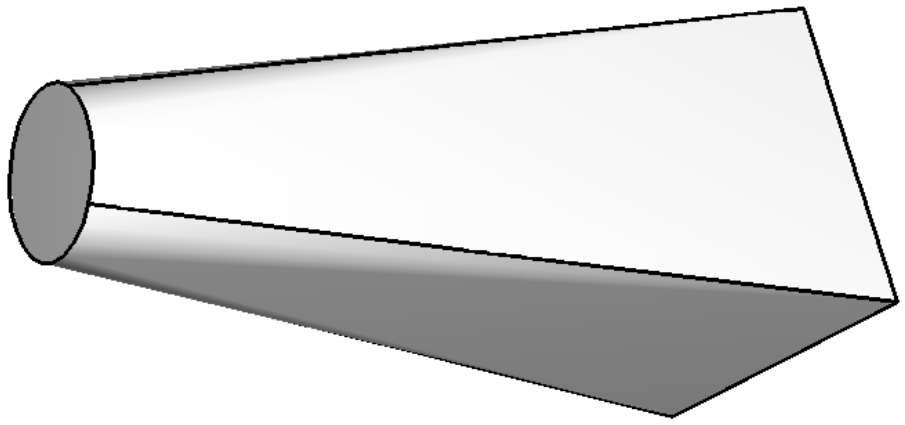


Figure 1: A loft feature with no tangency.

Normal

The **Normal** setting forces the faces created by the lofted edges to be at 90° to the first loft section as they are initially lofted toward the second section. Similarly, this setting forces the faces to be at 90° as they approach the last section of the loft feature. If you set **Start Tangency** to **Normal**, the initial part of the lofted feature will be similar to a straight extrusion as shown in *Figure 2*.

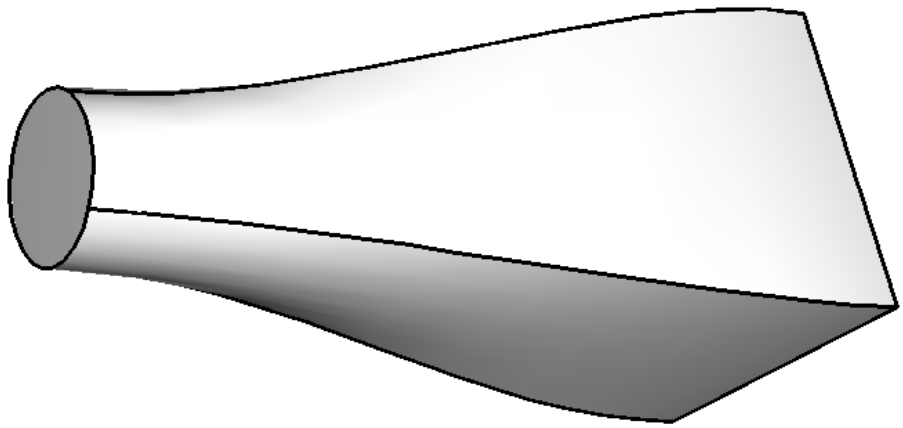


Figure 2: A loft feature with normal tangency at both ends.

Radial

The **Radial** setting forces the faces created by the lofted edges to be at 0° to the first loft section as they are initially lofted toward the second section. Similarly, this setting forces the faces to be at 0° as they approach the last section of the loft feature. Thus, the faces initially radiate outward from the starting loft section or inward toward the ending loft section. If you set **Start Tangency** to **Radial**, the initial part of the lofted feature will be similar to an extrusion with a draft angle approaching 90° as shown in *Figure 3*.

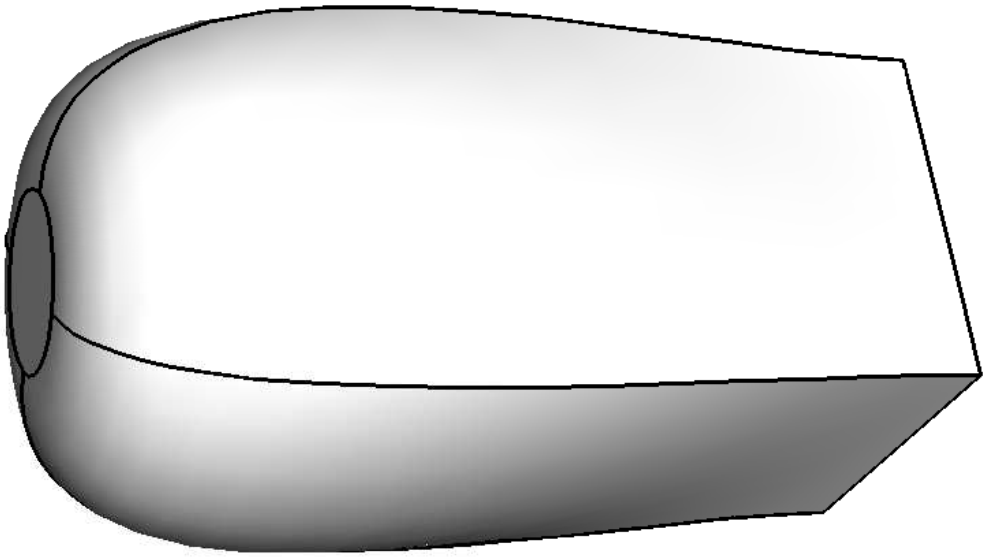


Figure 3: A loft feature with radial tangency at the left end and normal tangency at the right end.

Warning:

If you attempt to create a loft feature with only two loft sections and a dissimilar number of vertices, setting both **Start Tangency** and **End Tangency** to **Radial** may cause the loft feature to fail.

Specify

The **Specify** setting allows you to control both the **Angle** applied to the lofted edges and the **Magnitude %** that represents a relative distance through which the angle will affect the loft. If you set **Start Tangency** to **Specify**, the initial part of the lofted feature will be similar to an extrusion with a draft angle of **Angle** degrees as shown in *Figure 4*.

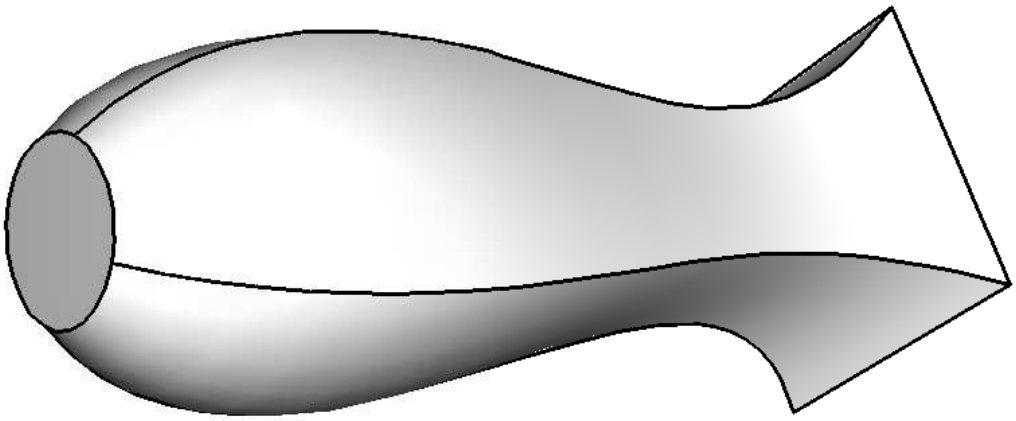


Figure 4: A loft feature with specified tangency at both ends.

Figure 4 shows a **Start Tangency** angle of 45° (left) and an **End Tangency** angle of 135° (right), both applied with magnitudes of 25%. For reference, a **Normal** tangency setting corresponds to specifying an angle of 90° and a magnitude of 25% and a **Radial** tangency setting corresponds to specifying an angle of 0° and a magnitude of 25%.

The angle of the loft faces at any point depends on the **Angle** and **Magnitude %** settings, the distance between consecutive loft sections, and the severity of change between the loft sections. Depending on these conditions, the requirement to make a smooth transition from one loft section to the next may override some loft tangency effects. If you require greater control over the shape of the loft, use the **Select path** method to define paths that the loft feature will follow exactly.

Self-intersection checks

Due to the complexity of features that you can create by lofting, a set of tests is available to ensure that the geometry will be valid for analysis. You can define loft sections and paths such that the loft feature would intersect itself. A loft feature with self-intersections would be impractical as a manufactured part and would also be difficult or impossible to mesh and analyze.

When you toggle on **Perform self-intersection checks** in the **Feature Options** dialog box, Abaqus/CAE tests for self-intersection while it is creating the loft feature. If any faces of the loft intersect other faces, Abaqus/CAE displays an error message stating that there are invalid intersections and does not create the loft feature. The time required to complete the tests varies with the complexity of the loft you are attempting to create. For example, if the shape of your loft varies greatly from section to section or if you have defined a complex loft path, the tests will significantly increase the time required to create the loft feature. If you choose not to include the tests, Abaqus/CAE will create the loft feature regardless of whether the geometry is valid.

Additional information

- *What is feature-based modeling?*
- *Adding a solid loft feature*
- *Adding a shell loft feature*
- *Creating a loft cut*
- *What are self-intersection checks?*

Using the Sketcher in conjunction with the Part module

Sketches are two-dimensional profiles that form the geometry of the features defining an Abaqus/CAE native part. You use the Sketcher to create these sketches; in the Part module you use them directly to define a planar part or a beam, or you extrude, sweep, or revolve them to form a three-dimensional or axisymmetric part. Whenever you need to create the base feature of a new part, add a feature to a part, or modify an existing feature, the Part module automatically enters the Sketcher and you operate on the sketch that forms the two-dimensional profile of the feature. When you have finished sketching, Abaqus/CAE automatically returns you to the Part module.

If you are adding a feature or modifying an existing feature, you must choose the plane on which to sketch. For a detailed description of how Abaqus/CAE determines the orientation of the part relative to the sketch plane, see *How Abaqus/CAE orients your sketch*.

Additional information

• *The Sketch module*

Understanding toolsets in the Part module

The Part module provides a set of toolsets that allow you to add and modify the features that define a part. This section describes how these toolsets are used within the Part module.

For more detailed information about each toolset, refer to:

- *The Datum toolset*
- *The Edit Mesh toolset*
- *The Feature Manipulation toolset*
- *The Filter toolset*
- *The Partition toolset*
- *The Query toolset*
- *The Reference Point toolset*
- *The Geometry Edit toolset*
- *The Set and Surface toolsets*

The Display Group toolset is discussed in *Using display groups to display subsets of your model*.

In this section:

- *Using the Datum toolset in the Part module*
- *Using the Feature Manipulation toolset in the Part module*
- *Using the Partition toolset in the Part module*
- *Using the Query toolset in the Part module*
- *Using the Reference Point toolset in the Part module*
- *Using the Geometry Edit toolset in the Part module*
- *Using the Set toolset in the Part module*

Using the Datum toolset in the Part module

A datum can be thought of as reference geometry or a construction aid that helps you create a feature when the part does not contain the necessary geometry; you create datum geometry using the Datum toolset. A datum is a feature of a part and is regenerated along with the rest of the part. Furthermore, datum geometry is visible unless you toggle it off by selecting **View->Part Display Options->Datum** from the main menu bar. A datum created in the Part module appears with each instance of the part in the Assembly module or any other assembly-based module.

Datum points are projected onto the Sketch plane in the Sketcher, and the projected point can be selected. However, you cannot refer to datum axes or planes in the Sketcher. Examples of how you might use datum planes and axes in the Part module are given below.

Datum plane

You can sketch directly on datum planes, and any features you sketch on a datum plane will be projected onto the part. Projecting a sketch from a datum plane is useful if the part does not already contain a convenient sketch plane.

For example, suppose you want to cut a hole straight through the three-dimensional triangular part shown in *Figure 1* parallel to the *X*-axis.

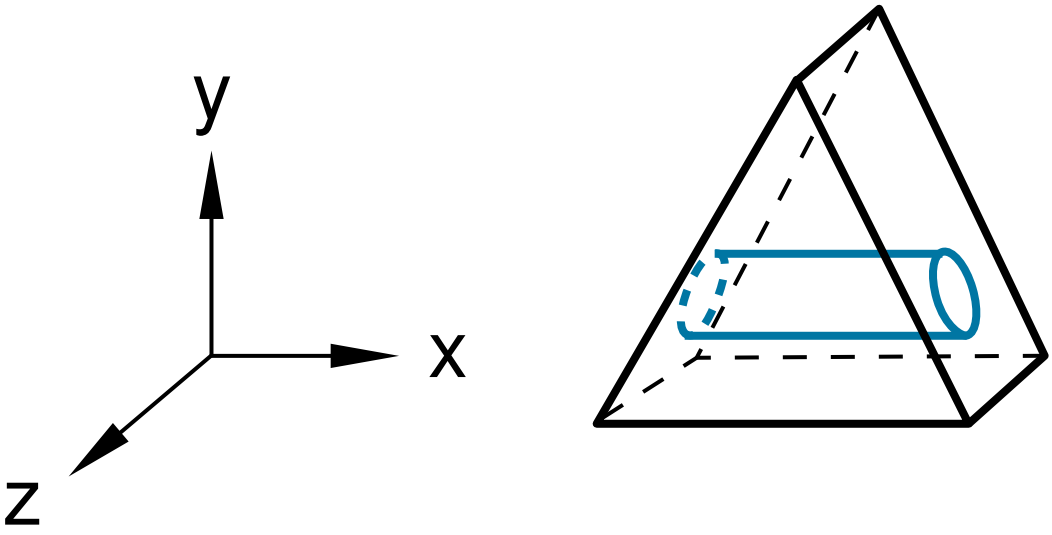


Figure 1: The desired cut feature.

The part does not already have a face that is suitable for sketching the profile of the hole; sketching the profile directly on a face results in a hole normal to the face, as shown in *Figure 2*.

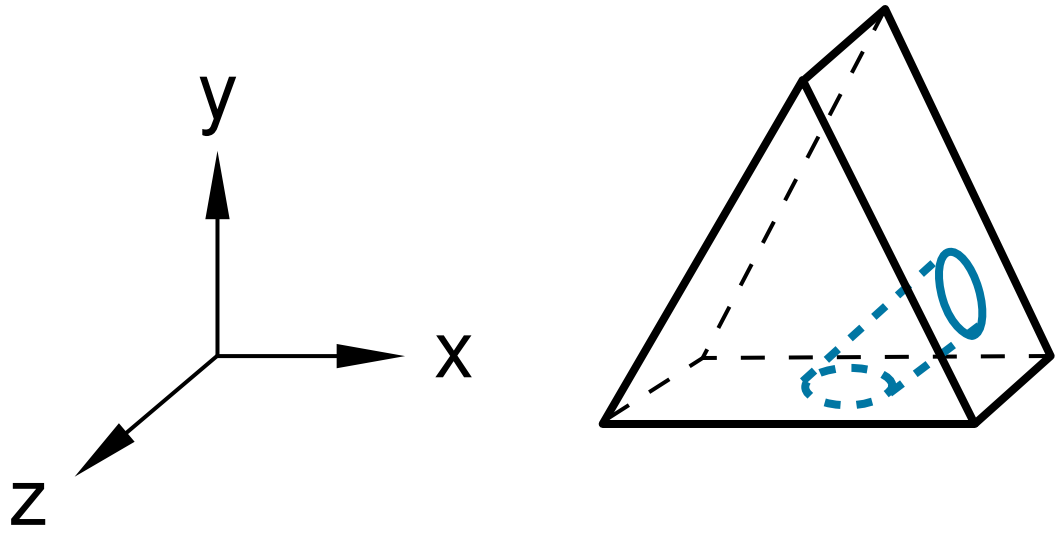


Figure 2: A cut normal to the face.

To cut the desired hole, first use the Datum toolset to create a datum plane on the *Y–Z* principal plane, as shown in *Figure 3*.

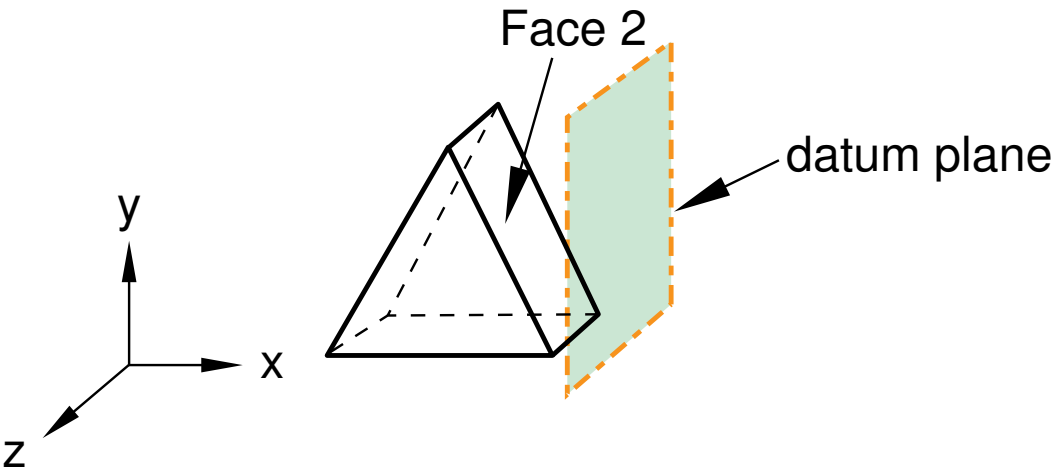


Figure 3: A datum plane.

Second, sketch the profile of the cut on the new datum plane, as shown in *Figure 4*.

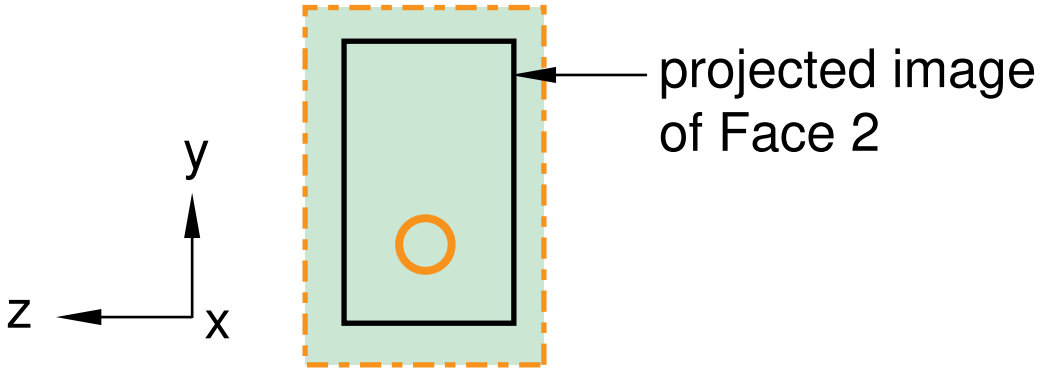


Figure 4: A sketch on the datum plane.

When you exit the Sketcher, Abaqus/CAE cuts the sketched hole through the part, perpendicular to the datum plane and parallel to the *X*-axis. This cut is illustrated in *Figure 5*.

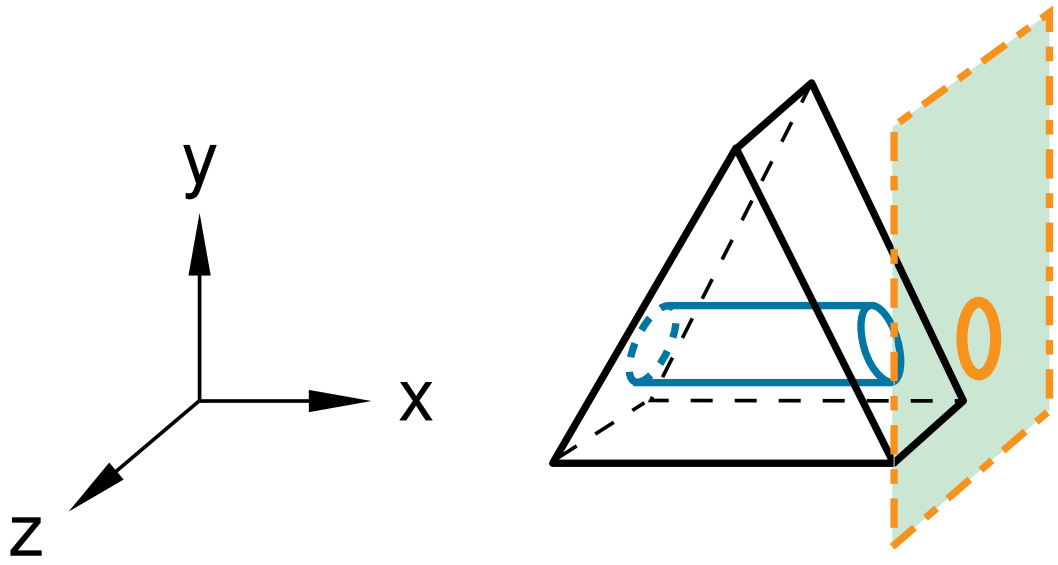


Figure 5: The desired cut.

Datum axis

You can use the Datum toolset to create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid when adding or modifying a feature to a three-dimensional solid. Creating a datum axis is useful when the part does not already contain the necessary axis.

For example, suppose you want to cut a slot through the part as shown in *Figure 6*.

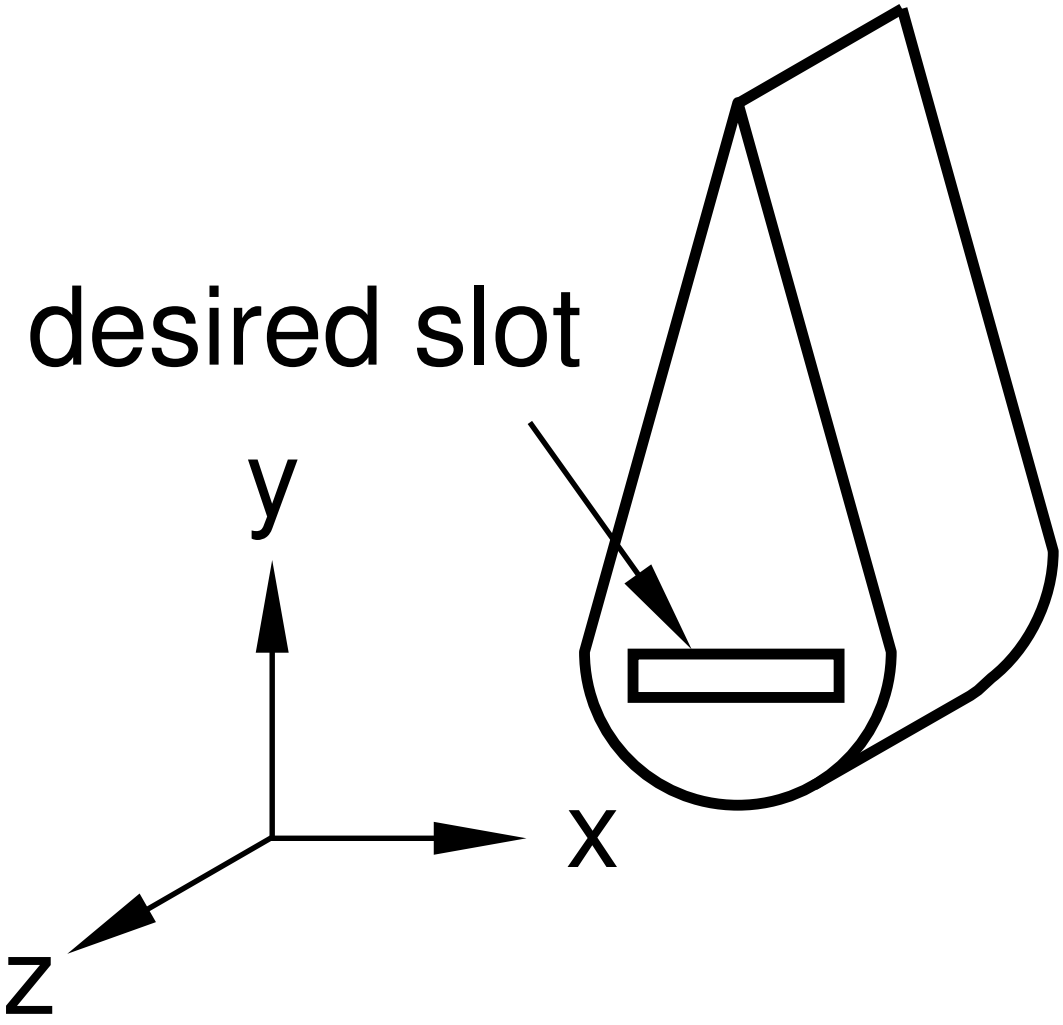


Figure 6: The desired slot.

Sketching the slot is difficult because selecting either of the two straight edges of the part as the sketch's vertical axis causes the sketch gridlines to align with the line you select, not with the *X*- or *Y*-axis. To make it easier to create the slot with the desired orientation, first use the Datum toolset to create a datum axis along the *Y*-axis, as shown in *Figure 7*.

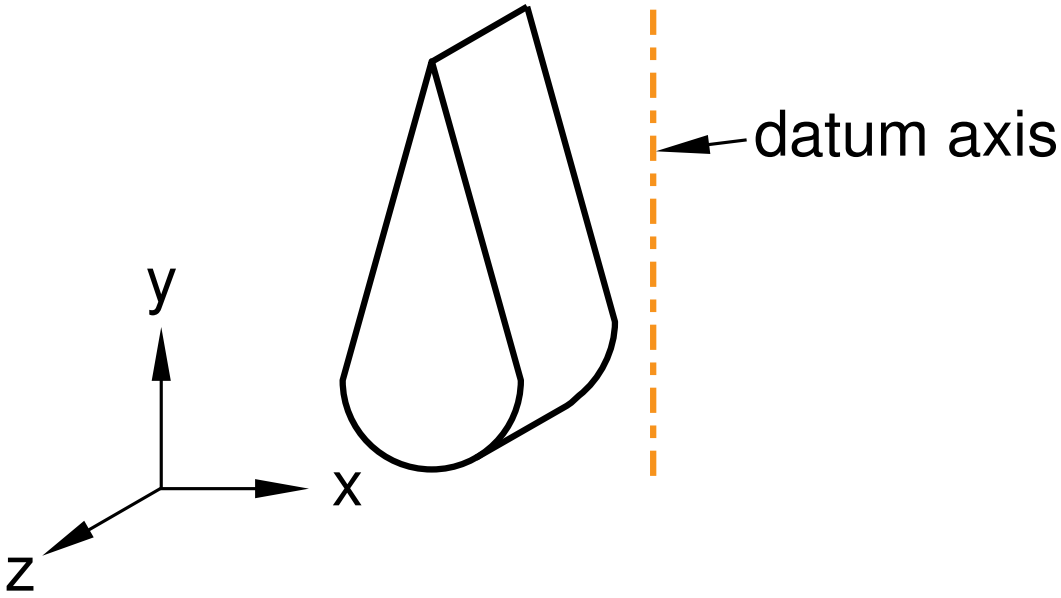


Figure 7: The datum axis.

When you select the datum axis to appear vertical and on the right, the Sketcher starts, and its grid is aligned with the part's *X*- and *Y*-axes, as shown in *Figure 8*.

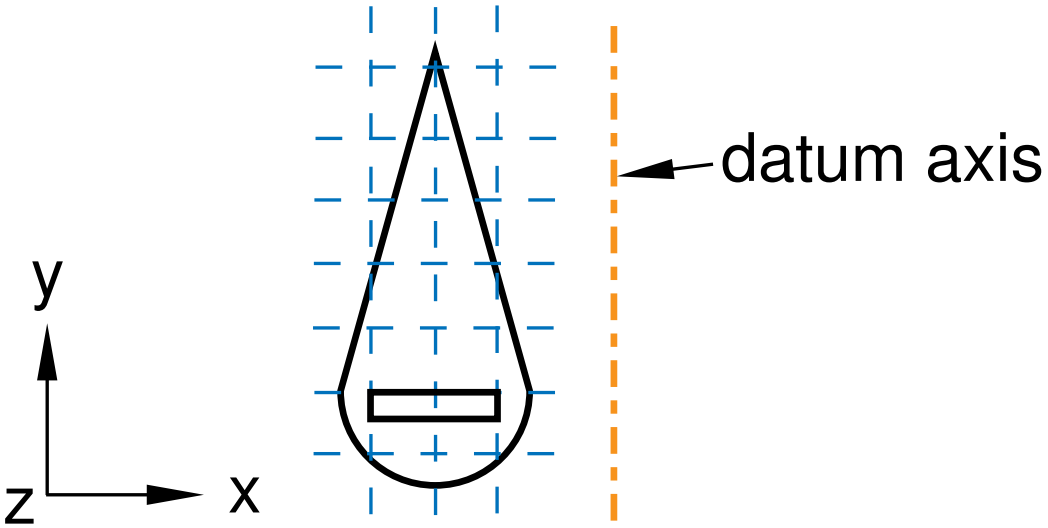


Figure 8: The resulting sketch orientation.

Additional information

- *Understanding toolsets in the Part module*
- *The Datum toolset*

Using the Feature Manipulation toolset in the Part module

The following are considered to be features of a part:

- Geometric features, such as extruded solids, revolved shells, sketched wires, and rounded edges
- Repair operations
- Partitions
- Datum geometry

When the Feature Manipulation toolset asks you to select a feature, you can select it from the viewport. Alternatively, you can select the feature from the Model Tree.

If you click mouse button 3 on a feature in the Model Tree, the menu that appears allows you to do the following:

Edit

When you edit a feature, Abaqus/CAE displays the feature editor. You can either modify the feature's parameters directly or, if applicable, you can modify the sketch that forms the two-dimensional profile or sweep path of a feature.

Regenerate

When you modify features in a complex part, it may be convenient to postpone regeneration until you make all your changes, since regeneration can be time consuming. Select **Feature->Regenerate** when you are ready to regenerate the part.

Rename

Rename a part.

Suppress

Suppressing a feature temporarily removes it from the definition of the part. A suppressed feature is invisible, cannot be meshed, and is not included in the analysis of the model. You cannot suppress the base feature, and suppressing a parent feature will suppress all of its child features.

Resume

Resuming a feature restores a suppressed feature to the part; resuming a parent feature restores all of its child features. You can choose to resume all features, the set of features most recently suppressed, or a selected feature.

Delete

Deleting a feature removes it from the part. You cannot resume a deleted feature.

Query

When you query a part, Abaqus/CAE displays information in the message area and writes the same information to the replay file (*abaqus*`.rpy`) in the form of comments.

Options

The **Feature Options** dialog box allows you to tune the regeneration performance of the current model.

Additional information

- *Understanding toolsets in the Part module*
- *The Feature Manipulation toolset*

Using the Partition toolset in the Part module

Within the Part module, you can use the Partition toolset to partition a part into additional regions. After you partition a part, you can use the Property module to assign different sections to the resulting regions; for example, you might use partitions to delineate regions of the part that are comprised of different materials.

The partitions you create are features associated with the part, so that each instance of that part in the assembly will contain all the partitions created in the Part module. You can use the regions when working with the assembly in other modules; for example, you can apply a load over a region in the Load module. If you do not want to associate the partitions with every instance of the part, create an independent instance in the Assembly module and partition the independent instance. For more information, see *Partitioning the assembly*.

If you have created the assembly and applied attributes to it, such as loads, boundary conditions, and sets, you should be careful if you subsequently decide to return to the Part module and partition one of the original parts. The region to which the attribute is assigned may change unexpectedly if the region is affected by the partition. In general, you should try to finish creating your parts in the Part module before you start creating part instances and applying sets, loads, and meshes to the assembly. If you do return to the Part module to create a partition, you should at least check that the regions in the assembly to which attributes are assigned are still valid.

Additional information

- *Understanding toolsets in the Part module*
- *The Partition toolset*

Using the Query toolset in the Part module

You can use the Query toolset to request either general information or module-specific information.

For a discussion of the information displayed by general queries, see *Obtaining general information about the model*.

Select **Tools->Query** from the main menu bar, or click the  tool in the Query toolset to start the Query toolset.

The following queries are specific to the Part module:

Part attributes

Abaqus/CAE displays the part name, modeling space, and type in the message area along with the shape (solid, shell, wire, or point) and the number of entities (cells, faces, edges, and vertices).

Regeneration warnings

If any sets or surfaces in the selected part cannot be regenerated because their underlying geometry has been modified or deleted, Abaqus/CAE displays the set or surface name, the original number of faces, and the number of faces found during the query.

Substructure statistics

Abaqus/CAE displays the following information about the selected substructure part: the number of retained nodes, eigenmodes, and substructure loads in the part; the availability of the recovery matrix, gravity load vectors, reduced mass matrix, reduced structural damping matrix, and reduced viscous damping matrix in the substructure; and mass properties of the substructure.

Additional information

- *Understanding toolsets in the Part module*
- *The Query toolset*

Using the Reference Point toolset in the Part module

From the main menu bar, select **Tools->Reference Point** to create a reference point on a part. A part can include only one reference point. For more information, see *The reference point and point parts*, and *The Reference Point toolset*.

Abaqus/CAE displays the reference point at the desired location and labels it **RP**. You can change the reference point label by clicking mouse button 3 on the feature in the Model Tree and selecting **Rename** from the menu that appears. If desired, you can turn off the display of the reference point symbol and the reference point label; for more information, see *Controlling reference point display*.

Using the Geometry Edit toolset in the Part module

You use the Geometry Edit toolset to repair regions of a part that contain invalid or imprecise geometry. For more information, see *Editing techniques*. You can use the Query toolset to locate regions that need repairing. For more information, see *The Query toolset*.

Additional information

- *Understanding toolsets in the Part module*
- *The Geometry Edit toolset*

Using the Set toolset in the Part module

Sets created by selecting geometry from a part are called part sets, and you use the Set toolset to create and manage part sets. You can use part sets in the Part module to select regions that should be repaired by the Geometry Edit toolset. In addition, in the Property module you can assign sections to regions specified by part sets. When Abaqus/CAE prompts you to select a region, you can either select the region from the part in the current viewport, or you can select a named part set.

Only part sets are visible in the **Set Manager** in the Part module. When you instance a part in the Assembly module, Abaqus creates part instance sets that refer to any part sets that you previously created. For more information, see *Understanding sets and surfaces*, and *Using sets and surfaces in the Assembly module*.

Additional information

- *Understanding toolsets in the Part module*
- *The Set and Surface toolsets*

Using the Part module toolbox

You can access all the Part module tools through either the main menu bar or the Part module toolbox. *Figure 1* shows the hidden icons for all the part tools in the Part module toolbox.

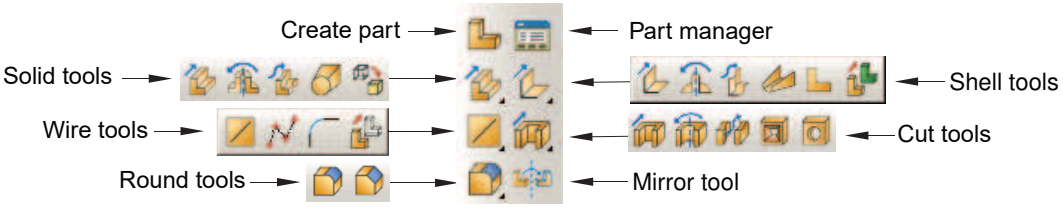


Figure 1: The Part module toolbox.

For information on using each of the Part module tools, refer to the following sections:

- *Managing parts*
- *Using the Create Part dialog box*
- *Adding a feature to a part*
- *Adding a solid feature*
- *Adding a shell feature*
- *Adding a wire feature*
- *Adding a cut feature*
- *Blending edges*

Managing parts

This section describes how you manage the parts in your model while working in the Part module.

In this section:

- *Managing parts*
- *Creating a new part*
- *Copying a part*

Managing parts

To create, copy, rename, and delete parts, use one of the following:

- The **Create**, **Copy**, **Rename**, and **Delete** items listed under the **Part** menu on the main menu bar.

  The **Copy**, **Rename**, and **Delete** items contain submenus listing all the parts in the current model.
- The **Part Manager** dialog box. The **Part Manager** dialog box contains functions similar to those listed under the **Part** menu on the main menu bar, but with a convenient browser that lists the names of all the parts available within the current model along with their modeling space (three-dimensional, two-dimensional, or axisymmetric), type (deformable, discrete rigid, analytical rigid, or Eulerian), and status (locked, invalid, or invalid (ignored)).

  To display the **Part Manager** dialog box, select **Part->Manager** from the main menu bar.

To lock or unlock parts for editing in the current model database, use the **Lock** and **Unlock** buttons in the **Part Manager** dialog box. The **Status** field indicates **Locked** parts . For more information on part locking, see *What is part and assembly locking?*.

To update the validity or to ignore the invalidity of parts, use the **Update Validity** and **Ignore Invalidity** buttons in the **Part Manager** dialog box. The **Status** field indicates **Invalid** or **Invalid (ignored)** parts. For more information on part validity, see *Working with invalid parts*.

To retrieve a part from the model database and display it in the current viewport, select the part from the **Part** list located in the context bar. The **Part** list contains all the parts in the current model.

Additional information

- *Managing parts*
- *Managing objects*

Creating a new part

Select **Part->Create** from the main menu bar to create a new part in the current viewport.

A model can contain multiple parts; each part exists in a local coordinate system, and you use the Assembly module to create instances of the parts and position those instances relative to each other in a global coordinate system. When you create a part, you name the part and select its type, modeling space, base feature, and approximate size; you then sketch the profile of the part's base feature. The part name should not be the same as the model name.

1. From the main menu bar, select **Part->Create**.

   The **Create Part** dialog box appears. For more information, see *Using the Create Part dialog box*.

   Tip: You can also create a part using the tool in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Type a name for the part. For information on naming Abaqus/CAE objects, see *Using basic dialog box components*. You can rename a part after you create it.
3. Choose the new part's modeling space, type, base feature, and approximate size. For more information, see *How is a part defined in Abaqus/CAE?*.

   Note:

   To change a part's modeling space or type you must use the Model Tree to edit the part. For more information, see *How is a part defined in Abaqus/CAE?*.

4. Click **Continue** to close the **Create Part** dialog box.

   The Sketcher starts, and the **Sketch** grid appears in the current viewport.

   If you are creating a three-dimensional revolved solid or shell, Abaqus/CAE displays a vertical construction line on the *Y*-axis of the sketch that serves as the axis of revolution. You can sketch on either side of this construction line, but the sketch must not cross the construction line.

   If you are creating an axisymmetric part, Abaqus/CAE displays a vertical construction line on the left side of the sketch that serves as the axis of revolution. You must sketch to the right of this construction line.

5. Use the Sketcher to sketch the two-dimensional profile of the base feature. For more information, see *The Sketch module*."

   If you are constructing a swept part, you must first sketch the sweep path and exit the Sketcher. The **Sketch** will then restart automatically, and you can sketch the profile to be swept.

6. When you have finished sketching the base feature, click mouse button 2 to exit the current **Sketch** tool.
7. In the prompt area, click **Done** to exit the Sketcher. If the base feature is a three-dimensional solid or shell extrusion, the **Edit Base Extrusion** dialog box appears. You must use the **Depth** field to enter the distance through which to extrude the profile. You can also select **Twist** or **Draft** to modify the form of extrusion that Abaqus/CAE will create. If the base feature is a three-dimensional revolved solid or shell, you must enter the angle through which to rotate the profile.

   Abaqus/CAE exits the Sketcher and displays the new part in the current viewport.

8. If necessary, use the Part module tools to add additional features to the base feature. For more information, see *What is feature-based modeling?*.

Additional information

- *Using the Create Part dialog box*
- *Managing parts*
- *How is a part defined in Abaqus/CAE?*
- *Including draft in an extrusion*
- *Including twist in an extrusion*
- *The Sketch module*
- *Importing and exporting geometry data and models*
- *Using the Part module toolbox*
- *What is feature-based modeling?*
- *Using basic dialog box components*

Copying a part

Select **Part->Copy->*part name*** from the main menu bar to copy a part to a new part. You can copy parts containing both geometry and orphan elements. The compress, scale, and mirror functions operate on both geometry and meshes. The separate disconnected regions function also works with parts containing both geometry and orphan elements; however, only geometry regions are copied to become new parts since regions do not exist for orphan elements.

1. From the main menu bar, select **Part->Copy->*part name***.

   Abaqus/CAE displays the **Part Copy** dialog box.
2. From the dialog box, enter the name of the new part.
3. You can create an identical copy of the original part, or you can select from the following **Copy options**:

   - **Compress features**
   - **Scale part by**
   - **Mirror part about**
   - **Separate disconnected regions into parts**

   For more information on the **Copy options**, see *Copying a part*.
4. Click **OK**.

   Abaqus/CAE closes the **Part Copy** dialog box and copies the selected part to a new part. The new part becomes the current part. You can return to the original part by selecting the part from the **Part** list on the context bar. For more information, see *The context bar*.

Additional information

• *Simplifying a part's feature list*

Using the Create Part dialog box

This section describes the options in the **Create Part** dialog box.

In this section:

- *Using the Create Part dialog box to define the properties of a part*
- *Choosing the modeling space of a new part*
- *Choosing the type of a new part*
- *Choosing the base feature of a new part*
- *Setting the approximate size of the new part*

Using the Create Part dialog box to define the properties of a part

When you create a part, you first use the **Create Part** dialog box to define the properties of the part, and then you use the **Sketch** to sketch the two-dimensional profile of the base feature.

You use the **Create Part** dialog box to define the following:

Name

Use the **Name** text field at the top of the **Create Part** dialog box to name the part you are creating. To rename a part, select **Part->Rename** from the main menu bar. The part name should not be the same as the model name. For more information on valid names, see *Using basic dialog box components*.

After you create a part, Abaqus/CAE displays the name of the new part in title bar of the current viewport.

Modeling Space

Use the **Modeling Space** radio buttons to choose the modeling space of the new part. You can define a part to be either three-dimensional, two-dimensional (planar), or axisymmetric in an Abaqus/Standard or Abaqus/Explicit model.

If you create an axisymmetric deformable part, you can toggle on **Include twist** in the **Create Part** dialog box to include a twist degree of freedom in your model. You can change the modeling space of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears. For more information, see *Choosing the modeling space of a new part*.

Type

In an Abaqus/Standard or Abaqus/Explicit model, use the **Type** radio buttons to choose the type of the new part. You can define a part to be either deformable, discrete rigid, analytical rigid, or Eulerian. If allowed, you can change the type of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears. For more information, see *Choosing the type of a new part*.

Base Feature

Use the **Base Feature** field to define the shape and the type of the new part's base feature. The shape and the type options that Abaqus/CAE displays depend on the part's modeling space and type. You cannot change the type of a part's base feature after you create it. For more information, see *Choosing the base feature of a new part*.

Approximate size

Use the approximate size text field to enter the size of the part. The size that you enter is used by Abaqus/CAE to calculate the size of the Sketcher sheet and the spacing of its grid. For more information, see *Setting the approximate size of the new part*. After you create the part and start sketching its profile, you can use the **Sketch** customization options to increase the sheet size. To display the Sketcher customization options click the  tool at the bottom of the Sketcher toolbox.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Creating a new part*
- *Using the Create Part dialog box*

Choosing the modeling space of a new part

Use the **Modeling Space** radio buttons at the top of the **Create Part** dialog box to choose the modeling space of the part you are creating.
Abaqus/CAE carries a part's modeling space through the modeling process; for example, modeling space determines which tools are available in the Part module and which elements are available in the Mesh module. You can change the modeling space of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears.

Modeling space refers to the space the part can inhabit, rather than to the part itself. Thus, you can create a part in three-dimensional modeling space but construct it using topologically two-dimensional shell or wire features. In an Abaqus/Standard or Abaqus/Explicit model, the new part's modeling space can be set to any of the following:

Three-dimensional

Abaqus/CAE positions the part in three-dimensional space.

Two-dimensional planar

Abaqus/CAE positions the part in planar, two-dimensional space.

Axisymmetric

Abaqus/CAE positions the part in axisymmetric two-dimensional space. If you create an axisymmetric deformable part, Abaqus/CAE allows you to include a twist degree of freedom in your model.

1. From the top of the **Create Part** dialog box, choose the desired **Modeling Space** radio button.
2. When you have finished choosing options, click **Continue** to close the **Create Part** dialog box.
   The Sketcher starts, and you sketch the profile of the new part's base feature.

Additional information

- *Part modeling space*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *How is a part defined in Abaqus/CAE?*
- *Creating a new part*

Choosing the type of a new part

Use the **Type** radio buttons in the **Create Part** dialog box to choose the type of the part you are creating in an Abaqus/Standard or Abaqus/Explicit model.

Abaqus/CAE carries a part's type through the modeling process; for example, you cannot assign section and material properties to a rigid part, you cannot mesh an analytical rigid part, and you can assign Eulerian sections and material assignment fields only to Eulerian parts. If allowed, you can change the type of a part after you have created it by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears.

In an Abaqus/Standard or Abaqus/Explicit model a new part's type can be set to one of the following:

Deformable

Any arbitrarily shaped axisymmetric, two-dimensional, or three-dimensional part that you can create or import can be specified as a deformable part. A deformable part represents a part that can deform under load; the load can be mechanical, thermal, or electrical. By default, Abaqus/CAE creates parts that are deformable.

Discrete rigid

A discrete rigid part is similar to a deformable part in that it can be any arbitrary shape. However, a discrete rigid part is assumed to be rigid and is used in contact analyses to model bodies that cannot deform.

Analytical rigid

An analytical rigid part is similar to a discrete rigid part in that it is used to represent a rigid surface in a contact analysis. However, the shape of an analytical rigid part is not arbitrary and must be formed from a set of sketched lines, arcs, and parabolas.

Eulerian

Eulerian parts are used to define a domain in which material can flow for an Eulerian analysis. Eulerian parts do not deform during an analysis; instead, the material within the part deforms under load and can flow across the rigid element boundaries. For more information about Eulerian analyses, see *Eulerian analyses*.

After you create either a discrete rigid part or an analytical rigid part, you must also do the following:

- Assign the rigid body reference point. You apply constraints or prescribe motion to the rigid body reference point in the Load module, and the same constraints or motion are applied to the entire rigid part. For more information, see *The reference point*.
- If the part is a discrete rigid part or an analytical rigid part, you must use the Surface toolset in the Assembly module to choose which side of the part represents the outer surface. For more information, see *The Set and Surface toolsets*.

1. From the middle of the **Create Part** dialog box, choose the desired **Type** radio button.
2. When you have finished choosing options, click **Continue** to close the **Create Part** dialog box.

   The Sketcher starts, and you sketch the profile of the new part's base feature.

Additional information

- *Analytical Rigid Surface Definition*
- *Rigid Body Definition*
- *Part types*
- *How is a part defined in Abaqus/CAE?*
- *Creating a new part*

- *Using the Create Part dialog box*
- *The reference point*
- *The Set and Surface toolsets*

Choosing the base feature of a new part

Use the radio buttons and the list within the **Base Feature** field at the bottom of the **Create Part** dialog box to select the base feature of the part you are creating. The choices depend on both the part's modeling space and the part's type; for example, an axisymmetric deformable body can have only a planar shell, planar wire, or point base feature. For detailed information about the different shapes and types of base features that you can create, see *The base feature*.

Your choice of the base feature's type is important because you cannot change the type after you create the part. You can modify the base feature, but you should be aware that any features you subsequently add to the part will be linked to the base feature. Consequently, if you modify the base feature, these dependent (or child) features may move or fail to regenerate.

1. From the bottom of the **Create Part** dialog box, choose the desired base feature shape (**Solid**, **Shell**, **Wire**, or **Point**). The available choices depend on the modeling space and the type of the part you are creating.
2. If you are creating a three-dimensional deformable or discrete rigid part, you must also choose its **Type** (**Extrusion**, **Revolution**, **Sweep**, **Planar**, or **Coordinates**).
3. When you have finished choosing options, click **Continue** to close the **Create Part** dialog box.

   The Sketcher starts, and you sketch the profile of the new part's base feature.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Creating a new part*
- *Using the Create Part dialog box*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *The Feature Manipulation toolset*

Setting the approximate size of the new part

Use the **Approximate size** text field at the bottom of the **Create Part** dialog box to set the approximate size of the new part. The size that you enter is used by Abaqus/CAE to calculate the size of the Sketcher sheet and the spacing of its grid. The approximate part size must be between 10000 (10^4) and 0.001 (10^{-3}) units. Abaqus/CAE does not use specific units, but the units must be consistent throughout the model.

When you exit the **Create Part** dialog box, Abaqus/CAE enters the Sketcher, and you sketch the profile of the base feature. The Sketcher displays a square sheet with an overlaying grid and adjusts the dimensions of the sheet to approximate the size of the part. As a result, the dimensions of the sketch will have the same order of magnitude as the part you are creating.

If you subsequently edit the part, Abaqus/CAE still determines the size of the Sketcher sheet from the same dimensions that it used when you created the base feature. Consequently, you should set the approximate size of the part to match the largest dimension of the finished part. If you find subsequently that the part exceeds the size of the Sketcher sheet, use the **Sketch** customization options to increase the sheet size.

1. Type the approximate size of the new part in the **Approximate size** text field at the bottom of the **Create Part** dialog box.
2. When you have finished choosing options, click **Continue** to close the **Create Part** dialog box.

   The Sketcher starts with a sheet size and grid spacing based on the approximate size of the new part, and you sketch the profile of the new part's base feature.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Using the Create Part dialog box*
- *The Sketcher sheet and grid*
- *Customizing the Sketcher*

Adding a feature to a part

Use the **Shape** menu to add a feature to the current part. You can do the following:

- Use the **Solid** tools to add a solid feature to a to a three-dimensional solid part. For more information, see *Adding a solid feature*.
- Use the **Shell** tools to add a shell feature to a part. For more information, see *Adding a shell feature*.
- Use the **Wire** tools to add a wire feature to a part. For more information, see *Adding a wire feature*.
- Use the **Cut** tools to add a cut feature to a part. For more information, see *Adding a cut feature*.
- Use the **Blend** tools to add a blend feature to a three-dimensional solid part. For more information, see *Blending edges*.
- Use the **Transform** tool to mirror a part about a selected plane. For more information, see *Mirroring a part*.

Adding a solid feature

This section describes the Part module tools used to add a solid feature to the three-dimensional solid part in the current viewport.

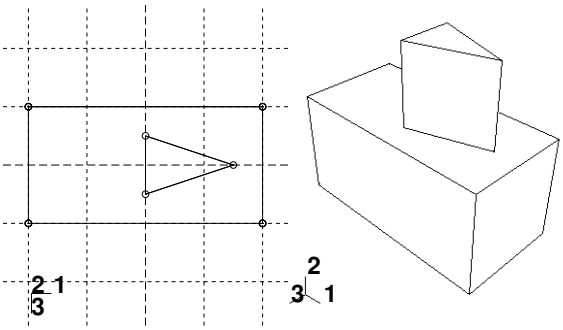
In this section:

- *Adding an extruded solid feature*
- *Adding a revolved solid feature*
- *Adding a swept solid feature*
- *Adding a solid loft feature*
- *Creating a solid feature from a shell*

Adding an extruded solid feature

Select **Shape->Solid->Extrude** from the main menu bar to add an extruded solid feature to the part in the current viewport. You can add an extruded solid feature only to three-dimensional parts.

You add an extruded solid feature by sketching a two-dimensional cross-section and defining the distance over which to extrude it. A sketch and the resulting extruded solid feature are illustrated in the following figure:



You can also define the distance over which to extrude by selecting a single face to extrude to. Abaqus/CAE extrudes the sketch until it meets the selected face.

In addition, you can select a center point and specify a pitch that Abaqus/CAE uses to twist the cross-section as it is extruded. Alternatively, Abaqus/CAE can expand or contract the cross-section along a specified draft angle as the cross-section is extruded. For more information, see *Including twist in an extrusion*, and *Including draft in an extrusion*.

1. From the main menu bar, select **Shape->Solid->Extrude**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add an extruded solid feature using the tool, located with the solid tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the extruded solid feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. Select the planar face from which the solid will be extruded. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The selected face is highlighted in the viewport.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

5. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If there is no straight edge with the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool to return to the original view.

6. Use the Sketcher to sketch the two-dimensional profile of the extrusion. In the prompt area, click **Done** to exit the Sketcher and to open the **Edit Extrusion** dialog box.

   Abaqus/CAE displays the part view that was active prior to entering the Sketcher. The part includes your sketched profile and an arrow indicating the extrusion direction.

7. Click in the **Edit Extrusion** dialog box to reverse the extrusion direction, if necessary.

   If the arrow direction is difficult to see, use the rotate tool to rotate the part.

8. Select one of the following end conditions:

   - Select **Blind** and enter a value in the **Depth** field to specify the distance through which Abaqus/CAE will extrude the sketched profile.
   - Select **Up to Face** to specify that Abaqus/CAE will extrude the profile up to a selected face.

9. If desired, do one of the following:

   - Toggle on **Include twist**, and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched extrusion profile must include an isolated point that indicates the center of twist.
   - Toggle on **Include draft**, and enter the draft angle (greater than −90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract.

10. Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the extruded solid feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.
11. Click **OK** to extrude the profile.

    If you selected the twist option and your sketch includes a single isolated point, Abaqus/CAE uses that point as the center of twist. If your sketch does not include an isolated point, Abaqus/CAE returns to the Sketcher for you to create one. If your sketch contains more than one isolated point, Abaqus/CAE returns to the Sketcher and prompts you to select an isolated point as the center of twist.

12. If you selected **Up to Face**, Abaqus/CAE prompts you to select the face to which to extrude the profile. Select a face to meet the following requirements:

    - the selected face does not have to be parallel to the sketch plane,
    - it can be a nonplanar face,
    - it must completely contain the extruded selection, and

- it cannot be a datum plane.

Abaqus/CAE creates the extruded solid feature.

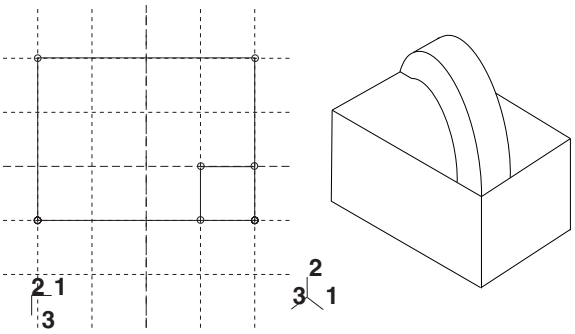
Additional information

- *What are extruding, revolving, and sweeping?*
- *Meshing complex solids with hexahedral elements*
- *What is feature-based modeling?*

Adding a revolved solid feature

Select **Shape->Solid->Revolve** from the main menu bar to add a revolved solid feature to the part in the current viewport. You can add a revolved solid feature only to three-dimensional parts.

You add a revolved solid feature by sketching a two-dimensional cross-section and a construction line on a selected face. The construction line serves as an axis of revolution, and Abaqus/CAE creates the solid feature by rotating the cross-section about the axis using a specified angle of revolution. In addition, you can specify a pitch and a direction along the axis of revolution that Abaqus/CAE uses to translate the sketch along the axis of revolution as it revolves the profile. A sketch and the resulting feature, revolved through an angle of 180° with pitch, are illustrated in the following figure:



1. From the main menu bar, select **Shape->Solid->Revolve**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a revolved solid feature using the tool, located with the solid tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the revolved solid feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. Select the planar face from which the solid will be revolved. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The selected face is highlighted in the viewport.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

5. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If there is no straight edge with the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

6. Use the horizontal , vertical , angle , or oblique  construction line tools to sketch the axis of rotation. You can position the construction line by selecting a datum axis from the underlying part. You cannot select the datum axis directly; you must select a point from either end of the datum axis.
7. Use the Sketcher to sketch the two-dimensional profile of the revolved feature; the sketch must not cross the axis of revolution.
8. In the prompt area, click **Done** to indicate you have finished sketching the profile and the axis. If the sketch contains more than one construction line, Abaqus/CAE prompts you to select the construction line that will serve as the axis of rotation.

   Abaqus/CAE displays the part view that was active prior to entering the Sketcher. The part includes your sketched profile and an arrow indicating the direction of revolution. The **Edit Revolution** dialog box appears.

9. In the **Edit Revolution** dialog box, enter the desired angle of revolution or accept the default value.
10. Click  next to **Revolve direction** to change the arrow direction and the associated direction of revolution.
11. If desired, toggle on **Include translation** and enter a positive value for the pitch. The pitch value is the distance through which the profile is translated along the axis of revolution during a rotation of 360°.

    An arrow appears to show the axis of revolution and to indicate the direction of sketch translation along the axis. Click  next to **Pitch direction** in the **Edit Revolution** dialog box to reverse the arrow, if necessary.

12. If desired, toggle on **Sweep sketch normal to path** to rotate the sketched profile normal to the path of revolution. This option is available only when **Include translation** is toggle on.

    The initial profile of the feature will be rotated from the sketch plane to create the feature.

13. Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the revolved solid feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.
14. Click **OK** to accept the indicated direction and to create the revolved solid feature.

Abaqus/CAE creates the revolved feature using your selected parameters.

Additional information

- *Creating construction geometry*
- *Adding a solid feature*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *Controlling the cross-section of a revolved feature with pitch*
- *Meshing complex solids with hexahedral elements*
- *What is feature-based modeling?*

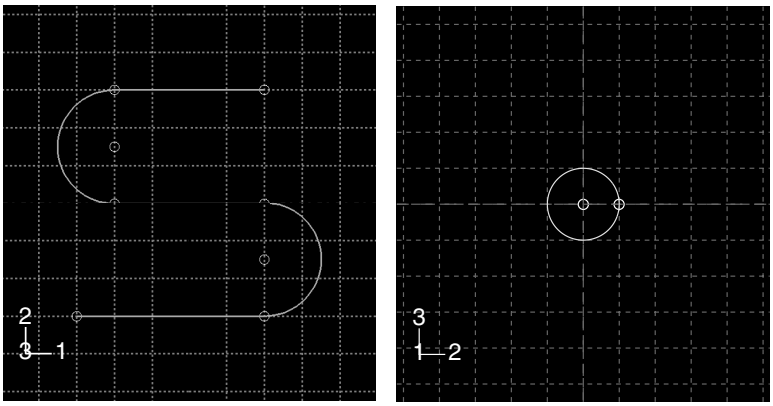
Adding a swept solid feature

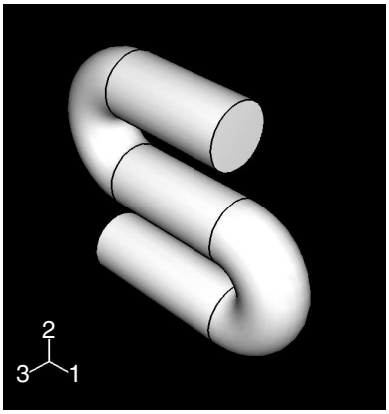
Select **Shape**->**Solid**->**Sweep** from the main menu bar to add a swept solid feature to the part in the current viewport. You can add a swept solid feature only to three-dimensional parts.

You add a swept solid feature by first defining a sweep path, then defining a sweep profile. There are different methods available for defining each component:

- You can define a sweep path either by sketching the path on a selected face or by selecting a series of edges or wires that you want the sweep path to follow. The sketch method provides greater flexibility but supports only two-dimensional paths. The edge/wire method enables you to define a three-dimensional sweep path along a feature such as a spline wire or a set of edges in a three-dimensional part.
- You can define a sweep profile either by sketching a sweep profile using the Sketcher or by selecting one of the faces in your model as the profile. The sweep profile is initially perpendicular to the path; you can keep this orientation constant along the entire sweep path, or you can keep the sweep profile perpendicular to the sweep path as it is swept along its length.

The sweep path, the sweep profile, and the resulting solid feature are illustrated in the following figures:





The sketch or set of edges specifying the sweep path and the sketch or face specifying the sweep profile define the swept solid feature; both the sweep path and the sweep profile can be modified using the Feature Manipulation toolset if you define them using the Sketcher. You can also create a swept solid whose sweep profile is offset from the sweep path. In this case Abaqus/CAE moves the sweep path to a parallel location that passes through the sweep profile and creates the swept solid at that location.

As you define your sweep feature, you can apply a twist or draft to it. For more information about these tools, see *What types of features can you create?*. You can toggle on **Keep profile normal constant** to maintain the same orientation for the sweep profile as it travels along the sweep path; if this option is toggled off, the sweep profile orientation changes with the normal to the sweep path. In addition, you can toggle on **Keep internal boundaries** in the Feature Manipulation toolset to maintain any faces or edges that are generated between the swept solid feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.

1. From the main menu bar, select **Shape->Solid->Sweep**.

   Tip: You can also add a swept solid feature using the  tool, located with the solid tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   The **Create Solid Sweep** dialog box appears.

2. If you want to sketch the sweep path, do the following:

   a. From the **Path** options, select **Sketch** and click .

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. If desired, specify the method you want to use to select an origin for your sketch of the swept solid feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

   c. Select the planar face on which to sketch the sweep path. If no suitable face exists, you can select a datum plane.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   d. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

   e. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If there is no straight edge with the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part so that the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

f. Sketch the sweep path. The sweep path must meet the following guidelines:

- The path can be closed, but the ends must meet smoothly; for example, the ends should not meet at a corner. For examples of valid sweep paths, see *Defining the sweep path and the sweep profile*.
- The path must be continuous; for example, it must not branch.
- The resulting solid cannot intersect with itself.

In the prompt area, click **Done** to indicate you have finished sketching the sweep path.

Abaqus/CAE exits the Sketcher and restores the original view of the part. A highlighted line indicates the sweep path and its direction. Abaqus/CAE also reopens the **Create Solid Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Path** options to indicate that the sweep path has been defined in the Sketcher.

3. If you want to specify the sweep path as a series of edges or wires, do the following:

a. From the **Path** options, select **Edges** and click .

Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

b. If desired, specify in the prompt area whether you want to select the edges in your sweep path **individually** or **by edge angle**. For more information about selecting objects, see *Using the angle and feature edge method to select multiple objects*.

c. Select the edges you want to include in the sweep path.

Abaqus/CAE shows the sweep path on your part and indicates the sweep direction.

d. In the prompt area, click **Yes** to confirm the sweep path direction or click **Flip** to reverse the sweep path direction.

Abaqus/CAE reopens the **Create Solid Sweep** dialog box and adds the word **(Defined)** next to the **Edges** label in the **Path** options to indicate that the sweep path has been defined using a series of edges.

4. If you want to sketch the sweep profile, do the following:

a. From the **Profile** options, select **Sketch** and click .

Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

b. Sketch the sweep profile. The sweep profile must meet the following guidelines:

- The profile must be closed.
- The resulting solid cannot intersect with itself.

You can sketch the profile anywhere on the Sketcher grid; Abaqus/CAE sweeps the profile along a path parallel to the sweep path. In the prompt area, click **Done** to indicate you have finished sketching the sweep profile.

Abaqus/CAE exits the Sketcher and restores the original view of the part.

c. Select an edge and the orientation of the edge on the Sketcher grid. The selected edge must not be parallel to the sweep path direction. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

Abaqus/CAE highlights the selected edge, enters the Sketcher again, and rotates the part so that the Sketcher grid lies on a plane normal to the beginning of the sweep path with the sweep path direction pointing out of the screen. In addition, the selected edge aligns with the grid in the desired orientation. The intersection of two dashed lines indicates the origin of the sweep path. Abaqus/CAE also reopens the **Create Solid Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Profile** options to indicate that the sweep profile has been defined in the Sketcher.

5. If you want to select a face as the sweep profile, do the following:

   a. From the **Profile** options, select **Face** and click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. Select a face from the viewport.

      Abaqus/CAE highlights the selected face and displays the **Create Solid Sweep** dialog box.

   Abaqus/CAE reopens the **Create Solid Sweep** dialog box and adds the word **(Defined)** next to the **Faces** label in the **Profile** options to indicate that the sweep profile has been defined using a face.

6. If desired, do any of the following:

   - Toggle on **Include twist**, and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched extrusion profile must include an isolated point that indicates the center of twist.
   - Toggle on **Include draft**, and enter the draft angle (greater than −90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract. You cannot apply a draft if the **Keep profile normal constant** option is selected.
   - Toggle on **Keep profile normal constant** to maintain the same profile orientation along the entire sweep path. If this option is toggled off, Abaqus/CAE adjusts the profile orientation so the angle between the sweep profile and the normal to the sweep path is always constant. You cannot toggle on this option if the **Include draft** option is selected.
   - Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the swept solid feature and the existing part. The internal boundaries can create regions that can be structured or swept meshed without having to resort to partitioning.

7. Click **OK** to create the new swept solid.

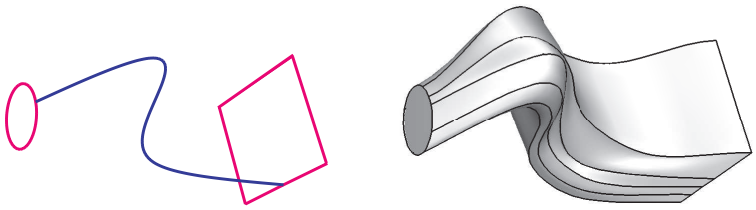
Additional information

- *Adding a solid feature*
- *Defining the sweep path and the sweep profile*
- *The Sketch module*
- *What is feature-based modeling?*

Adding a solid loft feature

Select **Shape->Solid->Loft** from the main menu bar to add a solid loft feature to the part in the current viewport. You can add a solid loft feature only to three-dimensional parts.

You add a solid loft feature by creating two or more sections from selected edges and by defining one or more loft paths. Loft sections, a loft path, and the resulting solid loft feature are illustrated in the following figure:



You can allow Abaqus/CAE to define a single loft path using a smooth path to connect the center of each loft section. If you allow Abaqus/CAE to define the path, you can apply tangency methods to the start and end sections of the loft. The curve and tangencies define the path of the loft feature between sections. Alternatively, you can define one or more loft paths by selecting curves that connect a point on each loft section to a point on the next loft section. Each loft path must provide a continuous line connecting each consecutive loft section. If a loft path is not smooth (if there is more than one tangent to any point along the path), Abaqus/CAE will display an error message when you try to create the loft. For more information about loft sections, loft paths, and loft tangencies, see *What is lofting?*.

Note:

You do not use the Sketcher while adding a loft feature. As a result, all of the edges that define the loft sections and the loft paths must exist in the part geometry before you create the loft. To create a loft path or to create a nonplanar loft section, you can use the tool, located with the wire tools in the Part module toolbox, to create a spline wire (see *Adding a point-to-point wire feature*, for more information).

1. From the main menu bar, select **Shape->Solid->Loft**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a solid loft feature using the tool, located with the solid tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   The **Edit Loft** dialog box appears.
2. Create the loft sections by selecting edges from the part in the viewport. See *Creating loft sections*, for detailed instructions on creating loft sections.
3. Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the lofted solid feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.
4. When you have finished creating the loft sections, click on the **Transition** tab in the **Edit Loft** dialog box.
5. Do one of the following:

- Click **Select path** to create the loft path (or paths) by selecting edges from the part in the viewport.
- Click **Specify tangencies** to create the loft path (or paths) by using loft tangencies.

See *Creating a loft path*, for detailed instructions on creating a loft path.

6. Click the **Preview** button in the **Edit Loft** dialog box.

   Abaqus/CAE displays a wireframe representation of the loft that would be created with your current settings.

7. If desired, you can add or remove loft sections, change the loft path definition method, or edit the loft tangency options to change the shape of the loft feature. Click **Preview** to see the effect of your changes in the viewport.
8. If desired, you can have Abaqus/CAE test for self-intersection as it creates the loft feature. This test prevents creating a feature that would be difficult or impossible to mesh and analyze, but it becomes computationally expensive as the complexity of the loft feature increases. For more information, see *Self-intersection checks*. To use self-intersection checks, select **Feature->Options** to open the **Feature Options** dialog box, and toggle on **Perform self-intersection checks**.
9. Click **Done** to create the loft and to close the **Edit Loft** dialog box.

   If you choose to test for self-intersection and the test fails, the **Edit Loft** dialog box will reappear so that you can make changes. Otherwise, the solid loft feature is created in the viewport.

Additional information

- *Adding a solid feature*
- *What is feature-based modeling?*

Creating a solid feature from a shell

Select **Shape->Solid->From Shell** from the main menu bar to create a solid feature from a three-dimensional shell part by selecting the faces that will form a closed part. Abaqus/CAE adds the solid material to change the region defined by the selected faces from a shell to a solid.

1. From the main menu bar, select **Shape->Solid->From Shell**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a solid feature from a shell by clicking the  tool, located with the solid tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Select faces from the shell that should be converted to a solid, and click mouse button 2 to indicate you have finished selecting faces.

   If you selected more than one face, Abaqus/CAE chooses the direction in which to add the solid material and changes the regions with the selected faces from a shell to a solid.

3. If you selected a single face, Abaqus/CAE highlights the face and displays an arrow indicating the direction in which material will be added to create the solid. If desired, click **Flip** to reverse the direction of the arrow.
4. Click mouse button 2 to confirm the direction of the arrow.

   Abaqus/CAE fills the shell in the direction indicated and creates a solid region.

5. Click **Done** to create the solid part.

Additional information

- *Adding a solid feature*
- *What is feature-based modeling?*

Adding a shell feature

This section describes the tools used to add a shell feature to the part in the current viewport.

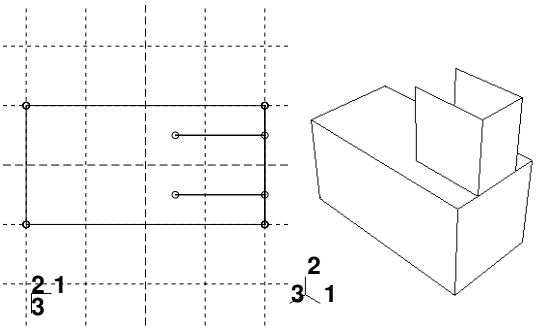
In this section:

- *Adding an extruded shell feature*
- *Adding a revolved shell feature*
- *Adding a swept shell feature*
- *Adding a shell loft feature*
- *Adding a planar shell feature*
- *Adding a shell-from-solid feature*

Adding an extruded shell feature

Select **Shape->Shell->Extrude** from the main menu bar to add an extruded shell feature to the part in the current viewport. You can add an extruded shell feature only to three-dimensional parts.

You add an extruded shell feature by sketching on a selected face and extending the profile a specified distance in a direction normal to the face. A sketch and the resulting extruded shell feature are illustrated in the following figure:



You can also define the distance over which to extrude by selecting a single face to extrude to. Abaqus/CAE extrudes the sketch until it meets the selected face.

In addition, you can select a center point and specify a pitch that Abaqus/CAE uses to twist the cross-section as it is extruded. Alternatively, Abaqus/CAE can expand or contract the cross-section along a specified draft angle as the cross-section is extruded. For more information, see *Including twist in an extrusion*, and *Including draft in an extrusion*.

1. From the main menu bar, select **Shape->Shell->Extrude**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add an extruded shell feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the extruded shell feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. Select the planar face from which the shell will be extruded. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The selected face is highlighted in the viewport.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You

can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

5. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

6. Use the Sketcher to sketch the profile of the line to be extruded. In the prompt area, click **Done** to exit the Sketcher and to open the **Edit Extrusion** dialog box.

   Abaqus/CAE displays the part view that was active prior to entering the Sketcher. The part includes your sketched profile and an arrow indicating the extrusion direction.

7. Click  in the **Edit Extrusion** dialog box to reverse the extrusion direction, if necessary.

   If the arrow direction is difficult to see, use the rotate tool to rotate the part.

8. Select one of the following end conditions:

   - Select **Blind** and enter a value in the **Depth** field to specify the distance through which Abaqus/CAE will extrude the sketched profile.
   - Select **Up to Face** to specify that Abaqus/CAE will extrude the profile up to a selected face.

9. If desired, do one of the following:

   - Toggle on **Include twist**, and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched extrusion profile must include an isolated point that indicates the center of twist.
   - Toggle on **Include draft**, and enter the draft angle (greater than −90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract.

     Note:

     When you use draft with a shell feature, discontinuous profiles cannot be used. A discontinuous profile is a profile that is connected but cannot be traced in one continuous motion. If you specify draft on a discontinuous profile, Abaqus/CAE applies the draft angle to each face individually and the extruded edges are not joined.

10. Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the extruded shell feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.
11. Click **OK** to extrude the profile.

If you selected the twist option and your sketch includes a single isolated point, Abaqus/CAE uses that point as the center of twist. If your sketch does not include an isolated point, Abaqus/CAE returns to the Sketcher for you to create one. If your sketch contains more than one isolated point, Abaqus/CAE returns to the Sketcher and prompts you to select an isolated point as the center of twist.

12. If you selected **Up to Face**, Abaqus/CAE prompts you to select the face to which to extrude the profile. Select a face to meet the following requirements:

    - the selected face does not have to be parallel to the sketch plane,
    - it can be a nonplanar face,
    - it must completely contain the extruded selection, and
    - it cannot be a datum plane.

    Abaqus/CAE creates the extruded shell feature.

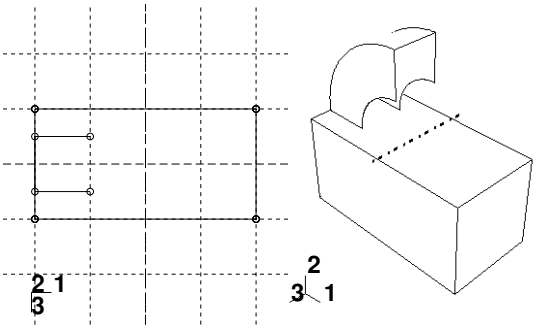
Additional information

- *The Datum toolset*
- *Adding a shell feature*
- *The Sketch module*
- *What is feature-based modeling?*

Adding a revolved shell feature

Select **Shape->Shell->Revolve** from the main menu bar to add a revolved shell feature to the part in the current viewport. You can add a revolved shell feature only to three–dimensional parts.

You add a revolved shell feature by sketching a profile and a construction line on a selected face. The construction line serves as an axis of revolution, and Abaqus/CAE creates the shell feature by rotating the profile about the axis using a specified angle of revolution. A sketch and the resulting feature, rotated about the axis of revolution through an angle of 90°, are illustrated in the following figure:



In addition, you can specify a pitch and a direction along the axis of revolution that Abaqus/CAE uses to translate the sketch along the axis of revolution as it revolves the profile.

1. From the main menu bar, select **Shape->Shell->Revolve**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a revolved shell feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the revolved shell feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. Select the planar face from which the shell will be revolved. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The selected face is highlighted in the viewport.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

5. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool to return to the original view.

6. Use the horizontal, vertical, angle, or oblique construction line tools to sketch the axis of rotation. You can position the construction line by selecting a datum axis from the underlying part. You cannot select the datum axis directly; you must select a point from either end of the datum axis.
7. Use the Sketcher to sketch the two-dimensional profile of the revolved feature; the sketch must not cross the axis of revolution.
8. In the prompt area, click **Done** to indicate you have finished sketching the profile and the axis. If the sketch contains more than one construction line, Abaqus/CAE prompts you to select the construction line that will serve as the axis of rotation.

   Abaqus/CAE displays the part view that was active prior to entering the Sketcher. The part includes your sketched profile and an arrow indicating the direction of revolution. The **Edit Revolution** dialog box appears.
9. In the **Edit Revolution** dialog box, enter the desired angle of revolution or accept the default value.
10. Click next to **Revolve direction** to change the arrow direction and the associated direction of revolution.
11. If desired, toggle on **Include translation**, and enter a positive value for pitch. The pitch value is the distance through which the profile is translated along the axis of revolution during a rotation of 360°.

    An arrow appears to show the axis of revolution and to indicate the direction of sketch translation along the axis. Click next to **Pitch direction** in the **Edit Revolution** dialog box to reverse the arrow, if necessary.
12. If desired, toggle on **Sweep sketch normal to path** to rotate the sketched profile normal to the path of revolution. This option is available only when **Include translation** is used.

    The initial profile of the feature will be rotated from the sketch plane to create the feature.
13. Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the revolved shell feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.
14. Click **OK** to accept the indicated direction and to create the revolved shell feature.

Abaqus/CAE creates the revolved feature using your selected parameters.

Additional information

- *Creating construction geometry*
- *Adding a shell feature*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *Controlling the cross-section of a revolved feature with pitch*
- *The Sketch module*
- *What is feature-based modeling?*

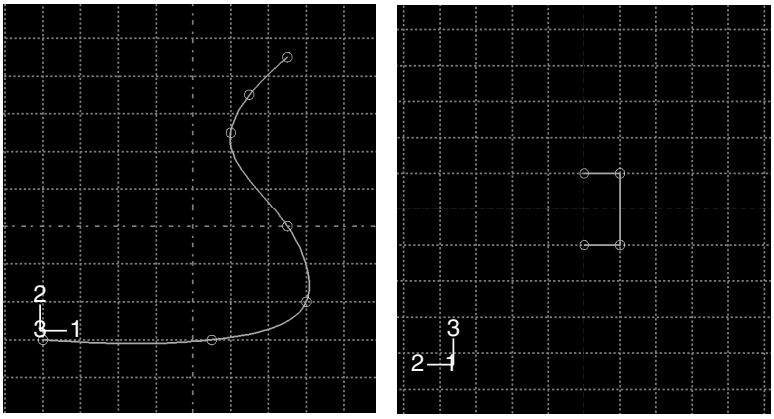
Adding a swept shell feature

Select **Shape**->**Shell**->**Sweep** from the main menu bar to add a swept shell feature to the part in the current viewport. You can add a swept shell feature only to three-dimensional parts.

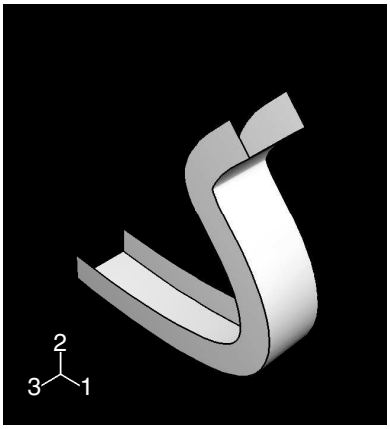
You add a swept shell feature by first defining a sweep path, then defining a sweep profile. There are different methods available for defining each component:

- You can define a sweep path either by sketching the path on a selected face or by selecting a series of edges or wires that you want the sweep path to follow. The sketch method provides greater flexibility but supports only two-dimensional paths. The edge/wire method enables you to define a three-dimensional sweep path along a feature such as a spline wire or a set of edges in a three-dimensional part.
- You can define a sweep profile either by sketching a sweep profile using the Sketcher or by selecting one of the faces in your model as the profile. The sweep profile is initially perpendicular to the path; you can keep this orientation constant along the entire sweep path, or you can keep the sweep profile perpendicular to the sweep path as it is swept along its length.

The sweep path (a spline) and the sweep profile are shown in the following figure:



The resulting swept shell feature is shown in the following figure:



The sketch or set of edges defining the sweep path and the sketch, edge, or edges defining the sweep profile combine to define the swept shell feature; both the sweep path and the sweep profile can be modified using the Feature Manipulation toolset if you define them using the Sketcher. You can define a swept shell whose sweep profile is offset

from the sweep path. In this case Abaqus/CAE moves the sweep path to a parallel location that passes through the sweep profile and creates the swept shell feature at that location.

As you define your sweep feature, you can apply a twist or draft to it. For more information about these tools, see *What types of features can you create?*. You can toggle on **Keep profile normal constant** to maintain the same orientation for the sweep profile as it travels along the sweep path; if this option is toggled off, the sweep profile orientation changes with the normal to the sweep path. In addition, you can toggle on **Keep internal boundaries** in the Feature Manipulation toolset to maintain any faces or edges that are generated between the swept shell feature and the existing part. The internal boundaries can create regions that can be structured or swept meshed without having to resort to partitioning.

1. From the main menu bar, select **Shape->Shell->Sweep**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a swept shell feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If you want to sketch the sweep path, do the following:

   a. From the **Path** options, select **Sketch** and click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. If desired, specify the method you want to use to select an origin for your sketch of the swept solid feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

      - Select **Auto-Calculate** to place the sketch origin automatically.
      - Select **Specify** to define a custom sketch origin.
      - Select **Session Default** to use the custom origin you specified earlier in the session.

   c. Select the planar face on which to sketch the sweep path. If no suitable face exists, you can select a datum plane.

      Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   d. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

   e. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

      Tip: If there is no straight edge with the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

      Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part so that the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

f. Sketch the sweep path. The sweep path must meet the following guidelines:

- The path can be closed, but the ends must meet smoothly; for example, the ends should not meet at a corner. For examples of valid sweep paths, see *Defining the sweep path and the sweep profile*.
- The path must be continuous; for example, it must not branch.
- The resulting solid cannot intersect with itself.

In the prompt area, click **Done** to indicate you have finished sketching the sweep path.

Abaqus/CAE exits the Sketcher and restores the original view of the part. A highlighted line indicates the sweep path and its direction. Abaqus/CAE also reopens the **Create Shell Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Path** options to indicate that the sweep path has been defined in the Sketcher.

3. If you want to specify the sweep path as a series of edges or wires, do the following:

a. From the **Path** options, select **Edges** and click .

Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

b. If desired, specify in the prompt area whether you want to select the edges in your sweep path **individually** or **by edge angle**. For more information about selecting objects, see *Using the angle and feature edge method to select multiple objects*.

c. Select the edges you want to include in the sweep path.

Abaqus/CAE shows the sweep path on your part and indicates the sweep direction.

d. In the prompt area, click **Yes** to confirm the sweep path direction or click **Flip** to reverse the sweep path direction.

Abaqus/CAE reopens the **Create Shell Sweep** dialog box and adds the word **(Defined)** next to the **Edges** label in the **Path** options to indicate that the sweep path has been defined using a series of edges.

4. If you want to sketch the sweep profile, do the following:

a. From the **Profile** options, select **Sketch** and click .

Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

b. Sketch the sweep profile. The sweep profile must meet the following guidelines:

- The profile must be closed.
- The resulting solid cannot intersect with itself.

You can sketch the profile anywhere on the Sketcher grid; Abaqus/CAE sweeps the profile along a path parallel to the sweep path. In the prompt area, click **Done** to indicate you have finished sketching the sweep profile.

Abaqus/CAE exits the Sketcher and restores the original view of the part.

c. Select an edge and the orientation of the edge on the Sketcher grid. The selected edge must not be parallel to the sweep path direction. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

Abaqus/CAE highlights the selected edge, enters the Sketcher again, and rotates the part so that the Sketcher grid lies on a plane normal to the beginning of the sweep path with the sweep path direction pointing out of the screen. In addition, the selected edge aligns with the grid in the desired orientation. The intersection of two dashed lines indicates the origin of the sweep path. Abaqus/CAE also reopens the **Create Shell Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Profile** options to indicate that the sweep profile has been defined in the Sketcher.

5. If you want to select one or more edges as the sweep profile, do the following:

   a. From the **Profile** options, select **Edges** and click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. Select one or more edges from the viewport.

      Abaqus/CAE highlights the selected face and displays the **Create Shell Sweep** dialog box.

   Abaqus/CAE reopens the **Create Shell Sweep** dialog box and adds the word **(Defined)** next to the **Edges** label in the **Profile** options to indicate that the sweep profile has been defined using one or more edges.

6. If desired, do one of the following:

   - Toggle on **Include twist**, and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched extrusion profile must include an isolated point that indicates the center of twist.
   - Toggle on **Include draft**, and enter the draft angle (greater than −90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract. You cannot apply a draft if the **Keep profile normal constant** option is selected.
   - Toggle on **Keep profile normal constant** to maintain the same profile orientation along the entire sweep path. If this option is toggled off, Abaqus/CAE adjusts the profile orientation so that the angle between the sweep profile and the normal to the sweep path is always constant. You cannot toggle on this option if the **Include draft** option is selected.
   - Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the swept solid feature and the existing part. The internal boundaries can create regions that can be structured or swept meshed without having to resort to partitioning.

7. Click **OK** to create the new swept shell.

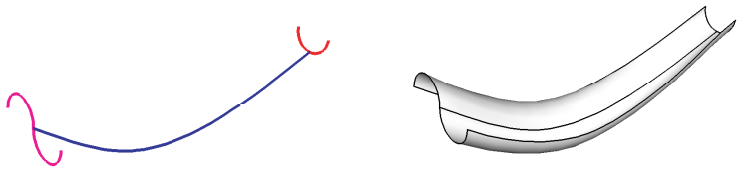
Additional information

- *Defining the sweep path and the sweep profile*
- *The Sketch module*
- *What is feature-based modeling?*

Adding a shell loft feature

Select **Shape->Shell->Loft** from the main menu bar to add a shell loft feature to the part in the current viewport. You can add a shell loft feature only to three-dimensional parts.

You add a shell loft feature by creating two or more sections from selected edges and by defining one or more loft paths. Loft sections, a loft path, and the resulting shell loft feature are illustrated in the following figure:



You can allow Abaqus/CAE to define a single loft path using a smooth path to connect the center of each loft section. If you allow Abaqus/CAE to define the path, you can apply tangency methods to the start and end sections of the loft. The curve and tangencies define the path of the loft feature between sections. Alternatively, you can define one or more loft paths by selecting curves that connect a point on each loft section to a point on the next loft section. Each loft path must provide a continuous line connecting each consecutive loft section. If a loft path is not smooth (if there is more than one tangent to any point along the path), Abaqus/CAE will display an error message when you try to create the loft. For more information about loft sections, loft paths, and loft tangencies, see *What is lofting?*.

Note:

You do not use the Sketcher while adding a loft feature. As a result, all of the edges that define the loft sections and the loft paths must exist in the part geometry before you create the loft. To create a loft path or to create a nonplanar loft section, you can use the tool, located with the wire tools in the Part module toolbox, to create a spline wire (see *Adding a point-to-point wire feature*, for more information).

1. From the main menu bar, select **Shape->Shell->Loft**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a shell loft feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   The **Edit Loft** dialog box appears.

2. Create the loft sections by selecting edges from the part in the viewport. See *Creating loft sections*, for detailed instructions on creating loft sections.

   Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the lofted shell feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.

3. When you have finished creating the loft sections, click on the **Transition** tab in the **Edit Loft** dialog box.
4. Do one of the following:

- Click **Select path** to create the loft path (or paths) by selecting edges from the part in the viewport.
- Click **Specify tangencies** to create the loft path (or paths) by using loft tangencies.

See *Creating a loft path*, for detailed instructions on creating a loft path.

5. Click **Preview**.

   Abaqus/CAE displays a wireframe representation of the loft that would be created with your current settings.

6. If desired, you can add or remove loft sections, change the loft path definition method, or edit the loft tangency options to change the shape of the loft feature. Click **Preview** to see the effect of your changes in the viewport.
7. If desired, you can have Abaqus/CAE test for self-intersection as it creates the loft feature. This test prevents creating a feature that would be difficult or impossible to mesh and analyze, but it becomes computationally expensive as the complexity of the loft feature increases. For more information, see *Self-intersection checks*. To use self-intersection checks, select **Feature->Options** to open the **Feature Options** dialog box, and toggle on **Perform self-intersection checks**.
8. Click **Done** to create the loft and to close the **Edit Loft** dialog box.

   If you choose to test for self-intersection and the test fails, the **Edit Loft** dialog box will reappear so that you can make changes. Otherwise, the shell loft feature is created in the viewport.

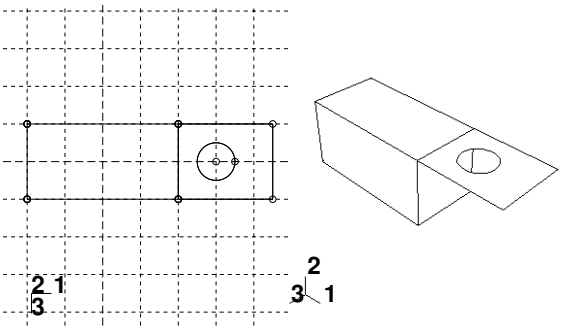
Additional information

- *Adding a shell feature*
- *What is feature-based modeling?*

Adding a planar shell feature

Select **Shape->Shell->Planar** from the main menu bar to add a planar shell feature to the part in the current viewport. The planar shell tool is always available, regardless of the modeling space of the part in the current viewport.

You add a planar shell feature by sketching the feature on a selected face. A sketch and the resulting planar shell feature are illustrated in the following figure:



The sketch defines a planar shell feature and can be modified using the Feature Manipulation toolset.

1. From the main menu bar, select **Shape->Shell->Planar**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a planar shell feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the planar shell feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. If the modeling space of the part is two-dimensional or axisymmetric, Abaqus/CAE enters the Sketcher and aligns the *X*- and *Y*-axes of the part and the sketch.

   If the modeling space of the part is three-dimensional, do the following:

   a. Select the planar face on which the shell will be positioned. If no suitable face exists, you can select a datum plane.

      Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   b. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

   c. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the

right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.



Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

4. Use the Sketcher to sketch the planar shell. In the prompt area, click **Done** to indicate you have finished sketching.

   The part returns to its original orientation with the planar shell positioned on the selected face. The shell feature is created only where it extends beyond the faces of the part; a shell feature cannot overlap a face.

Additional information

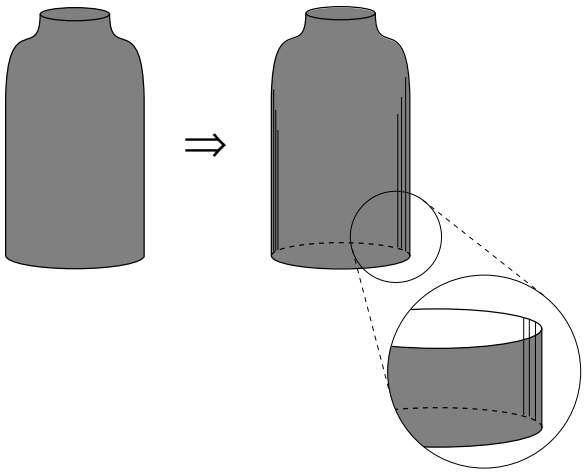
- *Adding a shell feature*
- *The Sketch module*
- *What is feature-based modeling?*

Adding a shell-from-solid feature

Select **Shape->Shell->From Solid** from the main menu bar to create a shell feature from the faces of a solid feature. You can add a shell-from-solid feature only to three-dimensional parts.

You add a shell-from-solid feature by selecting the cells to remove from the part; Abaqus/CAE converts any remaining faces associated with the removed cells to shells.

The **From Solid** tool is an easy way to create shells with curved edges, as shown in the following figure. The curved edges of the solid were created by filleting the edges using the round tool.



1. From the main menu bar, select **Shape->Shell->From Solid**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a shell–from–solid feature using the tool, located with the shell tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Select one or more cells to convert to shells. **[Shift] + Click** additional cells to add them to your selection and **[Ctrl] + Click** a selected cell to unselect it. Click mouse button 2 to indicate you have finished selecting cells to convert.

   Abaqus/CAE converts the selected cells to shells.

   Tip: Use the **Previous** button ( ) to undo one or more steps; use the cancel button ( ) to stop the creation of the shell from solid.

Additional information

- *Adding a shell feature*
- *The Sketch module*
- *What is feature-based modeling?*

Adding a wire feature

This section describes the Part module tools used to add a wire feature to the part in the current viewport. This section also describes adding a wire feature to an assembly.

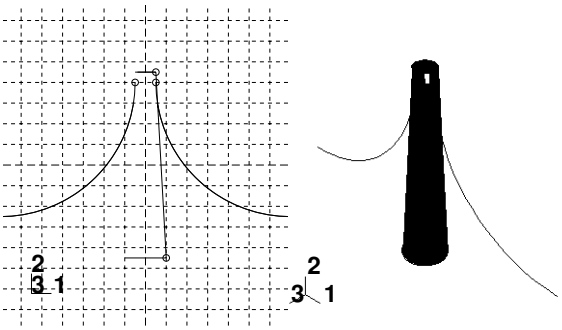
In this section:

- *Adding a sketched wire feature*
- *Adding a point-to-point wire feature*
- *Rounding vertices in a wire*
- *Adding a wire-from-edge feature*

Adding a sketched wire feature

Select **Shape->Wire->Sketch** from the main menu bar to add a sketched planar wire feature to the part in the current viewport. The planar wire tool is always available, regardless of the modeling space of the part in the current viewport.

You add a planar wire feature by sketching the feature on a selected plane. Abaqus/CAE removes any portion of the wire that overlaps an existing face. A sketch and the resulting planar wires are illustrated in the following figure:



The sketch fully defines a planar wire feature and can be modified using the Feature Manipulation toolset.

1. From the main menu bar, select **Shape->Wire->Sketch**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a sketched wire feature using the tool, located with the wire tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the sketched wire feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. If the modeling space of the part is two-dimensional or axisymmetric, Abaqus/CAE enters the Sketcher and aligns the *X*- and *Y*-axes of the part and the sketch.

   If the part is three-dimensional, do the following:

   a. Select the planar face on which the wire will be positioned. If no suitable face exists, you can select a datum plane.

      Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   b. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.



Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.
5. Use the Sketcher to sketch the planar wire. In the prompt area, click **Done** to indicate you have finished sketching.

   The part returns to its original orientation with the planar wire positioned on the selected face. The wire feature is created only where it extends beyond the faces of the part; a wire feature cannot overlap a face.

Additional information

- *What is feature-based modeling?*
- *Adding a wire feature*
- *The Sketch module*

Adding a point-to-point wire feature

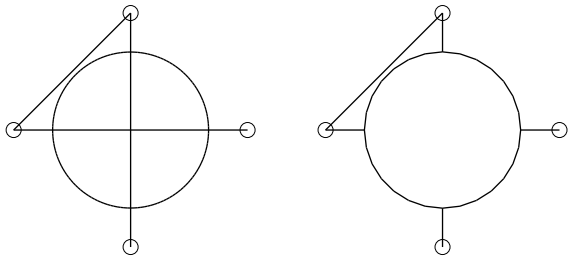
You can add a point-to-point wire feature to a part or to an assembly.
For assembly-level wire features, you can modify the wire feature or remove wires from the feature, as described in *Creating or modifying wire features for multiple connectors*.

Assembly-level wire features are nonmeshable. The wire feature contains wires connecting points from the part or assembly in the current viewport or connecting points from the part or assembly to ground. To model connectors you must define one or more wire features on your assembly. When you create a wire feature on a part or an assembly, you can create a geometry set that includes the wires defined in the wire feature. In addition, when you create a wire feature on a part, you can create geometry sets that include the vertices defined in the wire feature.

To add a wire feature on a part, select **Shape->Wire->Point to Point** from the main menu bar in the Part module. The point-to-point wire tool is always available in the Part module, regardless of the modeling space of the part in the current viewport. To add a wire feature on an assembly, select **Connector->Geometry->Create Wire Feature** from the main menu bar in the Interaction module.

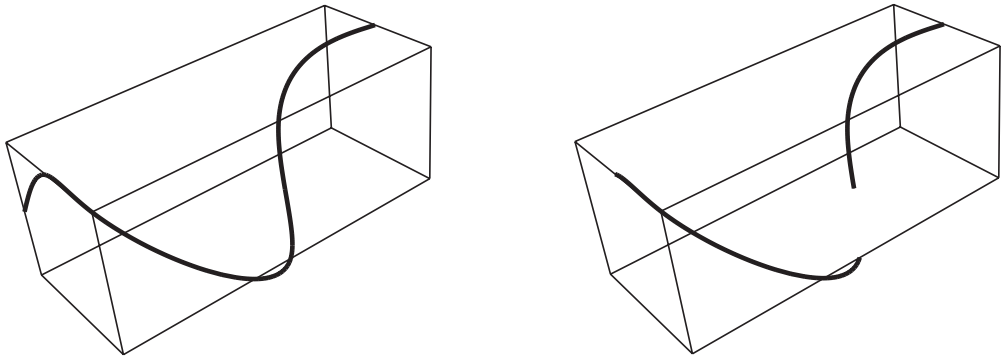
You add a point-to-point wire feature to a part by choosing a geometry type, polyline, or spline; for assemblies you can add only a polyline wire feature. You can choose to imprint the wire on the existing part by creating edges, merge the wire with the existing part, or create the wire separate from the existing part. The wire merge options are available only in the Part module.

If you are creating polyline wires, you must next choose a point selection method and pick points to connect from the current part or assembly. You can choose to select disjointed points (that is, not automatically connected end-to-end), chained points (that is, automatically connected end-to-end), or points that are connected to ground. Abaqus/CAE connects the point pairs with a straight line or connects the points to ground depending on the method that you select. A polyline wire feature connecting four points in a part is illustrated in the following figure:



The planar shell feature is shown for reference. The image on the left shows the full length of the point-to-point wire using the **Imprint wire** or **Separate wire** options, while the image on the right shows a point-to-point wire connecting the same set of points using the **Merge wire** option.

If you are creating spline wires on a part, the chained point selection method is the only one available. Abaqus/CAE calculates the shape of the curve by using a cubic spline fit between all points along the spline; in addition, the first and second derivatives of the spline are continuous. A spline wire feature is illustrated in the following figure:



The rectangular solid feature is shown for reference. The image on the left shows the full length of the spline wire using the **Imprint wire** or **Separate wire** options, while the image on the right shows a spline wire connecting the same set of points using the **Merge wire** option.

During creation of the wire feature on a part or assembly, you can modify the point selections. For part-level wire features, in most cases, once you have created the wire feature, you cannot modify it directly. You might be able to remove a wire edge from the feature using the Geometry Edit toolset. (For more information, see *Removing wire edges*). If you want to change the points that are connected or the connection order, you must delete the wire and create a new wire connecting the desired points. Since the point-to-point wire is dependent on points created by other features, you can modify the wire by using the Feature Manipulation toolset to modify the features that created the points.

Although you cannot create a part with a nonplanar wire base feature, you can create a nonplanar point-to-point wire feature by using a single point in space as the base feature and entering coordinates for the remaining points. In this case the starting point is the only point that you can edit to modify the wire. You can also use datum points, in which case you can edit all points.

1. Display the **Create Wire Feature** dialog box using one of the following methods:

   - From the main menu bar in the Part module, select **Shape->Wire->Point to Point**.

     Tip: You can also add a point-to-point wire feature to a part using the tool, located with the wire tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   - From the main menu bar in the Interaction module, select **Connector->Geometry->Create Wire Feature**.

     Tip: You can also add a wire feature to an assembly using the tool, located in the Interaction module toolbox. For a diagram of the tools in the Interaction module toolbox, see *Using the Interaction module toolbox*.

2. If you are creating a wire in the Part module, choose **Polyline** to create one or more straight lines or choose **Spline** to create a continuous spline curve.
3. If you are adding a wire feature to a part, specify a merge option in the **Wire Merge Scheme** portion of the dialog box. The wire merge options are available only in the Part module.

   - Choose **Imprint wire** to imprint the newly created wire on the existing part by creating edges.
   - Choose **Merge wire** to merge the newly created wire with the existing part.

- Choose **Separate wire** to create the wire separate from the existing part; no edges are created and the wire is not merged with the existing part.

4. In the **Point Pairs** portion of the dialog box, specify the point selection method.

   - Choose **Disjoint wires** to select points that are not automatically connected end-to-end. Use this method to specify wires to use for modeling connectors (see *Modeling connectors*).

     The first two points that you select become **Point 1** and **Point 2**, respectively, of a point pair; the next two points that you select become **Point 1** and **Point 2**, respectively, of the next point pair; and so on. When you use connectors to model multi-point constraints between two points, the motion of **Point 2** is constrained to the motion of **Point 1**.
   - Choose **Chained wires** to select points that are automatically connected end-to-end. The first point that you select becomes **Point 1** of a point pair, the second point that you select becomes **Point 2** of that point pair and **Point 1** of the next point pair, and so on.

     For spline wires, **Chained wires** is the only selection method available, and points are displayed separately instead of in pairs.
   - Choose **Wires to ground** to select points that are connected to ground. Use this method to specify point-to-ground wires to use for modeling connectors (see *Modeling connectors*). Abaqus/CAE automatically makes each point that you select **Point 2** of a point pair (that is, **Point 2** is connected to ground). However, you might want to connect **Point 1** to ground. If so, you can modify the wire definition after you complete the point selection. Select the row of the point pair that you want to modify, and click **Swap** to exchange the entries for **Point 1** and **Point 2** (as described in Step 5).

5. In the **Point Pairs** portion of the dialog box, click  to select the points that the wire will connect.

   - If you are adding a wire feature to a part, you can select points from the viewport or you can enter the coordinates in the text box in the prompt area. Press **Enter** to accept the coordinates and proceed to the next point in the chain of wires. Abaqus/CAE highlights all the points that you can pick. The possible choices are:

     - Vertices
     - The midpoints of lines and arcs
     - The centers of circles and arcs
     - Datum points

   - If you are adding a wire feature to an assembly, you can select points from the viewport. Abaqus/CAE highlights all the points that you can pick. The possible choices are:

     - Vertices
     - Orphan nodes
     - Reference points

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: If you are unable to select the desired points, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   As you select points, Abaqus/CAE displays a representation of the completed point-to-point wire based on your current selections highlighted in red.

6. When you have finished selecting points, click **Done** in the prompt area.

The **Create Wire Feature** dialog box reappears. The points that you selected to define the wire are listed in the **Point Pairs** table.

7. From the **Point Pairs** table, you can do the following:

   - To add more point pairs to a polyline or to add more points to a spline, repeat Steps 3 through 5.

     Note: When you add points to a spline, the new points will always extend the existing spline from the last existing point. When you add point pairs to a polyline, they will be connected to the existing polyline only if you reselect one of the existing points.

     For polylines, the added point-to-point wire segments are highlighted in magenta. For splines, the entire spline wire is highlighted in red because its shape depends on the entire set of points, both new and existing.

   - To edit a point, select the point in the table, click , and reselect a point. The selection in the viewport is updated to show the newly edited point.
   - To identify a specific point pair in the viewport, select the desired row. For polylines, the line connecting the selected point pair is highlighted in red. For splines, the symbol for the point on the selected row is highlighted.
   - To remove a point pair from a polyline or a point from a spline, select the desired row and click .
   - To exchange the entries for **Point 1** and **Point 2** in a point pair for polylines, select the desired row and click .

   You can also enter table data from an ASCII file. To enter data from a file, click mouse button 3 while holding the cursor over a cell in the table; then select **Read From File** from the menu that appears. For more information, see *Entering tabular data*.

8. In the **Set Creation** portion of the dialog box, do the following:

   - Toggle on **Create set of wires** if you want Abaqus/CAE to create a geometry set of wires.
   - Toggle on **Create set of vertices** if you want Abaqus/CAE to create a geometry set of the **Point 1** entries and a geometry set of the **Point 2** entries in the wire definition. The **Create set of vertices** option is available only in the Part module.

9. Click **OK** to create the point-to-point wire feature.

   Part-level wire features appear as solid lines in the viewport and appear in the Model Tree in the **Features** container under the part.

   Assembly-level wires features are nonmeshable, appear as dashed lines in the viewport, and appear in the Model Tree in the **Features** container under the assembly.

Additional information

- *What is feature-based modeling?*
- *Adding a wire feature*
- *Creating or modifying wire features for multiple connectors*
- *The Sketch module*

Rounding vertices in a wire

Select **Shape->Wire->Round** from the main menu bar to "round," or fillet, a vertex between two edges in a wire part.

1. From the main menu bar, select **Shape->Wire->Round**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also round a vertex between two edges in a wire part using the  tool, located with the wire tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   Abaqus/CAE prompts you to select the vertices you want to round and displays prompts in the prompt area to guide you through the procedure.

2. Select the vertices to round. You can **[Shift] + Click** additional vertices to add them to your selection, and **[Ctrl] + Click** a selected vertex to unselect it.

   Tip: If you are unable to select the desired vertices, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

3. When you have finished selecting vertices, click **Done** in the prompt area.

   A default radius appears in the prompt area.

4. If necessary, type a new radius in the text field in the prompt area.
5. Click **Done** to accept the new radius and complete the rounding process.

   Abaqus/CAE redraws the part with the selected vertices rounded.

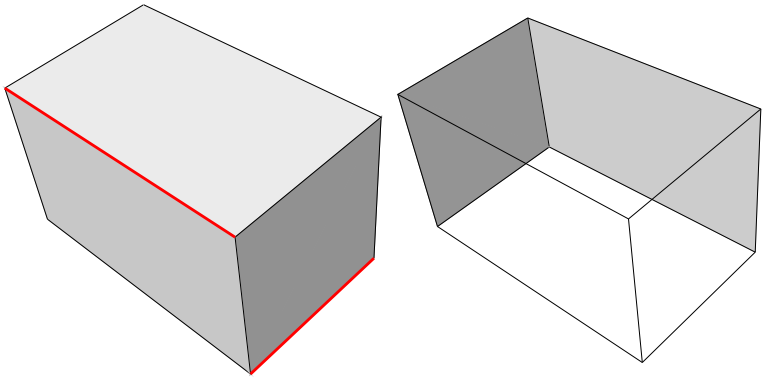
Additional information

- *What is feature-based modeling?*
- *Adding a wire feature*
- *The Sketch module*

Adding a wire-from-edge feature

Select **Shape->Wire->From Edge** from the main menu bar to replace existing 3D shell or solid edges in the viewport with a wire feature.

You add a wire from edge feature by picking one or more edges from the current part. Abaqus/CAE removes the faces along the selected edges, converts the part from a solid to a shell if necessary, and creates wires to replace the removed edges. A wire-from-edge feature is illustrated in the following figure:



The image on the left shows the selected edges on the original solid part. The image on the right shows the resulting part. Each of the selected edges was associated with two faces; Abaqus/CAE removes those faces and converts the solid to a shell. The selected edges, and all the other edges that are no longer associated with faces, make up the new wire feature.

A wire-from-edge feature cannot be modified.

1. From the main menu bar, select **Shape->Wire->From Edge**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a wire-from-edge feature using the tool, located with the wire tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Select the edges to be replaced by wires.
3. Click **Done** to create the wire feature.

   Abaqus/CAE displays a dialog box indicating that the faces and cells associated with the selected edges will be deleted.

4. Click **Yes** to continue or **No** to cancel the procedure.

   If you choose **Yes**, Abaqus/CAE removes the faces associated with the selected edges and adds wires to replace all the removed feature edges.

Additional information

- *What is feature-based modeling?*
- *Adding a wire feature*
- *The Sketch module*

Adding a cut feature

This section describes the Part module tools used to add a cut feature to the part in the current viewport.

In this section:

- *Creating an extruded cut*
- *Creating a loft cut*
- *Creating a revolved cut*
- *Creating a swept cut*
- *Cutting a circular hole*

Creating an extruded cut

Select **Shape->Cut->Extrude** from the main menu bar to create an extruded cut through the part geometry in the current viewport. The extruded cut tool is always available, regardless of the modeling space of the part in the current viewport.

You create an extruded cut into a three-dimensional part by sketching the two-dimensional cross-section of the cut on a selected face and defining the distance through which Abaqus/CAE extrudes the cut. You can select one of the following methods to define the distance through which the cut is extruded:

- **Blind** extends the cut from the sketch plane in a selected direction but only to a specified depth.
- **Up to Face** extends the cut from the sketch plane to a selected face.
- **Through All** extends the cut from the sketch plane in a selected direction through the geometry.

The three methods are illustrated in *Figure 1*.

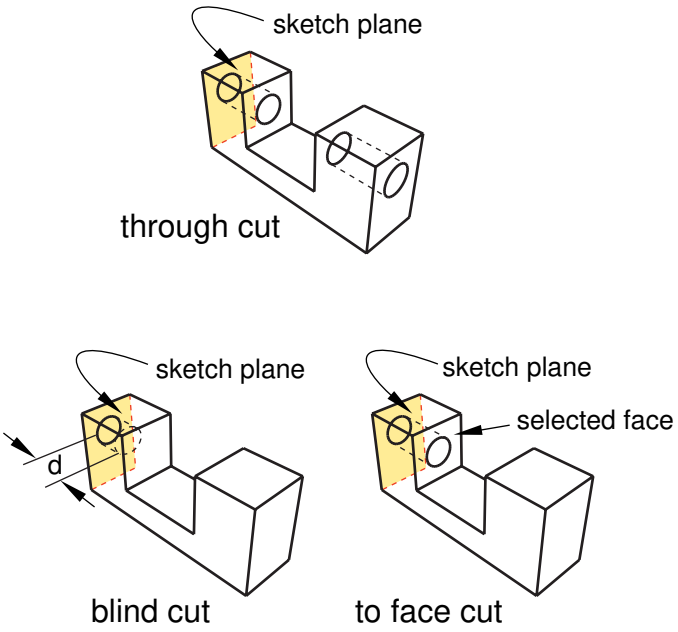


Figure 1: Three methods for creating an extruded cut.

You create an extruded cut in a two-dimensional or axisymmetric planar part by sketching the two-dimensional cross-section of the cut directly on the plane of the part. The cut always passes completely through the part.

When creating an extruded cut in a three-dimensional part, you can select a center point and specify a pitch that Abaqus/CAE uses to twist the cross-section as it is extruded. Alternatively, Abaqus/CAE can expand or contract the cross-section along a specified draft angle as the cross-section is extruded. For more information, see *Including twist in an extrusion*, and *Including draft in an extrusion*.

1. From the main menu bar, select **Shape->Cut->Extrude**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

Tip: You can also create an extruded cut using the tool, located with the cut tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the extruded cut feature. Select one of the following options from the **Sketch Origin** field in the prompt area:
   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. If the current viewport contains a two-dimensional or axisymmetric planar part, Abaqus/CAE enters the Sketcher and you sketch the closed profile of the extruded cut on the plane of the part.

   If the current viewport contains a three-dimensional part, you must do the following:

   a. Select the planar face from which the cut will be extruded. If no suitable face exists, you can select a datum plane or an orphan element face.

      Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

      The selected face is highlighted in the viewport.

   b. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

   c. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

      Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

      Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

      If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

   d. Use the Sketcher to sketch the closed two-dimensional profile of the extruded cut.

4. In the prompt area, click **Done** to indicate you have finished sketching the profile.

5. If the current viewport contains a two-dimensional or axisymmetric planar part, the part returns to its original orientation, and Abaqus/CAE cuts the plane with the sketched profile.

   If the current viewport contains a three-dimensional part, Abaqus/CAE displays the part in its original orientation showing the base part, your sketched profile, and an arrow indicating the extrusion direction. The **Edit Cut** dialog box appears. Complete the following steps to create the extruded cut in the three-dimensional part:

   a. Click  in the **Edit Cut** dialog box to reverse the extrusion direction, if necessary.

      If the arrow direction is difficult to see, use the rotate tool to rotate the part.

   b. Select one of the following end conditions:

- Select **Blind** and enter a value in the **Depth** field to specify the distance through which Abaqus/CAE will extrude the sketched cut profile.
- Select **Up to Face** to specify that Abaqus/CAE will extrude the cut up to a selected face.
- Select **Through All** to specify that Abaqus/CAE will extrude the cut from the sketch plane completely through the geometry.

6. If desired, choose one of the following:
   - Select **Twist** and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched cut profile must include an isolated point that indicates the center of twist.
   - Select **Draft** and enter the draft angle (greater than −90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract.
7. Click **OK** to extrude the profile.

   If you selected the twist option and your sketch includes a single isolated point, Abaqus/CAE uses that point as the center of twist. If your sketch does not include an isolated point, Abaqus/CAE returns to the Sketcher for you to create one. If your sketch contains more than one isolated point, Abaqus/CAE returns to the Sketcher and prompts you to select an isolated point as the center of twist.
8. If you selected **Up to Face**, Abaqus/CAE prompts you to select the face to which to extrude the profile. Select a face to meet the following requirements:
   - the selected face does not have to be parallel to the sketch plane,
   - it can be a nonplanar face,
   - it must completely contain the extruded selection, and
   - it cannot be a datum plane or orphan element face.

   Abaqus/CAE creates the extruded cut feature.



Note:

Cut features are applied to part geometry only. Any orphan elements within the cut region are unaffected by the cut.

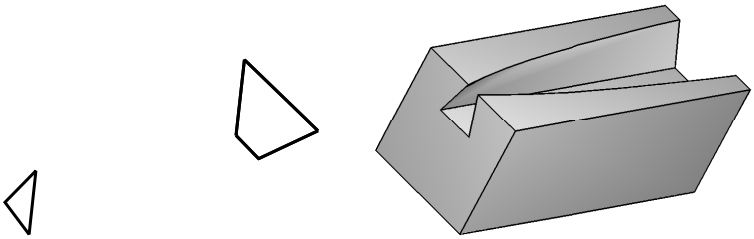
Additional information

- *The Datum toolset*
- *Adding a cut feature*
- *The Sketch module*
- *What is feature-based modeling?*

Creating a loft cut

Select **Shape->Cut->Loft** from the main menu bar to add a loft cut to the part geometry in the current viewport. You can add a loft cut only to three-dimensional parts.

You add a loft cut by creating two or more sections from selected edges and by defining one or more loft paths. Loft sections and the resulting loft cut are illustrated in the following figure (the loft path is not shown in this figure):



You can allow Abaqus/CAE to define a single loft path using a smooth path to connect the center of each loft section (as in the previous figure). If you allow Abaqus/CAE to define the path, you can apply tangency methods to the start and end sections of the loft. The curve and tangencies define the path of the loft feature between sections. Alternatively, you can define one or more loft paths by selecting curves that connect a point on each loft section to a point on the next loft section. Each loft path must provide a continuous line connecting each consecutive loft section. If a loft path is not smooth (if there is more than one tangent to any point along the path), Abaqus/CAE will display an error message when you try to create the loft. For more information about loft sections, loft paths, and loft tangencies, see *What is lofting?*.

Note:

You do not use the Sketcher while adding a loft feature. As a result, all of the edges that define the loft sections and the loft paths must exist in the part geometry before you create the loft. To create a loft path or to create a nonplanar loft section, you can use the tool, located with the wire tools in the Part module toolbox, to create a spline wire (see *Adding a point-to-point wire feature*, for more information).

1. From the main menu bar, select **Shape->Cut->Loft**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also add a loft cut using the tool, located with the cut tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

   The **Edit Loft** dialog box appears.
2. Create the loft sections by selecting edges from the part in the viewport. See *Creating loft sections*, for detailed instructions on creating loft sections.
3. When you have finished creating the loft sections, click on the **Transition** tab in the **Edit Loft** dialog box.
4. Create the loft path (or paths) by again selecting edges from the part in the viewport or by using loft tangencies. See *Creating a loft path*, for detailed instructions on creating a loft path.
5. Click **Preview**.

Abaqus/CAE displays a wireframe representation of the loft that would be created with your current settings.

6. If desired, you can add or remove loft sections, change the loft path definition method, or edit the loft tangency options to change the shape of the loft feature. Click **Preview** to see the effect of your changes in the viewport.
7. If desired, you can have Abaqus/CAE test for self-intersection as it creates the loft feature. This test prevents creating a feature that would be difficult or impossible to mesh and analyze, but it becomes computationally expensive as the complexity of the loft feature increases. For more information, see *Self-intersection checks*. To use self-intersection checks, select **Feature->Options** to open the **Feature Options** dialog box, and toggle on **Perform self-intersection checks**.
8. Click **Done** to create the loft and to close the **Edit Loft** dialog box.

   If you choose to test for self-intersection and the test fails, the **Edit Loft** dialog box will reappear so that you can make changes. Otherwise, the loft cut is created in the viewport.



Note:

Cut features are applied to part geometry only. Any orphan elements within the cut region are unaffected by the cut.

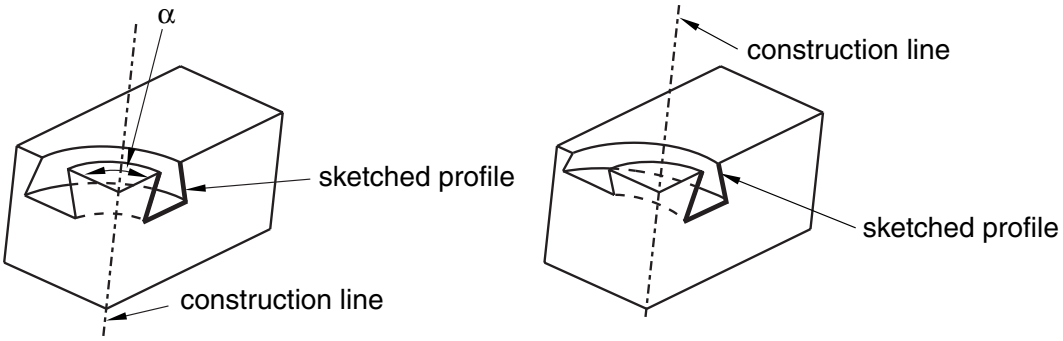
Additional information

- *Adding a cut feature*
- *What is feature-based modeling?*

Creating a revolved cut

Select **Shape->Cut->Revolve** from the main menu bar to create a revolved cut through the part geometry in the current viewport. You can cut a revolved cut only through three-dimensional parts.

You add a revolved cut by sketching a closed two-dimensional cross-section and a construction line on a selected face. The construction line serves as an axis of revolution, and Abaqus/CAE creates the revolved cut by rotating the cross-section about the axis using a specified angle of revolution. In addition, you can specify a pitch and a direction along the axis of revolution that Abaqus/CAE uses to translate the sketch along the axis of revolution as it revolves the profile. Two revolved cut features are shown in the following figure; the figure on the right shows a revolved cut with pitch.



1. From the main menu bar, select **Shape->Cut->Revolve**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a revolved cut using the tool, located with the cut tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If desired, specify the method you want to use to select an origin for your sketch of the revolved cut feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

3. Select the planar face from which the cut will be revolved. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The selected face is highlighted in the viewport.

4. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

5. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If the edge of the selected face is curved or does not provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

6. Use the horizontal , vertical , angle , or oblique  construction line tools to sketch the axis of rotation. You can position the construction line by selecting a datum axis from the underlying part. You cannot select the datum axis directly; you must select a point from either end of the datum axis.
7. Use the Sketcher to sketch the two-dimensional profile of the revolved cut; the sketch must not cross the axis of revolution.
8. In the prompt area, click **Done** to indicate you have finished sketching the profile and the axis. If the sketch contains more than one construction line, Abaqus/CAE prompts you to select the construction line that will serve as the axis of rotation.

   Abaqus/CAE displays the part view that was active prior to entering the Sketcher. The part includes your sketched profile and an arrow indicating the direction of revolution. The **Edit Revolution** dialog box appears.

9. In the **Edit Revolution** dialog box, enter the desired angle of revolution or accept the default value.
10. Click  next to **Revolve direction** to change the arrow direction and the associated direction of revolution.
11. If desired, select the **Include translation** option and enter a positive value for the pitch. The pitch value is the distance through which the profile is translated along the axis of revolution during a rotation of 360°.

    An arrow appears to show the axis of revolution and to indicate the direction of sketch translation along the axis. Click  next to **Pitch direction** in the **Edit Revolution** dialog box to reverse the arrow, if necessary.

12. If desired, toggle on **Sweep sketch normal to path** to rotate the sketched profile normal to the path of revolution. This option is available only when **Include translation** is used.

    The initial profile of the feature will be rotated from the sketch plane to create the feature.

13. Click **OK** to accept the indicated direction and create the revolved cut feature.

    Abaqus/CAE creates the revolved cut feature using your selected parameters.



Note:

Cut features are applied to part geometry only. Any orphan elements within the cut region are unaffected by the cut.

Additional information

- *Creating construction geometry*
- *Adding a cut feature*
- *Defining the axis of revolution for axisymmetric parts and for revolved features*
- *Controlling the cross-section of a revolved feature with pitch*
- *The Sketch module*
- *What is feature-based modeling?*

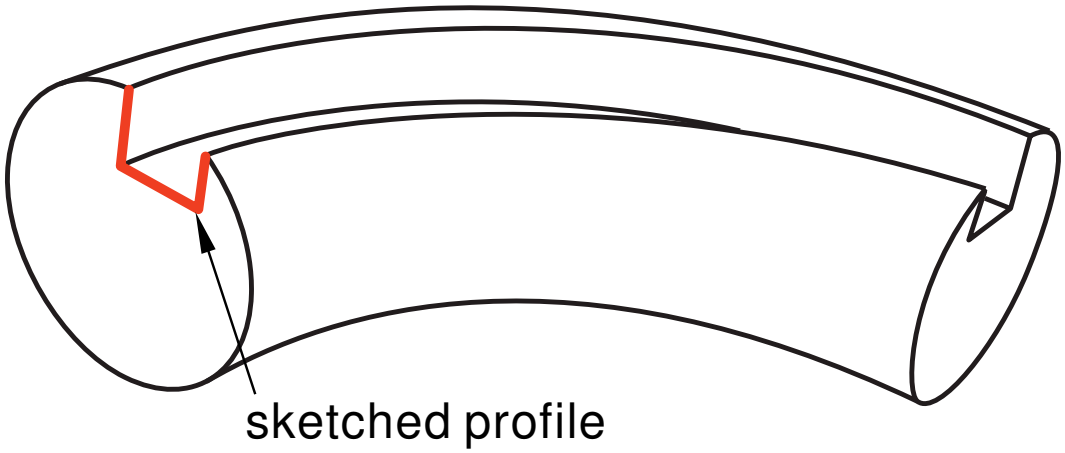
Creating a swept cut

Select **Shape->Cut->Sweep** from the main menu bar to create a swept shell cut through the part geometry in the current viewport. You can create a swept cut through only three–dimensional parts.

You add a swept cut feature by first defining a sweep path, then defining a sweep profile. There are different methods available for defining each component:

- You can define a sweep path either by sketching the path on a selected face or by selecting a series of edges or wires that you want the sweep path to follow. The sketch method provides greater flexibility but supports only two-dimensional paths. The edge/wire method enables you to define a three-dimensional sweep path along a feature such as a spline wire or a set of edges in a three-dimensional part.
- You can define a sweep profile either by sketching a sweep profile using the Sketcher or by selecting one of the faces in your model as the profile. The sweep profile is initially perpendicular to the path; you can keep this orientation constant along the entire sweep path, or you can keep the sweep profile perpendicular to the sweep path as it is swept along its length.

The following figure shows an example of sketching a sweep path on a selected face and sketching a closed sweep profile. The top edge of the profile is not shown.



The sketch or set of edges defining the sweep path and the sketch, edge, or edges defining the sweep profile define the swept cut feature; both the sweep path and the sweep profile can be modified using the Feature Manipulation toolset if you define them using the Sketcher. You can define a swept cut whose sweep profile is offset from the sweep path. In this case Abaqus/CAE moves the sweep path to a parallel location that passes through the sweep profile and creates the swept cut feature at that location.

As you define your sweep feature, you can apply a twist or draft to it. For more information about these tools, see *What types of features can you create?*. You can toggle on **Keep profile normal constant** to maintain the same orientation for the sweep profile as it travels along the sweep path; if this option is toggled off, the sweep profile orientation changes with the normal to the sweep path. In addition, you can toggle on **Keep internal boundaries** in the Feature Manipulation toolset to maintain any faces or edges that are generated between the swept shell feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning.

1. From the main menu bar, select **Shape->Cut->Sweep**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.



Tip: You can also create a swept cut using the  tool, located with the cut tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If you want to sketch the sweep path, do the following:

   a. From the **Path** options, select **Sketch** and click .

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. If desired, specify the method you want to use to select an origin for your sketch of the swept solid feature. Select one of the following options from the **Sketch Origin** field in the prompt area:

   - Select **Auto-Calculate** to place the sketch origin automatically.
   - Select **Specify** to define a custom sketch origin.
   - Select **Session Default** to use the custom origin you specified earlier in the session.

   c. Select the planar face on which to sketch the sweep path. If no suitable face exists, you can select a datum plane or an orphan element face.

   Tip: If you are unable to select the desired planar face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   d. If you selected **Specify** as the **Sketch Origin** method, specify the origin location by clicking a point in the viewport or by entering the three-dimensional coordinates of the origin in the prompt area. You can also set this custom origin as the default origin for all sketches in your session by toggling on **Set as session default**.

   e. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the selected face. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If there is no straight edge with the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part so that the selected face aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation.

   If you are unsure of the part's orientation relative to the Sketcher grid, use the view manipulation tools from the View Manipulation toolbar to view its position. Use the reset view tool  to return to the original view.

   f. Sketch the sweep path. The sweep path must meet the following guidelines:

   - The path can be closed, but the ends must meet smoothly; for example, the ends should not meet at a corner. For examples of valid sweep paths, see *Defining the sweep path and the sweep profile*.
   - The path must be continuous; for example, it must not branch.
   - The resulting solid cannot intersect with itself.

   In the prompt area, click **Done** to indicate you have finished sketching the sweep path.

Abaqus/CAE exits the Sketcher and restores the original view of the part. A highlighted line indicates the sweep path and its direction. Abaqus/CAE also reopens the **Create Cut Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Path** options to indicate that the sweep path has been defined in the Sketcher.

3. If you want to specify the sweep path as a series of edges or wires, do the following:

   a. From the **Path** options, select **Edges** and click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. If desired, specify in the prompt area whether you want to select the edges in your sweep path **individually** or **by edge angle**. For more information about selecting objects, see *Using the angle and feature edge method to select multiple objects*.

   c. Select the edges you want to include in the sweep path.

      Abaqus/CAE shows the sweep path on your part and indicates the sweep direction.

   d. In the prompt area, click **Yes** to confirm the sweep path direction or click **Flip** to reverse the sweep path direction.

   Abaqus/CAE reopens the **Create Cut Sweep** dialog box and adds the word **(Defined)** next to the **Edges** label in the **Path** options to indicate that the sweep path has been defined using a series of edges.

4. If you want to sketch the sweep profile, do the following:

   a. From the **Profile** options, select **Sketch** and click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. Sketch the sweep profile. The sweep profile must meet the following guidelines:

      - The profile must be closed.
      - The resulting solid cannot intersect with itself.

      You can sketch the profile anywhere on the Sketcher grid; Abaqus/CAE sweeps the profile along a path parallel to the sweep path. In the prompt area, click **Done** to indicate you have finished sketching the sweep profile.

      Abaqus/CAE exits the Sketcher and restores the original view of the part.

   c. Select an edge and the orientation of the edge on the Sketcher grid. The selected edge must not be parallel to the sweep path direction. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

      Abaqus/CAE highlights the selected edge, enters the Sketcher again, and rotates the part so that the Sketcher grid lies on a plane normal to the beginning of the sweep path with the sweep path direction pointing out of the screen. In addition, the selected edge aligns with the grid in the desired orientation. The intersection of two dashed lines indicates the origin of the sweep path. Abaqus/CAE also reopens the **Create Cut Sweep** dialog box and adds the word **(Defined)** next to the **Sketch** label in the **Profile** options to indicate that the sweep profile has been defined in the Sketcher.

5. If you want to select a face as the sweep profile, do the following:

   a. From the **Profile** options, select **Face**, then click .

      Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   b. Select a face from the viewport.

Abaqus/CAE highlights the selected face and reopens the **Create Solid Sweep** dialog box with the word **(Defined)** added next to the **Faces** label in the **Profile** options to indicate that the sweep profile has been defined using a face.

6. If desired, do any of the following:
   - Toggle on **Include twist**, and enter the pitch. The pitch is the extrusion distance in which a 360° twist would occur. The sketched extrusion profile must include an isolated point that indicates the center of twist.
   - Toggle on **Include draft**, and enter the draft angle (greater than –90° and less than 90°). A positive draft angle indicates that external faces of the profile expand and internal faces contract. You cannot apply a draft if the **Keep profile normal constant** option is selected.
   - Toggle on **Keep profile normal constant** to maintain the same profile orientation along the entire sweep path. If this option is toggled off, Abaqus/CAE adjusts the profile orientation so that the angle between the sweep profile and the normal to the sweep path is always constant. You cannot toggle on this option if the **Include draft** option is selected.
   - Toggle on **Keep internal boundaries** to maintain any faces or edges that are generated between the swept solid feature and the existing part. The internal boundaries can create regions that can be structured or swept meshed without having to resort to partitioning.
7. Click **OK** to create the new swept cut.



Note:

Cut features are applied to part geometry only. Any orphan elements within the cut region are unaffected by the cut.

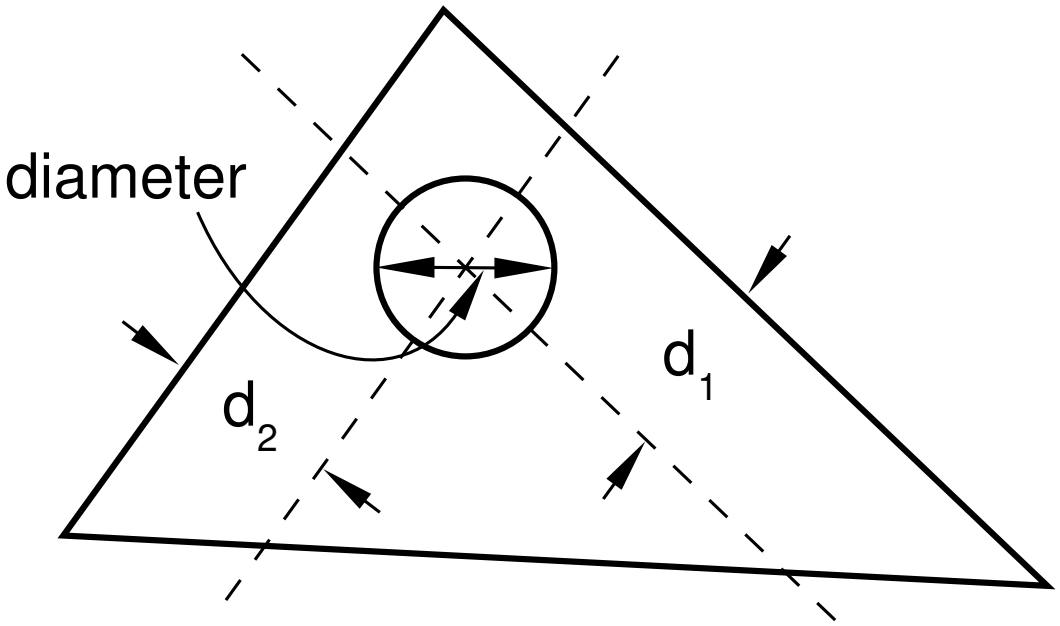
Additional information

- *Defining the sweep path and the sweep profile*
- *The Sketch module*
- *What is feature-based modeling?*

Cutting a circular hole

Select **Shape->Cut->Circular Hole** from the main menu bar to cut a circular hole through the part in the current viewport. The circular hole tool is always available, regardless of the modeling space of the part in the current viewport.

You cut a circular hole by specifying the distance from two selected straight edges and specifying the diameter of the hole, as shown in the following figure:



The part must contain at least two straight edges; for example, you cannot use this tool to cut a hole through a circular part.

If the current viewport contains a two-dimensional or axisymmetric planar part, the hole always passes through all of the part. However, if the current viewport contains a three–dimensional part, Abaqus/CAE prompts you to select the type of cut. You can select either **Through All** or **Blind** to define the cut depth.

The distance from the hole to each edge, the diameter of the hole, and the depth of a blind hole are the features that define a circular hole, and all three can be modified using the Feature Manipulation toolset. You cannot change the type of cut—through all or blind—after the cut has been created.

1. From the main menu bar, select **Shape->Cut->Circular Hole**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

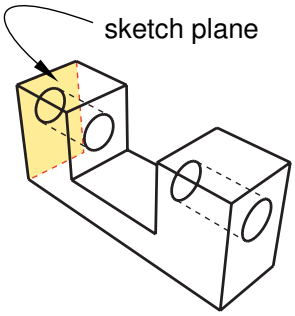
   Tip: You can also cut a circular hole using the tool, located with the cut tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. If the current viewport contains a two-dimensional part, select the first edge from which to position the center of the hole.

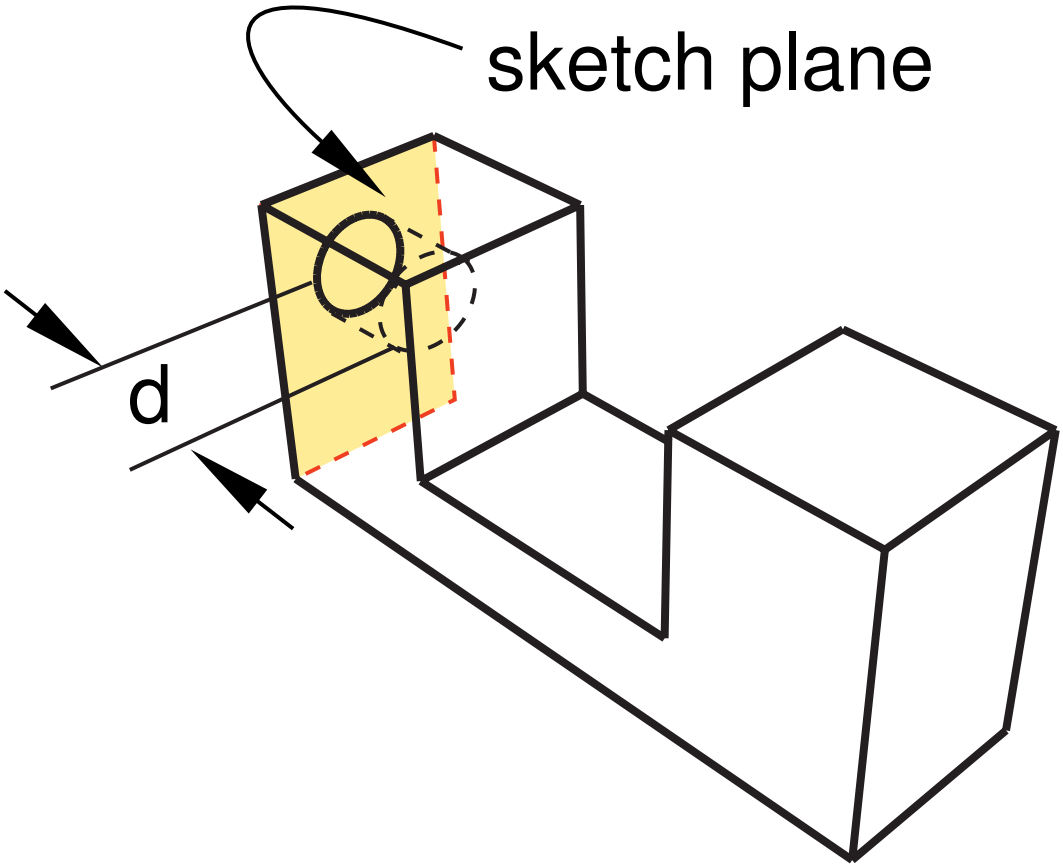
   If the current viewport contains a three-dimensional part, you must do the following:

   a. From the buttons in the prompt area, select one of the following types of cut:

- Click **Through All** to cut a circular hole that extends from a selected face in a selected direction through the part. A through cut is illustrated in the following example:



- Click **Blind** to cut a circular hole that extends from a selected face in a selected direction but only to a specified depth. A blind cut is illustrated in the following example:



b. Select the face from which the hole will be cut.

Tip: If you are unable to select the desired face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

An arrow appears, indicating the direction of the axis of the cut hole.

c. From the buttons in the prompt area, click **Flip** to reverse the arrow, if necessary. Click **OK** to accept the indicated direction.



Tip: If the arrow direction is difficult to see, use the rotate tool to rotate the part.

d. Select the first edge from which to position the center of the hole. The selected edges need not lie in the same plane as the selected face, but they must not be perpendicular to it.

3. In the text field in the prompt area, type the distance from the selected edge to the center of the hole.
4. Select the second edge from which to position the center of the hole. The two edges must not be parallel.
5. In the text field in the prompt area, type the distance from the selected edge to the center of the hole.
6. In the text field in the prompt area, type the diameter of the hole.

   If the current viewport contains a two−dimensional or axisymmetric planar part, Abaqus/CAE cuts the part with the circular hole.

   If the current viewport contains a three−dimensional part and you selected a blind cut, a default hole depth appears in the prompt area. Click mouse button 2 to accept the default value, or enter a new hole depth.

   The part returns to its original orientation with the circular hole cut from the selected face.



Note:

Cut features are applied to part geometry only. Any orphan elements within the cut region are unaffected by the cut.

Additional information

- *Adding a cut feature*
- *The Sketch module*
- *What is feature-based modeling?*

Using the Edit Feature dialog box

When you first select a feature to edit, the **Edit Feature** dialog box appears.

This section describes the options in the **Edit Feature** dialog box; the available options depend on the feature selected. You use the **Edit Feature** dialog box to change the following:

Depth

Use the **Depth** parameter to change the depth of a blind extruded feature. Type a new value for the extrusion depth, and click **Apply** to see the modified feature in the viewport. You cannot change the manner in which the extrusion distance was defined when the feature was first created; for example, from **Blind** to **Up to Face**. For more information, see *Defining the extrusion distance*.

Radius

Use the **Radius** parameter to change the radius of a round/fillet feature. Type a new value for the fillet radius, and click **Apply** to see the modified feature in the viewport. For more information, see *Rounding edges*, and *Rounding vertices in a wire*.

Flip extrude direction

Toggle **Flip extrude direction** to change the extrusion direction of the selected feature. Click **Apply** to see the modified feature in the viewport. This option is available only for extruded features added to a base feature. For more information, see *Controlling the direction of an extruded feature*.

Keep internal boundaries

Toggle **Keep internal boundaries** to maintain any faces or edges that are generated between the feature and the existing part. The internal boundaries may create regions that can be structured or swept meshed without having to resort to partitioning. This option is available only for extruded, revolved, and loft solid and shell features added to a base feature.

Draft angle

Use the **Draft angle** parameter to change the draft of an extruded feature. Type a new value for the draft angle, and click **Apply** to see the modified feature in the viewport. A draft angle of zero will create a straight extrusion. You cannot add draft to an extrusion if it was not defined when the feature was created. For more information, see *Including draft in an extrusion*.

Pitch

Use the **Pitch** parameter to change the twist of an extruded feature or the pitch of a revolved feature. Type a new value for the pitch, and click **Apply** to see the modified feature in the viewport. You cannot add pitch to a feature if it was not defined when the feature was created.

In an extruded feature the pitch defines the extrusion distance in which the profile would be twisted by 360°. For more information, see *Including twist in an extrusion*. Similarly, in a revolved feature the pitch is the distance through which the profile would be translated during a rotation of 360°. For more information, see *Defining the axis of revolution for axisymmetric parts and for revolved features*.

Angle

Use the **Angle** parameter to change the angle of revolution of a revolved feature. Type a new value, and click **Apply** to see the modified feature in the viewport.

Flip revolve direction

Toggle **Flip revolve direction** to reverse the direction of revolution of a revolved feature. Click **Apply** to see the modified feature in the viewport. For more information, see *Controlling the direction of a revolved feature*.

Flip pitch direction

Toggle **Flip pitch direction** to change the direction of translation in a revolved feature with pitch. Click **Apply** to see the modified feature in the viewport. This option is available only for revolved features and works in conjunction with the **Pitch** parameter. For more information, see *Controlling the direction of a revolved feature*.

Sweep sketch normal to path

Toggle **Sweep sketch normal to path** to change the orientation of the initial sketch that is used to create a revolved feature with pitch. Click **Apply** to see the modified feature in the viewport. This option is available only for revolved features with pitch. For more information, see *Controlling the cross-section of a revolved feature with pitch*.

Edit Section Sketch

Use the **Edit Section Sketch** button to change the profile of the selected feature. Make your changes, close the Sketcher, and click **Apply** to see the modified feature in the viewport.

Edit Sweep Path Sketch

Use the **Edit Sweep Path Sketch** button to change the path of a swept feature. Make your changes, close the Sketcher, and click **Apply** to see the modified feature in the viewport. For more information, see *Defining the sweep path and the sweep profile*.

Stitch tolerance

Use the **Stitch tolerance** parameter to increase or decrease the size of the gaps that Abaqus/CAE fixes when you stitch the gaps between free edges in a part. For more information, see *Stitching edges to create faces*.

Regenerate on OK

Toggle **Regenerate on OK** to control feature regeneration. **Regenerate on OK** is toggled on by default. Toggle this option off if you do not want the feature to regenerate when you click **OK**. For more information, see *Regenerating a part or assembly*.

Additional information

- *How is a part defined in Abaqus/CAE?*
- *Creating a new part*
- *Using the Create Part dialog box*

Using the Edit Loft dialog box

This section describes the procedures used to create the components of a solid, shell, or cut loft feature.

In this section:

- *Creating loft sections*
- *Creating a loft path*

Creating loft sections

Loft sections represent the shape that the loft feature will have at a particular point along the loft path. At least two sections are required to create a loft feature. You create loft sections by picking from existing edges from the part in the current viewport.

1. From the **Edit Loft** dialog box, select **Insert Before** or **Insert After** to create the first loft section.
2. Select the edges that define the loft section. **[Shift] + Click** additional edges to add them to your selection, and **[Ctrl] + Click** a selected edge to unselect it. Selected edges are highlighted in red and will remain highlighted until the loft feature is completed.

   A loft section can be comprised of any edges from the current part and must meet the following conditions:

   - For a solid loft or cut loft the section must be closed, and the first and last sections must be planar.
   - For a shell loft if one section is open, all must be open; and, if one section is closed, all must be closed.
   - The section must be one continuous loop with no branches.
   - Edges used in one loft section cannot be used in another section of the same loft.

   All shell loft sections, and intermediate sections for solid lofts and cut lofts, can be nonplanar. To create nonplanar sections, you can select edges that were created using a spline wire or select edges of an existing nonplanar face.
3. In the prompt area, click **Done** to indicate that you have finished selecting edges to define the loft section.

   Abaqus/CAE returns to the **Edit Loft** dialog box, and the section appears in the **Sections** list.
4. Repeat Steps 2 and 3 to define all the loft sections. Use the **Insert Before** and **Insert After** buttons to indicate where in the list you want the next loft section to appear in relation to the loft section that is currently highlighted. The section that is highlighted in the list is also highlighted in magenta in the viewport to distinguish it from the other loft sections. Loft sections are numbered in the order that you create them, but Abaqus/CAE creates the loft feature in the order shown in the **Sections** list. You can change the order of the list only by adding or deleting loft sections.

   You cannot edit completed loft sections.

Additional information

- *What is lofting?*
- *Creating a loft path*
- *Adding a solid loft feature*
- *Adding a shell loft feature*
- *Creating a loft cut*

Creating a loft path

The paths of a loft feature connect a point on the starting section to a point on the ending section. If more than two loft sections are defined, the paths must also pass through a point on each intermediate section. Each loft path must provide a smooth curve (only one tangent line must exist at any point along the path) connecting all the loft sections in the same order that they will be connected when you create the loft feature.

1. From the **Transition** tabbed page in the **Edit Loft** dialog box, choose either of the following methods to create a loft path:

   - **Specify tangencies**. Abaqus/CAE will create a single loft path using a smooth curve connecting the center of each loft section.
   - **Select path**. Select edges to define a loft path.

   You cannot edit the loft path method after the loft feature is created.

2. If you chose **Specify tangencies** to define the loft path, you can choose a **Start Tangency** and an **End Tangency** for the loft feature. You can choose from the following loft tangency options:

   - **None** is the default and the only setting that, if selected, must be applied at both ends of the loft.
   - **Normal**. Loft faces are at a 90° angle near the starting or ending loft section.
   - **Radial**. Loft faces are at a 0° angle near the starting or ending loft section.
   - **Specify**. You specify an angle ranging from 0° to 180° that is applied to the loft faces, and you specify a magnitude between 0 and 100 percent that determines the relative distance through which the angle is applied.

   The tangency effect diminishes with distance and does not affect the loft feature at intermediate sections. For more information and examples of loft tangency, see *Defining loft tangency*.

   You cannot edit the tangency after the loft feature is created.

3. If you chose **Select path**, you must have created at least one suitable smooth curve in the part before starting the loft procedure. Select each loft path as follows:

   a. Click **Add**.

   The **Edit Loft** dialog box disappears.

   b. Select edges to define the loft path. Use **[Shift] + Click** to add additional edges to your selection and **[Ctrl] + Click** to unselect a selected edge. Selected edges are highlighted in blue and will remain highlighted until Abaqus/CAE creates the loft feature.

   You must create a smooth curve connecting one point on each loft section to a point on the next loft section following the same order as the loft section list. Abaqus/CAE will create the loft feature by following each loft path in the order of the loft section list.

   c. Click **Done**.

   The **Edit Loft** dialog box reappears.

   d. If desired, use the preceding steps to add more loft paths. To successfully create a loft feature with multiple loft paths, the paths must not share any points.

   Click **Delete** to remove a highlighted path from the list.

   e. Toggle off **Global Smoothing** if you want the effects of the paths to be localized to adjacent surfaces; Abaqus/CAE will apply simple paths to any edges of the loft for which you did not define a path. If you leave global smoothing on, Abaqus/CAE will apply an averaging function to create complex paths for edges where you did not define a path.

Loft paths cannot be added or deleted after the loft feature is created; loft paths can be edited only by moving the vertices used to create each path.

Additional information

- *What is lofting?*
- *Creating loft sections*
- *Adding a solid loft feature*
- *Adding a shell loft feature*
- *Creating a loft cut*

Blending edges

This section describes the Part module tools used to blend edges of the part in the current viewport.

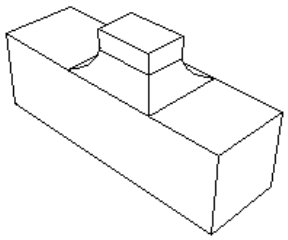
In this section:

- *Rounding edges*
- *Chamfering edges*

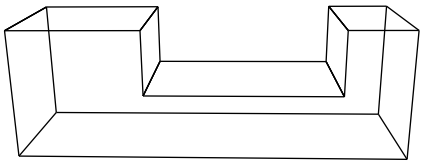
Rounding edges

You can round selected edges of the part in the current viewport.

Select **Shape->Blend->Round/Fillet** from the main menu bar to round selected edges of the part in the current viewport. You can "round," or fillet, both convex and concave edges. A part with rounded edges is illustrated in the following example:



You define the radius of the fillet, and Abaqus/CAE applies the radius to all of the selected edges as a group; therefore, subsequent feature manipulation operations, such as edit, delete, and suppress, will be applied to the entire group of selected edges. Consequently, if you select more than one edge to round, you cannot modify just one of the rounded edges. In addition, the shape of the resulting edges can depend on the order in which you apply the fillets, as shown in the following figure. The fillets on the left side of the part were created by selecting all three edges and applying the round/fillet tool to the group of selected edges in a single operation. In contrast, the fillets on the right side of the part were created by selecting each edge individually and applying the round/fillet tool to each edge in sequence.



The round/fillet tool is available only when the current viewport contains a three-dimensional solid or shell part. In addition, you cannot fillet an edge that contains a wire. The radius of a rounded edge defines the feature and can be modified using the Feature Manipulation toolset.

1. From the main menu bar, select **Shape->Blend->Round/Fillet**.

   Abaqus/CAE prompts you to select the edges to round.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also round selected edges using the tool, located with the blend tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Select the edges to round, and click mouse button 2 to commit your selection. **[Shift] + Click** additional edges to add them to your selection, and **[Ctrl] + Click** a selected edge to unselect it.



Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

A default radius appears in the prompt area.

3. If necessary, type a new radius in the text field in the prompt area. Click mouse button 2 to commit the radius.

   Abaqus/CAE redraws the part with the selected edges rounded.

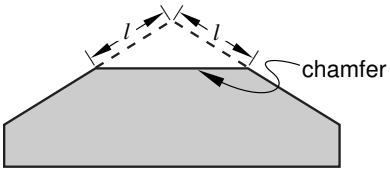
Additional information

- *What is feature-based modeling?*

Chamfering edges

You can chamfer or bevel selected edges of the part in the current viewport.

Select **Shape->Blend->Chamfer** from the main menu bar to chamfer or bevel selected edges of the part in the current viewport. You enter the distance that the chamfer extends into each face, and Abaqus/CAE uses the distance to define the chamfer, as illustrated in the following example:



Abaqus/CAE applies the chamfer to all of the selected edges as a group; therefore, subsequent feature manipulation operations—such as edit, delete, and suppress—will be applied to the entire group of selected edges. Consequently, if you select more than one edge to chamfer, you cannot modify just one of the chamfered edges.

The chamfer tool is available only when the current viewport contains a three−dimensional solid or shell part. In addition, you cannot chamfer an edge that contains a wire. The length of a chamfer defines the feature and can be modified using the Feature Manipulation toolset.

1. From the main menu bar, select **Shape->Blend->Chamfer**.

   Abaqus/CAE prompts you to select the edges to chamfer.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also chamfer selected edges using the tool, located with the blend tools in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. Select the edges to chamfer, and click mouse button 2 to commit your selection. **[Shift] + Click** additional edges to add them to your selection, and **[Ctrl] + Click** a selected edge to unselect it.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   A default chamfer length appears in the prompt area.

3. If necessary, type a new chamfer length in the text field in the prompt area. Click mouse button 2 to commit the chamfer length.

   Abaqus/CAE redraws the part with the selected edges chamfered.

Additional information

- *What is feature-based modeling?*

Mirroring a part

Select **Shape->Transform->Mirror** from the main menu bar to transform the part in the current viewport into a mirror image of itself. This operation is similar to copying a part and using the mirror option (*Copying a part*). However, transforming the original part retains the complete feature creation history and the ability to edit those features. You can choose to keep both the original geometry and the mirrored geometry, and you can select any planar face or datum plane as the mirror plane.

The mirror tool is available only when the current viewport contains a three-dimensional solid or shell part. Mirror features, like all features, can be deleted, suppressed, and resumed; however, a mirror feature contains no modifiable parameters, so it cannot be edited after you create it.

1. From the main menu bar, select **Shape->Transform->Mirror**.

   Tip: You can also mirror a part using the  tool, located in the Part module toolbox. For a diagram of the tools in the Part module toolbox, see *Using the Part module toolbox*.

2. From the prompt area, toggle on **Keep original geometry** to retain the original geometry when the part is mirrored.
3. From the prompt area, toggle on **Keep internal boundaries** to retain any intersecting boundaries between the original geometry and the new geometry when the part is mirrored.

   This option has no effect if used without **Keep original geometry**.
4. Select a datum plane or planar face from the viewport.

   Abaqus/CAE creates the mirrored geometry using the options you chose.

Additional information

- *What is feature-based modeling?*

The Property module

You can use the Property module to define materials, sections, and composite layups as well as other properties of a part or part region.

You can use the Property module to perform the following tasks:

- Define materials.
- Define beam section profiles.
- Define sections.
- Assign sections, orientations, normals, and tangents to parts.
- Define composite layups.
- Define a skin reinforcement.
- Define inertia (point mass, rotary inertia, and heat capacitance) on a part.
- Define springs and dashpots between two points or between a point and ground.
- Define material calibrations.

For information on defining inertia, see *Inertia*. For information on defining skin reinforcements, see *Skin and stringer reinforcements*. For information on defining springs and dashpots, see *Springs and dashpots*.

In this section:

- *Entering and exiting the Property module*
- *Understanding properties*
- *Which properties can I assign to a part?*
- *Understanding the Property module editors*
- *Using material libraries*
- *Using the Property module toolbox*
- *Creating and editing materials*
- *Defining general material data*
- *Defining mechanical material models*
- *Defining thermal material models*
- *Defining electrical and magnetic material models*
- *Defining other types of material models*
- *Creating and editing sections*
- *Creating and editing composite layups*
- *Assigning sections, orientations, normals, and tangents to a part*
- *Using discrete orientations for material orientations and composite layup orientations*
- *Creating material calibrations*
- *Using the Special menu in the Property module*
- *Using the Query toolset to obtain assignment information*

Entering and exiting the Property module

You can enter the Property module at any time during an Abaqus/CAE session by clicking **Property** in the **Module** list located in the context bar. When you enter the Property module, **Material**, **Section**, **Profile**, **Assign**, **Special**, **Feature**, and **Tools** menus appear in the main menu bar. A **Part** list appears in the context bar that allows you to select the part to which you want to assign properties.

To exit the Property module, select another module from the **Module** list. You need not take any specific action to save your material, section, and other definitions before exiting the module; they are saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

Additional information

- *Using the Special menu in the Property module*

Understanding properties

You can specify the properties of a part or part region by creating a section and assigning it to the part. In most cases, sections refer to materials that you have defined. Beam sections also refer to profiles that you have defined. This section of the guide explains materials, profiles, sections, rebar, and section assignment. You create materials, profiles, and sections using the Property module editors, as described in *Understanding the Property module editors*.

In this section:

- *Defining materials*
- *Defining profiles*
- *Defining sections*
- *Defining composite layups*
- *Understanding rebar in shell sections*

Defining materials

A material definition specifies all the property data relevant to a material. You specify a material definition by including a set of material behaviors, and you supply the property data with each material behavior you include. You use the material editor to specify all the information that defines each material.

Each material that you create is assigned its own name and is independent of any particular section; you can refer to a single material in as many sections as necessary. Abaqus/CAE assigns the properties of a material to a region of a part when you assign a section referring to that material to the region.

Defining profiles

A profile specifies the properties of a beam section that are related to its cross-sectional shape and size (for example, cross-section area and moments of inertia).
When you define a beam section, you must include a reference to a profile in the section definition.

You can create the following types of profiles:

Shape-based profiles

Shape-based profiles define the specific shape and dimensions of the beam cross-section. Abaqus uses the information provided by the shape-based profile to calculate the engineering properties of the section.

You can create this type of profile by first selecting from a list of shape options and then specifying that particular shape's dimensions. For example, if you select a box shape, you must then specify the height and width of the box as well as the thickness of the four walls. You can select from the following shape options:

- Arbitrary
- Box
- Circular
- Hexagonal
- I, L, T
- Pipe
- Rectangular
- Trapezoidal
- Channel
- Hat

The shape options are shown in *Figure 1*.

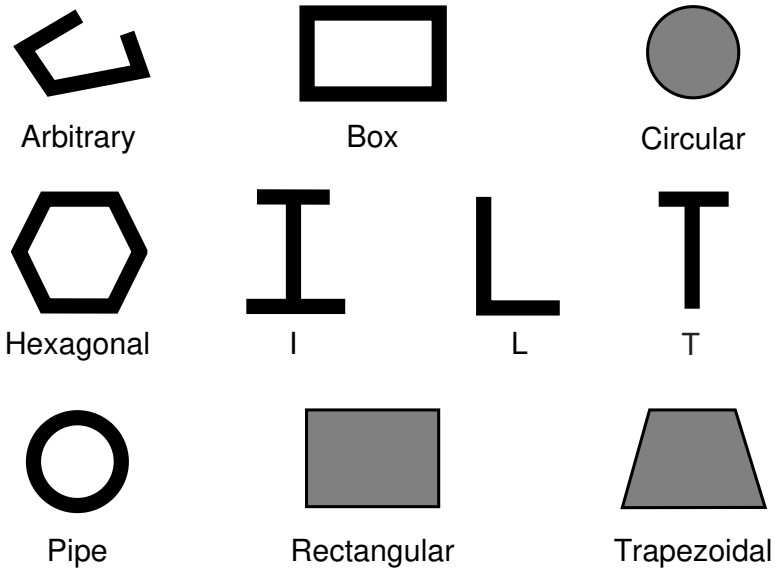


Figure 1: Available shape options.

For detailed information on each profile shape, see *Beam Cross-Section Library*.

Generalized profiles

Generalized profiles specify the engineering properties of the section directly. You can create a generalized profile by specifying values for the area, moments of inertia, torsional constant, and, if applicable, sectorial moment and warping constant. For more information, see *Using a General Beam Section to Define the Section Behavior*.

Each profile that you create has its own name and is independent of any particular beam section; you can refer to a single profile in as many beam sections as necessary. After you have assigned the beam section and beam orientation to the part, you can use the part display options to view an idealized representation of the shape-based or generalized beam profile. Displaying beam profiles is useful for checking that the correct profile has been assigned to a particular region and that the assigned beam orientation results in the expected orientation of the profile. For more information, see *Controlling beam profile display*.

Defining sections

A section contains information about the properties of a part or a region of a part. The information required in the definition of a section depends on the type of region in question. For example, if the region is a deformable wire, shell, or two-dimensional solid, you must assign a section to that region that provides information about the region's cross-sectional geometry. Likewise, a rigid region requires a section that describes its mass properties. Most sections must refer to a material name. Beam sections must also refer to a profile name.

When you assign a section to a part, Abaqus/CAE automatically assigns that section to each instance of the part. As a result, the elements that are created when you mesh those part instances will have the properties specified in that section.

Sections are named and created independently of any particular region, part, or assembly. You can assign a single section to as many different regions as necessary. You can use the Property module to create solid sections, shell sections, beam sections, fluid sections, and other sections.

Solid sections

Solid sections define the section properties of two-dimensional, three-dimensional, and axisymmetric solid regions.

- **Homogeneous solid sections**. Homogeneous solid sections consist of a material name. In addition, if the section will be used with a two-dimensional region, you must also specify the section thickness. (You have the option of specifying a plane stress or plane strain thickness even if the section will be assigned to a three-dimensional region. Abaqus/CAE ignores the thickness information if it is not needed for the region type.)

  For more information, see *Creating homogeneous solid sections*.
- **Generalized plane strain sections**. Generalized plane strain sections consist of a material name, thickness, and wedge angles about the global 1- and 2-axes. You can assign generalized plane strain sections only to two-dimensional planar regions.

  For more information, see *Creating generalized plane strain sections*.
- **Eulerian sections**. Eulerian sections consist of a list of material names. This list specifies all of the materials that can be present in an Eulerian domain. You can assign Eulerian sections only to Eulerian parts.

  For more information, see *Creating Eulerian sections*. For an overview of Eulerian analyses, see *Eulerian analyses*.
- **Composite solid sections**. Composite solid sections consist of layers of materials. For each layer of material, you must specify a material name, thickness, and orientation.

  For more information, see *Creating composite solid sections*.
- **Electromagnetic solid sections**. Electromagnetic solid sections are valid for electromagnetic models and consist of a material name. In addition, if the section will be used with a two-dimensional region, you must also specify the section thickness. (You have the option of specifying a plane stress or plane strain thickness even if the section will be assigned to a three-dimensional region. Abaqus/CAE ignores the thickness information if it is not needed for the region type.)

  For more information, see *Creating electromagnetic solid sections*.

Shell sections

Shell sections define the section properties of shell regions. Shells model structures in which one dimension (the thickness) is significantly smaller than the other two dimensions and in which the stresses in the thickness

direction are negligible. You can define one or more layers of reinforcement (rebar) in shell sections. For more information, see *Understanding rebar in shell sections*.

- **Homogeneous shell sections**. Homogeneous shell sections consist of a shell thickness, material name, section Poisson's ratio, and optional rebar layers. You can choose to provide the section property data before the analysis or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis. If the latter is chosen, options are provided to control the section integration and temperature variation through the thickness.

  For more information, see *Creating homogeneous shell sections*.
- **Composite shell sections**. Composite shell sections consist of layers of materials, a section Poisson's ratio, and optional rebar layers. For each layer of material, you must specify a material name, thickness, and orientation. You can choose to provide the section property data before the analysis or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis. If the latter is chosen, options are provided to control the section integration and temperature variation through the thickness.

  For more information, see *Creating composite shell sections*.
- **Membrane sections**. Membranes represent thin surfaces in space that offer strength in the plane of the surface but have no bending stiffness. Membrane sections consist of a material name, membrane thickness, section Poisson's ratio, and optional rebar layers.

  For more information, see *Creating membrane sections*.
- **Surface sections**. Surface sections represent surfaces in space that have no inherent stiffness and behave like membrane elements with zero thickness. Surface sections consist of optional rebar layers.

  For more information, see *Creating surface sections*.
- **General shell stiffness sections**. General shell stiffness sections allow you to define a shell's mechanical response by directly specifying the stiffness matrix and thermal expansion response. General shell stiffness sections consist of a section stiffness matrix and scaling moduli. Optionally, you can also specify a thermal expansion coefficient and thermal stresses in the section.

  For more information, see *Creating general shell stiffness sections*.

Beam sections

Beams are used in two and three dimensions to model slender, rod-like structures that provide axial strength and bending stiffness. Beams represent structures in which the cross-section is assumed to be small compared to the length. You can assign beam sections only to wire regions. In addition, you must assign a beam section orientation to all regions with beam sections.

- **Beam sections**. Beam sections consist of a section Poisson's ratio and a reference to a profile. Additional information is required depending on whether you choose to calculate (integrate) the section stiffness either before or during analysis.

  For information on profiles, see *Defining profiles*. For more information on beam sections, see *Creating beam sections*.
- **Truss sections**. Trusses, like beams, are used in two and three dimensions to model slender, rod-like structures that provide axial strength but no bending stiffness. Truss sections consist of a material name and cross-sectional area.

  For more information, see *Creating truss sections*.

You can use the part display options to view an idealized representation of the beam or truss profile along the wire region. For more information, see *Controlling beam profile display*.

Other sections

Other sections you can create include gasket sections, cohesive sections, acoustic infinite sections, and acoustic interface sections.

- **Gasket sections** (Abaqus/Standard analyses only). Gaskets model thin sealing components that are positioned between structural components. Gasket sections are used to provide pressure-closure behaviors for sealing components. Gasket sections consist of a material name, initial gasket thickness, initial gap, initial void, and cross-sectional area.

  For more information, see *Creating gasket sections* and *Gaskets*.

- **Cohesive sections**. Cohesive sections are used to model finite thickness adhesives, negligibly thin adhesive layers for debonding applications, as well as gaskets. No specialized gasket behavior (typically defined in terms of pressure versus closure) is available. Cohesive sections consist of a material name, response, initial thickness, and out-of-plane thickness.

  For more information, see *Creating cohesive sections* and *Adhesive joints and bonded interfaces*.

- **Acoustic infinite sections**. Acoustic infinite sections are used to model an acoustic medium undergoing small pressure changes involving exterior domains. Acoustic infinite sections consist of an acoustic medium material name. In addition, if the section will be used with a two-dimensional region, you must also specify the section thickness. (You have the option of specifying a plane stress or plane strain thickness even if the section will be assigned to a three-dimensional region. Abaqus/CAE ignores the thickness information if it is not needed for the region type.)

  For more information, see *Creating acoustic infinite sections*.

- **Acoustic interface sections**. Acoustic interface sections are used to couple an acoustic medium to a structural model. Acoustic interface sections consist of an acoustic medium material name. In addition, if the section will be used with a two-dimensional region, you must also specify the section thickness. (You have the option of specifying a plane stress or plane strain thickness even if the section will be assigned to a three-dimensional region. Abaqus/CAE ignores the thickness information if it is not needed for the region type.)

  For more information, see *Creating acoustic interface sections*.



Warning:

The type of section that you assign to a part must be consistent with the element type that you assign to instances of that part in the Mesh module. For example, if you assign a truss section to a wire part in the Property module, you should assign a truss element type (and not a beam element type) to any instances of that part in the Mesh module.

Defining composite layups

You use a composite layup to model a part that contains many plies, where each ply is defined by a material, a thickness, and a reference orientation. Composite layups are similar to composite shell or composite solid sections. A ply in a composite section is the same as a ply in a composite layup; however, a composite section always contains the same number of plies. In contrast, a composite layup can contain a different number of plies in different regions. Abaqus/CAE converts a composite layup to its constituent composite sections when you analyze your model. Abaqus/CAE allows you to define three types of composite layups—shells, continuum shells, and solids. For more information, see *Composite layups*.

Understanding rebar in shell sections

You can define one or more layers of reinforcement (rebar) in shell sections by specifying a unique layer name for each rebar layer. You also select the name of the material forming each rebar layer and specify the cross-sectional area per bar, spacing, and orientation of the rebar in each layer.

To define the orientation of each rebar layer, you can specify an orientation angle or an orientation name. The angular orientation of a rebar layer is defined relative to the rebar reference orientation. You use the **Assign** menu to assign a rebar reference orientation to shell regions. If you specify an orientation name, you must supply the user subroutine *ORIENT*. For more information, see *Defining rebar layers*, and *Assigning a rebar reference orientation*.

In the Step module you must request output for rebar to include rebar output in the data that Abaqus writes to the output database and to view plots of the rebar orientations in the Visualization module. In the Visualization moduleAbaqus/CAE treats rebar layers as section points for output purposes, and you can create material orientation plots to show the rebar orientation.

For more information on rebar, see *Defining Reinforcement*.

Additional information

- *Understanding output requests*
- *Selecting section point data by category*
- *Plotting material orientations*

Which properties can I assign to a part?

Once you have created a section, you can assign the following properties to a part:

Section

You can assign the section to a region of a part. The **Section Assignment Manager** allows you to view, create, edit, suppress, resume, and delete section assignments. In the Property module, Abaqus/CAE colors a region green to indicate that the region has a section assignment. If there are overlapping section assignments, Abaqus/CAE colors the region yellow.

Beam Section Orientation

You can assign beam section orientations to wire regions. You assign an orientation to a beam section by defining the approximate local 1-direction of the cross-section.

Material Orientation

You can assign material orientations to shell and solid regions. The global coordinate system determines the default material orientation. You can define a material orientation by selecting an existing datum coordinate system or discrete field or by defining a discrete orientation. For an Abaqus/Standard analysis, you can define the material orientation in a user subroutine.

Rebar Reference Orientation

The angular orientation of a rebar layer is defined relative to the rebar reference orientation. You can assign rebar reference orientations to shell regions. The global coordinate system determines the default rebar reference orientation. You can define a rebar reference orientation by selecting an existing datum coordinate system from the viewport and then selecting an axis on the datum coordinate system that approximates the direction of the shell normal.

Element Normal

You can assign shell/membrane normal directions to orphan elements, shell and membrane regions, and axisymmetric parts with wire regions. The shell/membrane normals affect the material orientation assigned to the region. If you reverse the normal of a shell region, the material 2-direction will be reversed. The reversal of the material 2-direction has no effect on the analysis results. However, you should use care when interpreting section point output for shells.

Element Tangent

You can assign beam/truss tangent directions to orphan elements and wire regions. Beam section orientations depend on the beam tangent directions. If you reverse the tangent direction, the local 2-direction will be reversed, and you should use care when interpreting results, in particular when you identify the beam section point locations.

You can use the **Assign** menu in the Property module main menu bar to assign properties to a part. You can select the region to which to assign a property in the following ways:

- Select the region directly in the viewport.
- Select elements individually or using the angle method (to assign shell normals to orphan elements).
- Use the Set toolset to create a set consisting of regions of parts or orphan elements. (The Set toolset is available from the **Tools** menu in the main menu bar.) You can then assign the property to the region or elements defined by the set.

If you assign a section to a region and then rename or delete the section, that section is no longer applied to the region. If a region of your model lacks section properties, your analysis job will fail and the problem will be reported by the Job module.

However, the original names of renamed or deleted sections continue to be associated with the regions to which they have been assigned until you take one of the following actions:

- Assign a different section to the region.
- Create a new section that has the original section name and is the appropriate type for the region (for example, a shell section for a shell region); the properties defined in the new section are applied to the region automatically.
- If you have renamed a section, change the name of the section back to its original name.

(You can use the Query toolset to determine the name of the section assigned to the region; for more information, see *Understanding the role of the Query toolset*.)

Similarly, if you refer to a material in a section definition and then rename or delete the material, the section becomes invalid; properties defined in that section are no longer applied to regions to which the section is assigned. However, the original names of renamed or deleted materials continue to be associated with sections that refer to those materials; therefore, you can use techniques similar to the ones listed above to restore sections.

For detailed instructions on assigning properties to a model and managing section assignments, see the following sections:

- *Assigning a section*
- *Managing section assignments*
- *Assigning a beam orientation*
- *Assigning a material orientation or rebar reference orientation*
- *Assigning shell/membrane normal directions*
- *Assigning beam/truss tangent directions*
- *Using discrete orientations for material orientations and composite layup orientations*

Understanding the Property module editors

When you create or edit a material, profile, or section, you must enter data in the appropriate editor. For example, when you create a material, you must enter data in the material editor. This section provides information on each editor type.

In this section:

- *Creating materials*
- *Creating profiles*
- *Creating sections*
- *Creating composite layups*
- *Selecting material behaviors*
- *Specifying material parameters and data*
- *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*

Creating materials

To create a material, select **Material->Create** from the main menu bar. An **Edit Material** dialog box appears in which you can enter a name for the material and create or edit material properties. The material editor is shown in *Figure 1*.

Note:

Once you have created a material, it cannot be renamed using the material editor; you must use **Material->Rename** to change the name of an existing material.

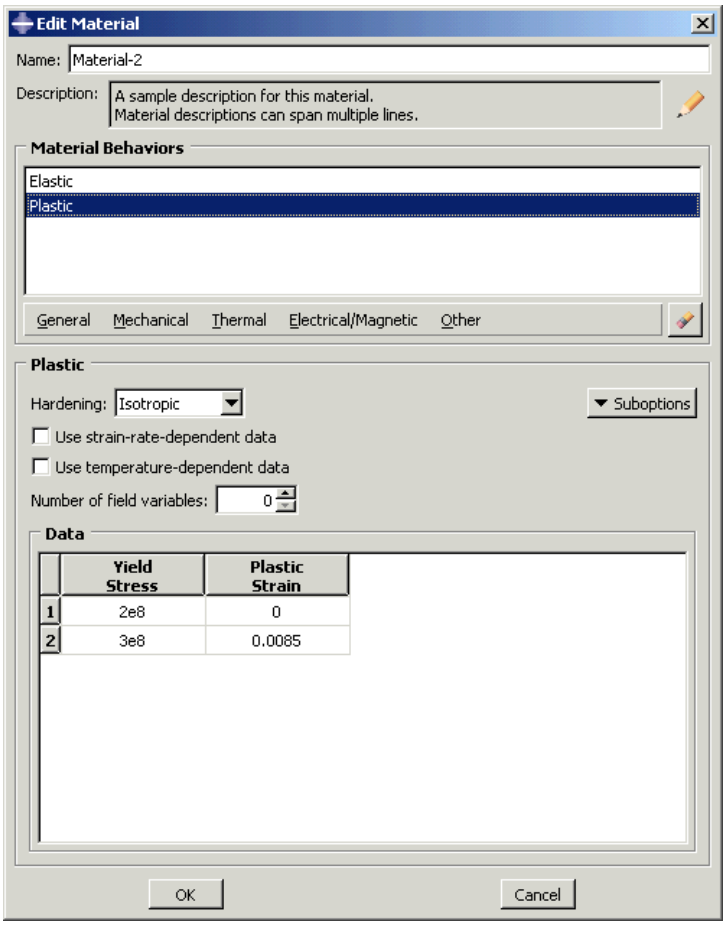


Figure 1: The material editor.

The material editor consists of the following:

Material Behaviors list

A list of the behaviors you have included in the material definition.

Behavior menu

A set of menus beneath the behavior list from which you select material behaviors.

Behavior definition area

The lower portion of the window in which the parameters, tabular data fields, and suboptions associated with a selected behavior appear.



Note:

You can display help on particular aspects of the editor that are not discussed here by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest.

Additional information

- *Creating or editing a material*
- *Browsing and modifying material behaviors*
- *Specifying material parameters and data*

Creating profiles

To create a profile, select **Profile->Create** from the main menu bar. A **Create Profile** dialog box appears in which you can enter a name for the profile and select the profile type. Once you have finished entering this information, click **Continue** in the **Create Profile** dialog box to display the profile editor, which allows you to create and edit profiles.

All profile editors display a diagram of the profile shape and text fields in which you can enter all of the data necessary to define the profile. For example, the I-shaped profile editor is shown in *Figure 1*. The editor contains a diagram of the I-shaped profile and data fields in which you can enter each dimension.

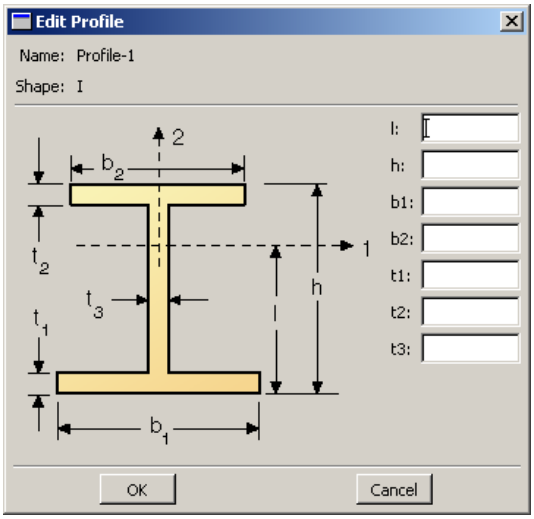


Figure 1: The I-shaped profile editor.

Once you have created a profile, you can refer to that profile in a beam section definition. For example, a box-shaped profile named `SupportBeam` is selected in the beam section editor shown in *Figure 2*.

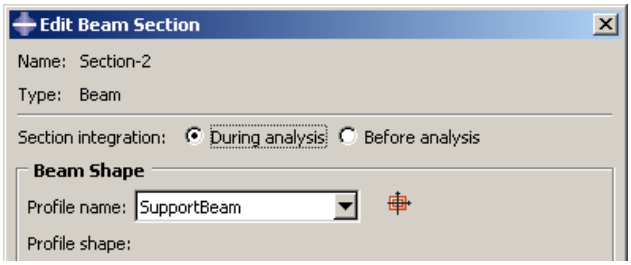


Figure 2: Specifying a profile name in the beam section editor.

For more information on profiles, see *Defining profiles*.

Additional information

- *Defining profiles*
- *Beam Cross-Section Library*

Creating sections

You can use the Property module to create the following types of sections:

- Homogeneous solid sections
- Generalized plane strain sections
- Eulerian sections
- Composite solid sections
- Electromagnetic solid sections
- Homogeneous shell sections
- Composite shell sections
- Membrane sections
- Surface sections
- General shell stiffness sections
- Beam sections
- Truss sections
- Gasket sections
- Cohesive sections
- Acoustic infinite sections
- Acoustic interface sections

To create a section, select **Section->Create** from the main menu bar. A **Create Section** dialog box appears in which you can name the section and specify the type of section that you want to create. Once you have specified a section name and type, click **Continue** in the **Create Section** dialog box to display the section editor, which allows you to create and edit sections.

The format of the section editor varies according to the type of section you are defining. For example, the homogeneous shell section editor is shown in *Figure 1*.

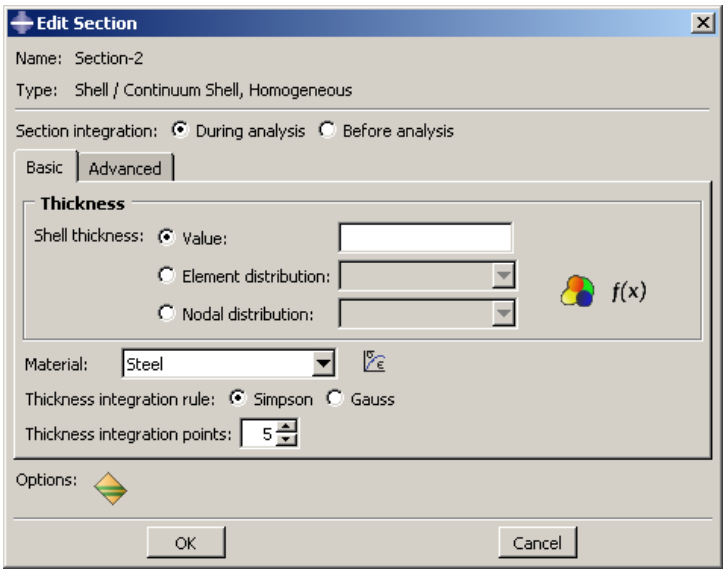


Figure 1: The homogeneous shell section editor.

Note:

You can display help on particular aspects of an editor that are not discussed here by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest. A help window will appear containing a relevant section from this guide.

Some editors contain a **Rebar Layers** option (icon), as shown in *Figure 1*. If you click this icon, another dialog box appears in which you can enter data concerning rebar layers, as shown in *Figure 2*.

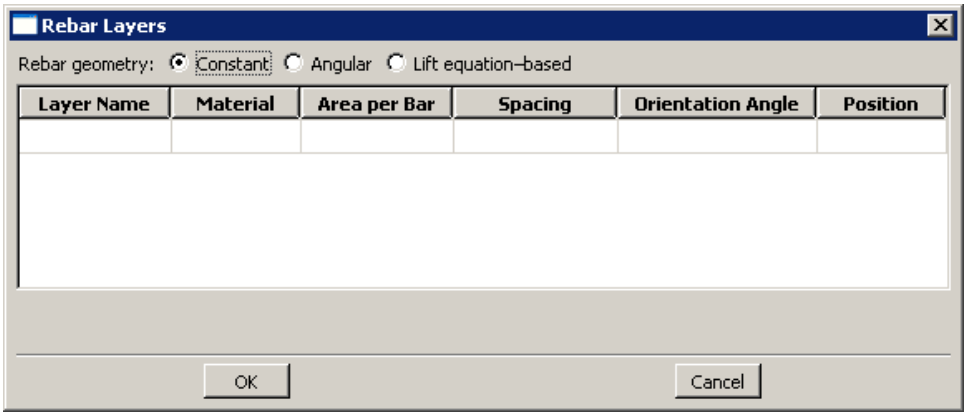


Figure 2: The Rebar Layers dialog box.

Note:

To display context-sensitive help for items in the **Rebar Layers** dialog box, you must select the item of interest and then press **[F1]**. (The **Help** menu in the main menu bar is unavailable while the option dialog box is displayed.)

Once you have entered all the data necessary to define the section, you can click **OK** to close the section editor and to save the section.

For detailed instructions on using section editors, see the following sections :

- *Creating homogeneous solid sections*
- *Creating generalized plane strain sections*
- *Creating Eulerian sections*
- *Creating composite solid sections*
- *Creating electromagnetic solid sections*
- *Creating homogeneous shell sections*
- *Creating composite shell sections*
- *Creating membrane sections*
- *Creating surface sections*
- *Creating general shell stiffness sections*
- *Creating beam sections*
- *Creating truss sections*
- *Creating gasket sections*
- *Creating cohesive sections*

- *Creating acoustic infinite sections*
- *Creating acoustic interface sections*
- *Defining rebar layers*
- *Creating profiles*

Additional information

• *Defining sections*
• *Creating and editing sections*

Creating composite layups

You can use the Property module to create the following types of composite layups:

- Shell
- Continuum shell
- Solid

To create a composite layup, select **Composite->Create** from the main menu bar. A **Create Composite Layup** dialog box appears in which you can name the layup, specify the initial ply count, and specify the type of composite layup that you want to create. Once you have finished entering this information, click **Continue** in the **Create Composite Layup** dialog box to display the composite layup editor, which allows you to create and edit layups.

The format of the composite layup editor varies according to the type of layup you are defining. For example, the shell composite layup editor is shown in *Figure 1*.

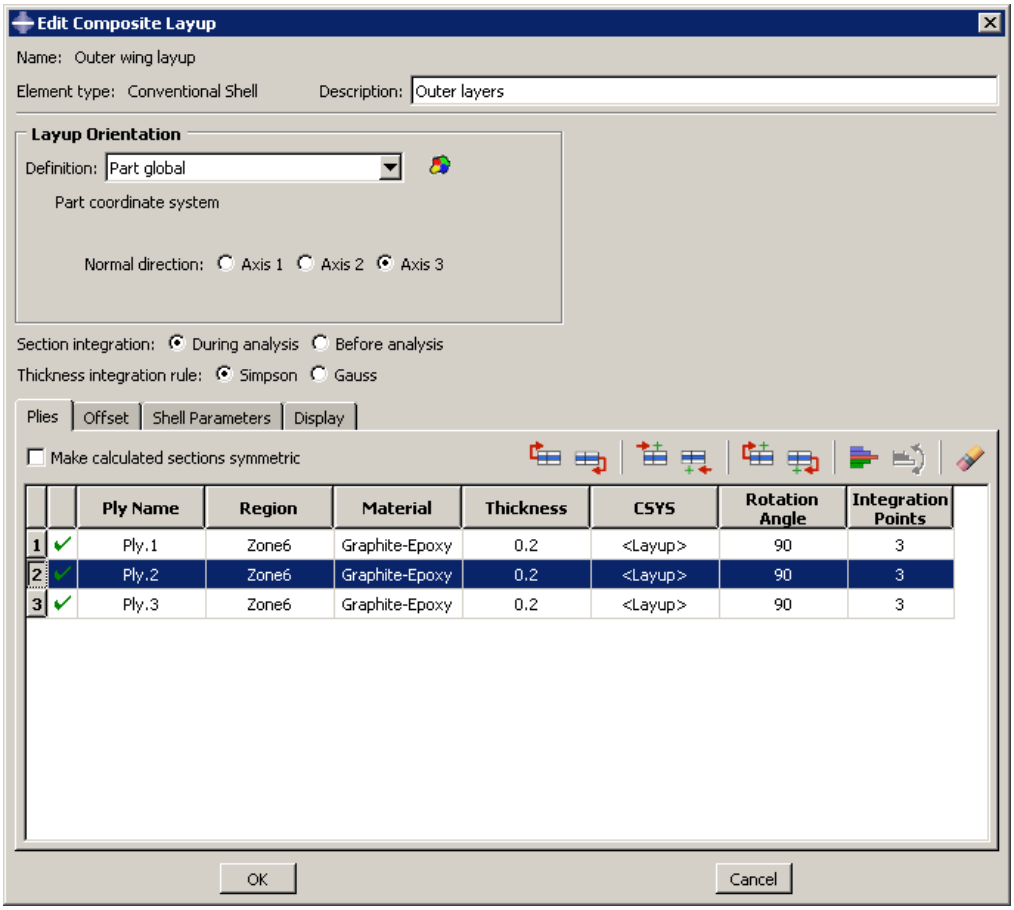


Figure 1: The shell composite layup editor.

Note:

You can display help on particular aspects of an editor that are not discussed here by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest. A help window will appear containing a relevant section from this guide.

Once you have entered all the data necessary to define the layup, you can click **OK** to close the editor and to save the composite layup.

For detailed instructions on using composite layup editors, see the following sections:

- *Creating conventional shell composite layups*
- *Creating continuum shell composite layups*
- *Creating solid composite layups*

Additional information

- *Defining composite layups*
- *Creating and editing composite layups*

Selecting material behaviors

The material editor contains several menus that allow you to add most of the material behaviors available in Abaqus/Standard or Abaqus/Explicit to a material definition.
(For information on which material behaviors are available in Abaqus/CAE, see *Abaqus keyword browser table* and *Keyword support from the input file reader*.)

The material editor menus reflect the division of all material behaviors into five categories: **General**, **Mechanical**, **Thermal**, **Electrical/Magnetic**, and **Other**. *Figure 1* shows the elasticity behaviors available under the **Mechanical** menu.

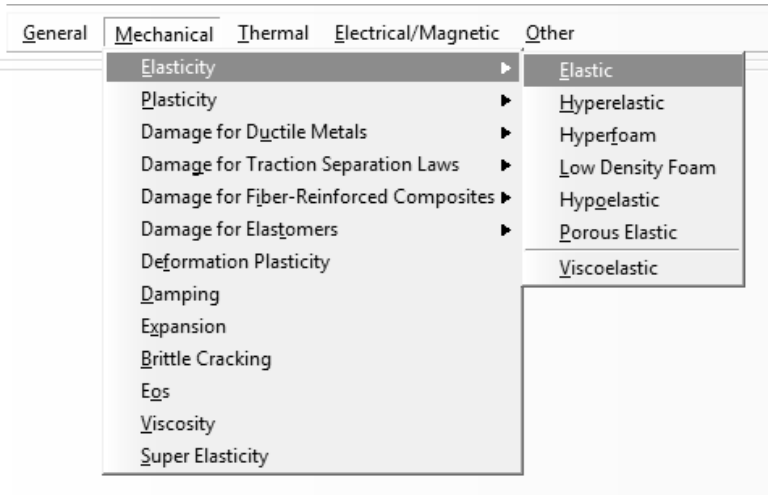


Figure 1: Elasticity behaviors under the Mechanical menu.

The lists of behaviors do not change to exclude behaviors that are invalid for the type of analysis you are running. In addition, Abaqus/CAE does not check that the data that you enter in the editor are valid or that your materials are appropriate for your analysis type. For example, if you request a dynamic analysis, Abaqus/Standard or Abaqus/Explicit requires that you specify the density of the materials used in the model so that it can calculate mass and inertia properties of the model. If you do not provide a material density in the material definition, Abaqus/CAE allows you to create the material; however, Abaqus/CAE will report an error when you submit your analysis job.

When you select a behavior, the name of the behavior appears in the **Material Behaviors** list at the top of the editor, and the behavior becomes part of your material definition. For example, the list in *Figure 2* reflects that the **Elastic** and **Plastic** behaviors have been chosen, as well as the **Fail Stress** suboption of the **Elastic** behavior.

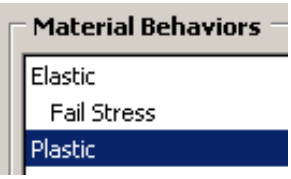


Figure 2: The Material Behaviors list.

Behaviors such as **Elastic** and **Plastic** are primary behaviors. Test data and suboptions such as **Fail Stress** appear beneath the corresponding primary behavior and are indented to indicate their subordinate position.

If you want to remove a behavior or suboption from a material definition, you can select that behavior or suboption from the **Material Behaviors** list and then click .

If you are creating a new material, the selected behavior list is initially blank. As you select behaviors, the behavior name appears in the list; if there are too many behaviors to see at once, a scroll bar appears on the right side of the list.

Additional information

- *Browsing and modifying material behaviors*
- *Specifying material parameters and data*

Specifying material parameters and data

When you select a behavior, the behavior definition area changes to show all of the associated parameters and data items for the selected behavior. The parameters are shown at the top of the behavior description area and the data items at the bottom.

Depending on your analysis requirements, you choose to either accept or change the default parameter values; for example, you choose whether to use isotropic elasticity by using the **Type** combo box on the elasticity form, as shown in *Figure 1*.

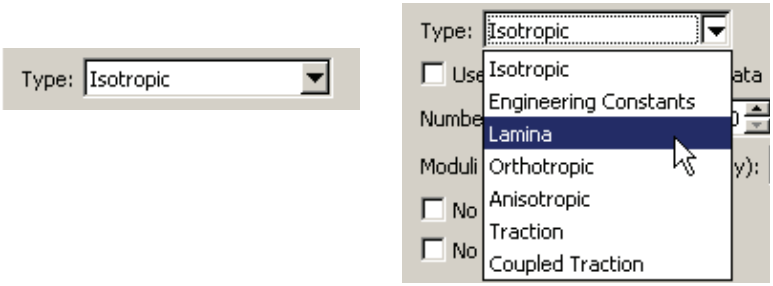


Figure 1: Type combo box.

A table containing fields for the remaining required material data appears beneath the parameter area; for example, *Figure 2* shows the table that appears when you choose isotropic elasticity.

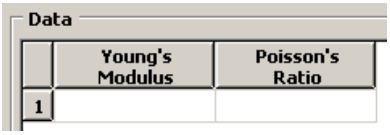


Figure 2: Isotropic elasticity table.

Different fields become available depending on how you have set the parameters. For example, when you choose lamina elasticity rather than isotropic elasticity, the table in *Figure 3* appears.

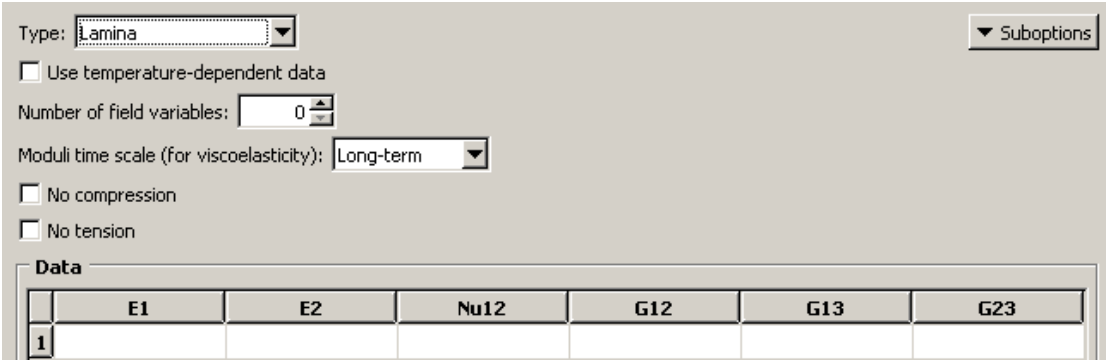


Figure 3: Lamina elasticity table.

You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For example, an option exists for automatically entering data from a file. Another option exists for creating an *X–Y* data object from the data in the table; you can plot the *X–Y* data in the Visualization module and visually check its validity. For detailed information on each option, see *Entering tabular data*.

For detailed information on specific features in the material editor, see the following sections:

- *Creating or editing a material*

- *Browsing and modifying material behaviors*
- *Entering strain-rate-dependent data*
- *Entering temperature-dependent data*
- *Specifying field variable dependence*
- *Selecting and modifying suboptions or test data*
- *Displaying X–Y plots of hyperelastic material behavior*
- *Displaying X–Y plots of viscoelastic material behavior*
- *Displaying X–Y plots of hyperfoam material behavior*

Additional information

• *Browsing and modifying material behaviors*

Evaluating hyperelastic, hyperfoam and viscoelastic material behavior

Abaqus/CAE provides a convenient **Evaluate** option that allows you to view the behavior predicted by a hyperelastic, hyperfoam, or viscoelastic material and that allows you to choose a suitable material formulation.

You can evaluate any hyperelastic or hyperfoam material, but a viscoelastic material can be evaluated and viewed only if it is defined in the time domain and includes hyperelastic, hyperfoam, and/or elastic material data. If your material definition includes viscoelastic data defined in the frequency domain, you cannot evaluate its viscoelastic material behavior in Abaqus/CAE, but its material evaluation data are written to the data (`.dat`) file.

The **Evaluate** option prompts Abaqus/CAE to perform one or more standard tests using an existing material. (For information on standard tests for hyperelastic, hyperfoam, and viscoelastic materials, see *Hyperelasticity*, *Hyperelastic Behavior in Elastomeric Foams*, and *Linear Viscoelasticity*, respectively.) Once the standard tests are completed, Abaqus/CAE enters the Visualization module and displays the test results in new viewports as *X–Y* plots. (For more information on *X–Y* plots, see *X–Y plotting*.) Abaqus/CAE also displays an informational dialog box containing the stability limits and coefficients for each hyperelastic strain energy potential and the viscoelastic material parameters for the viscoelastic response. The information from the evaluation is saved in the *material_name_i*`.dat` file, where *i* starts at `1` and is incremented for subsequent evaluations of the same material. You can review the evaluation results and adjust the material definition as necessary.

To initiate the evaluation procedure, select **Material->Evaluate->*material name*** from the main menu bar. Alternatively, you can select the material of interest in the **Material Manager** and then click **Evaluate**. The **Evaluate Material** dialog box appears in which you can specify how you want Abaqus/CAE to perform the standard tests.

- For detailed instructions on evaluating hyperelastic material behavior, see *Displaying X–Y plots of hyperelastic material behavior*.
- For detailed instructions on evaluating hyperfoam material behavior, see *Displaying X–Y plots of hyperfoam material behavior*.
- For detailed instructions on evaluating viscoelastic material behavior, see *Displaying X–Y plots of viscoelastic material behavior*.



Note:

The material evaluation procedure generates jobs with the same names as the materials; therefore, these material names must adhere to the same rules as job names (see *Using basic dialog box components*, for more information on naming objects).

The **Evaluate** option is particularly useful in the following scenarios:

Comparing test data with the behavior predicted by a particular strain energy potential

When you define a hyperelastic or hyperfoam material using experimental data, you also specify the strain energy potential that you want to apply to the data. Abaqus uses the experimental data to calculate the coefficients necessary for the specified strain energy potential. However, it is important to verify that an acceptable correlation exists between the behavior predicted by the material definition and the experimental data.

You can use the **Evaluate** option to calculate the material response based on the experimental data using the strain energy potential that you have specified in the material definition. When the tests are complete, Abaqus/CAE enters the Visualization module and displays *X–Y* plots of the test results. Each plot includes the experimental data and a curve for each evaluated strain energy potential. Abaqus/CAE also opens a dialog box containing the stability limits and coefficients for each strain energy potential.

For example, the *X–Y* plot in *Figure 1* shows the results of a planar test using the Ogden N=3 strain energy potential.

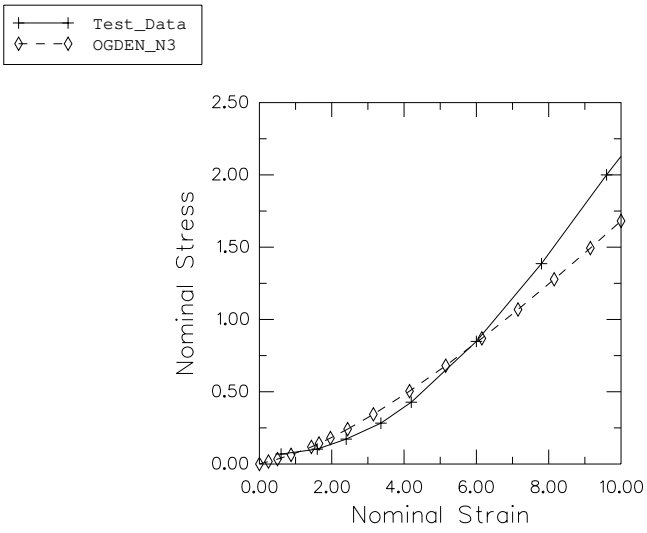


Figure 1: Results of a planar test.

In addition, the following information is reported to the data (`.dat`) file:

- The coefficients calculated for the strain energy potential.
- Any material instabilities that were detected during the tests.

The path to the data (`.dat`) file appears in the message area of the Abaqus/CAE main window once the analysis has completed successfully.

Evaluating multiple strain energy potentials

If you are defining a hyperelastic material using experimental data and you are unsure which strain energy potential to specify, you can select **Unknown** from the **Strain energy potential** list in the material editor. You can then use the **Evaluate** option to perform standard tests with the experimental data using multiple strain energy potentials.

When the tests are complete, Abaqus/CAE enters the Visualization module and displays an *X*–*Y* plot for each test and a dialog box containing the stability limits and coefficients for each strain energy potential. Each plot includes the experimental data and a curve for each evaluated strain energy potential. You can visually compare the strain energy potential curves and the experimental data curve and select the strain energy potential that provides the best fit.

Once you have determined which strain energy potential provides the best fit with the experimental data, you must return to the material editor in the Property module and change the **Strain energy potential** selection from **Unknown** to the strain energy potential that you have chosen.

Viewing behavior predicted by coefficients

If you have acquired coefficients for a particular strain energy potential (either by evaluating one or more hyperelastic strain energy potentials, as described above, or from another source), you may want to verify that the behavior predicted by the strain energy potential acceptably matches your experimental data or meets other criteria.

You can use the **Evaluate** option to plot a curve of the strain energy potential using the coefficients you provided in the material definition. If the material definition also includes experimental data, a curve for that data also appears in the plot.

Viewing response curves for viscoelastic materials

If you have shear or volumetric test results, you may want to verify that the creep and relaxation behavior predicted by Abaqus acceptably matches your experimental data or meets other criteria. Likewise, if you have frequency data, you may want to verify that the predicted storage and loss components of the shear and bulk moduli match your data.

You can use the **Evaluate** option to plot curves using the coefficients you provided in the material definition. If the material definition includes experimental data, curves for those data also appear in the plots. The types of curves produced depend on the material definition. For viscoelastic materials defined using a Prony series, creep test data, or relaxation test data for time, you can produce creep and relaxation plots versus time. For viscoelastic materials defined using frequency data for time, you can produce plots of the storage and loss components of the shear and bulk moduli versus a logarithmic scale of frequencies.

Adjusting material data

If you are unsatisfied with the fit between the test data and the behavior predicted by the material, you can return to the Property module and adjust the test data and then evaluate the material again. You can repeat this process until you are satisfied with the material behavior. In some cases it may be possible to use this approach to optimize the coefficient values included in a hyperelastic material definition. For more information, see "Improving the accuracy and stability of the test data fit," in *Hyperelastic Behavior of Rubberlike Materials* and *Hyperelastic Behavior in Elastomeric Foams* for hyperelastic and hyperfoam materials, respectively.

For detailed instructions on evaluating materials, see the following sections:

- *Displaying X–Y plots of hyperelastic material behavior*
- *Displaying X–Y plots of hyperfoam material behavior*
- *Displaying X–Y plots of viscoelastic material behavior*

For more information on the strain energy potentials available in Abaqus, see "Strain energy potentials," in *Hyperelastic Behavior of Rubberlike Materials*.

Using material libraries

You can use a material library to maintain a consistent set of material property data for use in all Abaqus/CAE analysis models. Material libraries are available only in the Property module.

This section provides information on accessing, using, and managing material libraries.

In this section:

- *Material libraries*
- *Managing material libraries*
- *Adding materials from a library to your model*

Material libraries

Material libraries provide a convenient means of storing material property data for use in multiple analyses. You can create one or more libraries to save and organize material data related to a project, a group of projects, or an entire company.
Use of material libraries allows you to maintain a consistent set of material properties and to quickly add materials to your models.

Material library tools are located in the Model Tree area in the Property module.

Material libraries are considered plug-ins to Abaqus/CAE, although they are loaded automatically without use of the **Plug-ins** menu. Material libraries have the file extension `.lib` and are stored in the `abaqus_plugins` directories located within the Abaqus/CAE installation, your home directory, or the current directory (the directory from which you launched Abaqus/CAE). You can also use the **plugin_central_dir** environment variable in the `abaqus_v6.env` file or the Abaqus solver `custom_v6.env` file to specify one or more additional directory paths. (For more information on the location of the plug-in directories, see *Where are plug-in files stored?*) Abaqus/CAE searches all subdirectories of the specified plug-in locations for material library files.

By default, Abaqus/CAE displays material libraries in a tree format, similar to the Model Tree. In this format you can expand and collapse categories to help locate a desired material. *Figure 1* shows a simple material library containing metal materials. It is organized into categories for aluminum, copper, and steel; the aluminum category has been expanded. You can also view material libraries as an alphabetical list of material names, omitting the categories.

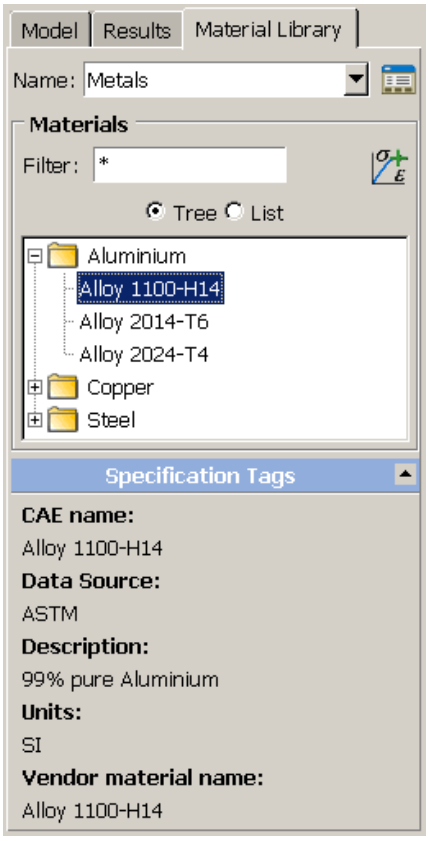


Figure 1: The Material Library.

You can use the **Filter** located above the materials list to search the material names for a list of characters. The filter may include any characters allowed in an Abaqus material name (for more information on object naming, see *Using basic dialog box components*). The filter is not applied to category names, so filtering may result in "empty" categories.

The  tool opens the **Material Library Manager**, where you can create, edit, rename, and reorganize material libraries. You can also use the manager to create, edit, rename, and delete **Specification Tags** to help identify materials in a library.

Figure 1 shows the predefined **Specification Tags**. The  tool copies a selected material from the library to the current model.

Additional information

- *Managing material libraries*
- *Adding materials from a library to your model*

Managing material libraries

The **Material Library Manager** allows you create, edit, and rename material libraries, library categories, and library materials.

The default **Specification Tags** indicate the source of the material data, a description, the units of measure, and a vendor material name. You can create, edit, rename, and delete specification tags for each material in a library. Material properties and categories are presented in alphabetical order, identical to the default view of the library in the main window of Abaqus/CAE; material categories appear first, followed by materials that are not in a category. You can use the manager to copy materials from a model to a library or from a library to a model. The **Material Library Manager** is shown in *Figure 1*.

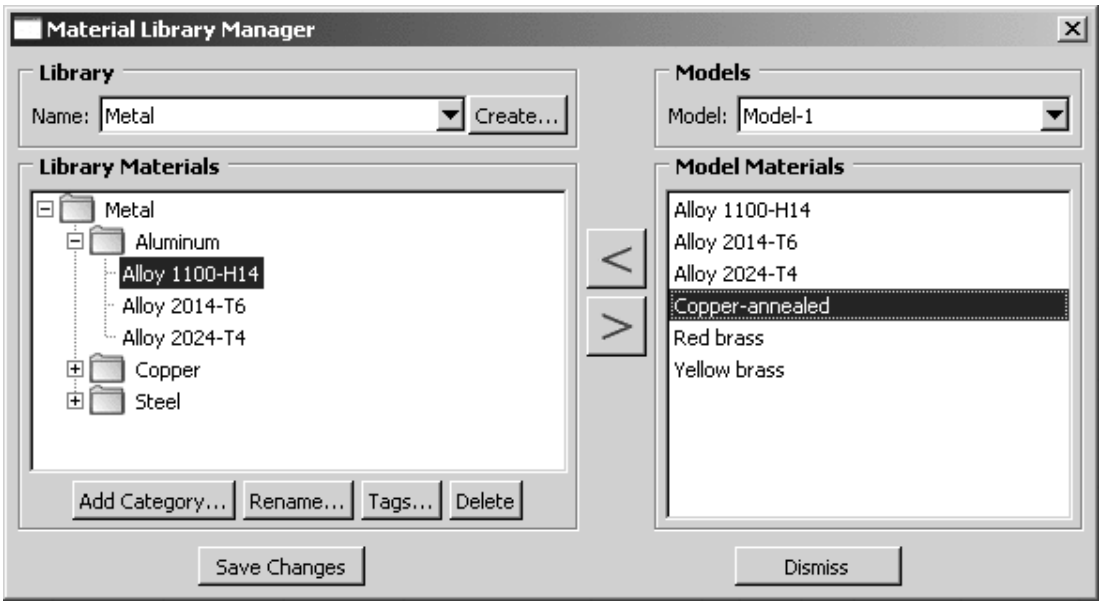


Figure 1: The Material Library Manager.

When you work with material libraries, it is recommended that you carefully consider the material and category names. You can create multiple categories and/or materials with exactly the same name. For example, you could have identical entries for a standard grade of steel, each containing properties in different sets of units. Using suitable category names, you can easily identify each material. However, if you display the library in the list format, the identical entries will all appear in the list. Modifying the material names to include the units information may make it easier to identify the desired material. For example, you can name one material `Steel 1020 US` and another `Steel 1020 SI`. Alternatively, you can click **Tags** in the material library manager to indicate the units in the **Specification Tags** that appear below the materials list in the main window. Renaming a material in the material library will change the name only in the library and does not change the underlying material name copied from or to the Abaqus/CAE model. A material added from the library to a model will still retain the old name.

You cannot view or edit material properties within the material library manager. To view or edit the properties of a material, you must add the material to a model and use the **Edit Material** dialog box (for more information, see *Creating or editing a material*).

1. From the **Material Library** tab in the Model Tree area, click the **Material Library Manager** icon  located to the right of the **Name** field.

   Abaqus/CAE displays the **Material Library Manager**.

2. From the top of the manager, click **Create** to create a new material library, or select a library name to edit an existing library.

3. If you selected **Create** in the previous step, the **Create Material Library** dialog box appears.
   a. Enter a name for the new library.
   b. Select **Home** or **Current** to choose the location of the `abaqus_plugins` directory in which the library will be saved.
   c. Click **OK**.

      Abaqus/CAE creates an empty material library file and opens it for editing in the manager. If necessary, Abaqus/CAE also creates the specified `abaqus_plugins` directory. For more information, see *Material libraries*.

4. To edit the current material library, select an item from the **Library Materials** list on the left side of the manager and choose from the following:

   - If you want to create categories to organize materials in the library, click **Add Category**. The **Create Category** dialog box appears.

     1. Enter a name for the category.
     2. Click **OK** in the **Create Category** dialog box.

        Abaqus/CAE creates the new category.

     The category appears at the same level of the library as the item you selected. For example, if you selected the library name, the new category appears with any other categories directly below the library name; if you selected an existing category, the new category appears within that category.
   - If you want to rename the selected item, click **Rename**. Enter a new name in the dialog box that appears, and click **OK**.

     Note:

     Renaming a library changes the name displayed within Abaqus/CAE; it does not change the name of the library (`.lib`) file.

   - If you want to edit the information that appears for a material when you expand the **Specification Tags** field at the bottom of the material library, click **Tags**. The tags contain the source, description, units, and vendor name of the material. By default, the vendor name tag contains the material name, and the other tags each contain "Imported from CAE." You can use tags to clarify the intended use of materials that have identical names or similar properties.
   - If you want to remove a material or an empty category from the library, select it and click **Delete**. You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple items.

     Note:

     You cannot delete a material library from within Abaqus/CAE; you must delete the library file from the saved location on your system.

5. Use the arrows located between the **Library Materials** list and the **Model Materials** list to copy material data from a library to a model, or vice versa.

   When you copy a material, Abaqus/CAE places the new material at the end of the selected category, if any, or at the end of the library.
   a. From the **Models** list in the **Material Library Manager**, select the model to which (or from which) you want to copy a material.
   b. Select the desired material from the **Library Materials** list or the **Model Materials** list. You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple items.

c. If you are copying a material into the library, select the category in the **Library Materials** list into which you want Abaqus/CAE to place the material.

d. Click on the appropriate arrow to copy the material.

Abaqus/CAE adds the new material to the end of the selected category for a library or to the end of the materials list for a model.

Changes that you make in the **Material Library Manager** are visible immediately in the manager dialog box. However, they are not committed to the library file until you click **Save Changes**. Abaqus/CAE does not update the library view in the main window until you dismiss the material library manager.

Additional information

- *Material libraries*
- *Adding materials from a library to your model*

Adding materials from a library to your model

To view material libraries, select the **Material Library** tab in the Model Tree area of the Property module. If more than one library is available, select one from the list at the top of the tabbed page. In the default **Tree** view, expand categories to view the materials within each category. To hide the categories and view an alphabetical list of all materials in the current library, select the **List** view.

To add a material from a library to the current model, highlight the material name in the tree or list view, and click the **Add Material** icon  at the upper right corner of the **Materials** list. Alternatively, you can double-click on a material name in the library to add it to the model.

1. Click **Material Library** at the top of the Model Tree area located on the left side of the main window.

    **Tip:** If the Model Tree is not displayed, select **View->Show Model Tree** from the main menu bar.

2. Select a library name from the **Name** list.

   If Abaqus/CAE did not find any material libraries at the start of the session, you may create a new library. For more information, see *Managing material libraries*.

   Abaqus/CAE displays the library contents. By default, the library is displayed in a tree format where materials may be separated into categories, similar to the Model Tree.

3. Locate the desired material in the library using any of the following methods:

   - Expand categories within the **Tree** view to view their contents.
   - Toggle on **List** to hide the categories and view all materials in the library.
   - Enter a string in the **Filter** field to show only those materials whose names contain that string.

4. Click on the name of a material to select it.
5. If desired, click the arrow to the right of **Specification Tags** (located below the materials list) to view more information about the selected material.
6. Click the **Add Material** icon  at the upper right corner of the **Materials** list to add the material to the current model.

Once you have added a material to your model, use the **Edit Material** dialog box to view or edit the material properties. Use the other tools in the Property module to associate the material with a section and assign the section to part of your model.

Additional information

- *The Property module*
- *Material libraries*
- *Managing material libraries*

Using the Property module toolbox

You can access all the Property module tools through either the main menu bar or the Property module toolbox. *Figure 1* shows the icons for all the property tools in the Property module toolbox.

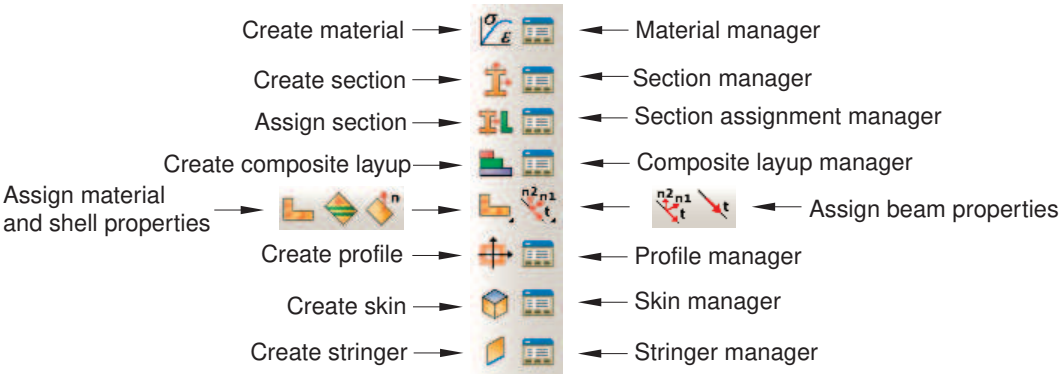


Figure 1: The Property module toolbox.

Creating and editing materials

This section describes each feature of the material editor individually.

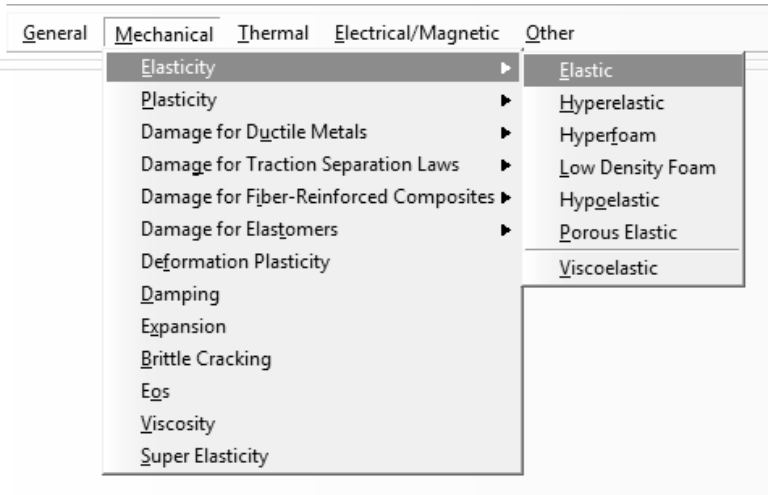
In this section:

- *Creating or editing a material*
- *Browsing and modifying material behaviors*
- *Entering strain-rate-dependent data*
- *Entering temperature-dependent data*
- *Specifying field variable dependence*
- *Selecting and modifying suboptions or test data*
- *Displaying X–Y plots of hyperelastic material behavior*
- *Displaying X–Y plots of viscoelastic material behavior*
- *Displaying X–Y plots of hyperfoam material behavior*

Creating or editing a material

You use the **Edit Material** dialog box to create a new material or edit an existing material. When you select **Material->Create** from the main menu bar, you can enter the name of your choice for the material or accept the default name, you can provide a description for the material, and you can define the material properties. When you select **Material->Edit**, you can redefine the material description or properties, but you must use **Material->Rename** to change the name of an existing material.

Use the menu bar under the **Material Behaviors** list to add properties to a material. Some of the menu items contain submenus; for example, the following figure shows the behaviors available under the **Mechanical->Elasticity** menu item:



Note:

To display information on a particular material behavior, click and hold that behavior and then press `F1`. A help window appears that contains information about the parameters and data associated with the behavior.

Use the **Material Behaviors** list to select an existing material behavior to edit.

Warning: Abaqus/CAE does not check for missing or invalid material behaviors until you submit the job for analysis. (Any warnings and errors are reported by the Job module.) Therefore, you must be careful to supply valid data for all of the material behaviors that the analysis requires.

1. Display the material editor using one of the following methods:
   - From the main menu bar, select **Material->Create**.

     Tip: You can also click **Create** in the **Material Manager** or select the create material tool in the Property module toolbox.

   - From the main menu bar, select **Material->Edit->*material name***.

     Tip: You can also select a material and click **Edit** in the **Material Manager**.

An **Edit Material** dialog box appears.

2. If you are creating a new material, enter the name of your choice for the material. For more information on naming objects, see *Using basic dialog box components*.
3. If desired, enter a description for the material.

   a. Click  in the **Edit Material** dialog box.

   The material description editor appears.

   b. In the material description editor, type information that you want to record about the material.
   c. Click **OK** to store the description and to close the material description editor.

   When you submit a job, Abaqus/CAE writes material descriptions to the input file using comment lines; the material descriptions are not written to the output database. For more information, see *Adding descriptions to your Abaqus/CAE model*.
4. Use the menu bar or **Material Behaviors** list to select a new or existing material behavior, respectively.

   The behavior definition area in the dialog box changes to show all the parameters and data associated with the selected material behavior.
5. Edit the parameters and data to complete the material definition.
6. If you wish to remove a material behavior, select it and click  to the right of the menu bar.
7. When you have finished editing the material definition, click **OK** to save the material and to close the dialog box.

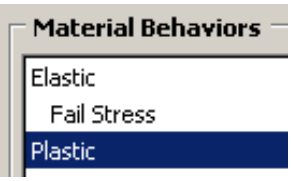
Additional information

- *Understanding the Property module editors*
- *Creating and editing materials*

Browsing and modifying material behaviors

The selected behavior list at the top of the material editor window displays the behaviors and suboptions that comprise the current material; the list is updated as you add and delete behaviors.

The following figure shows how the list would look if an elastic-plastic material complete with stress-based failure limits were defined:



Using the selected behavior list, you can add, delete, or change materials as follows:

Adding material behaviors

Select the behaviors needed to define your material from the menus just below the selected behavior list. When you select a behavior, its name appears in the list, and the parameters and data associated with the behavior appear in the data area in the bottom portion of the editor window. Suboptions appear beneath the corresponding primary behavior and are indented to indicate their subordinate position.

Deleting material behaviors

Within the selected behavior list, click the behavior or suboption you want to delete; then click the  icon located near the lower right corner of the behavior list. This procedure removes the behavior from both the behaviors list and the material definition. If you delete a behavior that has suboptions shown beneath it in the list, the suboptions are also deleted.

Changing material parameters or data

Within the selected behavior list, click the behavior whose data you want to change. When the parameters and data associated with the behavior appear in the data area in the bottom portion of the window, make the desired changes.

Additional information

- *Understanding the Property module editors*

Entering strain-rate-dependent data

If your material includes strain rate dependence, you can enter data to define how material properties vary with strain rate.

1. + Toggle on **Use strain-rate-dependent data** in the material editor.

   A column labeled **Rate** appears in the tabular data area.

2. Fill in each row with the appropriate values. For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.)

Additional information

• *Understanding the Property module editors*

Entering temperature-dependent data

If your material includes temperature dependence, you can enter data to define how material properties vary with increasing temperature.

1. + Toggle on **Use temperature-dependent data** in the material editor.

   A column labeled **Temp** appears in the tabular data area.

2. Fill in each row with the appropriate values. For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.)

Additional information

• *Understanding the Property module editors*

Specifying field variable dependence

The **Number of field variables** text field in the material editor allows you to specify the number of field variables to be referenced by a given material behavior. Columns for each field variable appear in the table in the data area of the editor.

1. Change the number of field variables in the **Number of field variables** box to the desired value using one of these methods:
   - Click the arrows to the right of the text field to increase or decrease the number of field variables.
   - Type the number directly in the text field.

   Either method adds field variable columns to the table in the data area.
2. Enter the appropriate data in each cell of the table. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.)

Additional information

• *Understanding the Property module editors*

Selecting and modifying suboptions or test data

If suboptions or test data are available for the current behavior, the **Suboptions**, **Test Data**, or **Uniaxial Test Data** menus will be available in the upper right corner of the data area. When you select one of the options from the menus, the **Suboption Editor** or the **Test Data Editor** appears in which you can enter the required data.



Note:

To display context-sensitive help for specific buttons, text fields, and other options in the **Suboption Editor** or the **Test Data Editor**, you must select the option of interest and then press **[F1]**. (The **Help** menu in the main menu bar is unavailable while the editors are displayed.) For detailed information on using **[F1]** to obtain help, see *Displaying context-sensitive help*.

1. Click **Suboptions**, **Test Data**, or **Uniaxial Test Data** in the upper right corner of the data area and select the option of your choice from the list that appears.

   The **Suboption Editor** or the **Test Data Editor**, as appropriate for your selection, appears in a separate dialog box.

2. Enter the required data inside the editor and then click **OK** to return to the material editor.

   You can enter data into a suboption or test data table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For example, an option exists for creating an *X–Y* data object from the data in the table; you can plot the *X–Y* data in the Visualization module and visually check its validity. Another option exists for automatically entering data from a file. For detailed information on each option, see *Entering tabular data*.

Additional information

- *Understanding the Property module editors*

Displaying *X–Y* plots of hyperelastic material behavior

Abaqus/CAE allows you to evaluate hyperelastic material behavior by automatically creating response curves using selected strain energy potentials. When the curve fitting is complete, Abaqus/CAE opens the Visualization module and displays *X–Y* plots of the test results and a dialog box containing the stability limits for each strain energy potential. You can review the results and adjust the material as necessary. For more information, see *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*.

1. From the main menu bar, select **Material->Evaluate->*material name***. The material that you select must include hyperelastic material data.

   Tip: You can also select the name of the material in the **Material Manager** and then click **Evaluate**.

   An **Evaluate Material** dialog box appears.

2. If you selected a hyperelastic material that also includes viscoelastic material data, toggle on **Perform hyperelastic evaluation** if it is not already selected.

   If desired, you can also evaluate the viscoelastic behavior of the material. For more information, see *Displaying X–Y plots of viscoelastic material behavior*.

3. In the **Available Input Data** field, do the following:

   a. Select the **Source** option of your choice:

   - Select **Test data** if you want Abaqus to calculate the necessary strain energy potential coefficients from the experimental data specified in the material definition.
   - Select **Coefficients** if you want Abaqus to use the coefficients specified in the material definition.

   b. If you selected **Test data** in the previous step, specify the test data type or types that you want Abaqus to use in calculating the strain energy potential coefficients. (Only data types for which you have specified data in the material definition appear in the list.)

   c. If you intend to evaluate the **Marlow** strain energy potential, specify the test data type that Abaqus will use to define the deviatoric response. You can also specify whether compression, tension, or both types of test data should be used and whether volumetric test data should be used to define the volumetric response. (For more information, see *Marlow Form*.)

   Note:

   If your hyperelastic material model includes lateral nominal strains, temperature-dependent data, or field variables, **Marlow** will be the only strain energy potential available for evaluation.

4. From the list of **Standard Tests**, select one or more tests for which you want response calculations using the data in the material definition.
5. For each test that you select, enter a minimum and maximum strain value that will be the upper and lower limits for the stress-strain response curves.
6. Click the **Strain Energy Potentials** tab, and do the following:

   - If you selected **Test data** as a data source, a list of all the available strain energy potentials appears. From the list, select one or more that you want Abaqus to apply to the experimental data. For more information on the strain energy potentials available in Abaqus see *Strain Energy Potentials*.
   - If you selected **Coefficients** as a data source, the name of the strain energy potential specified in the material definition appears. You can simply review the information and move on to the next step.

7. If the material that you are evaluating also includes viscoelastic material properties, click the **Viscoelastic** tab; you can either toggle off **Perform viscoelastic evaluation**, or select viscoelastic evaluation options. For more information, see *Displaying X–Y plots of viscoelastic material behavior*.
8. Click **OK** to begin the response calculations.

   If the evaluation fails during the extraction of material coefficients due to problems with nonlinear curve-fitting, Abaqus/CAE displays a dialog box containing the name of the data (`.dat`) file; the path to the data file is printed in the message area. The data file provides detailed information on each problem encountered. (For more information on the data file, see *About Output*.)

   If Abaqus completes the tests successfully, Abaqus/CAE enters the Visualization module and displays *X–Y* plots of the test results in new viewports. (For information on *X–Y* plots, see *X–Y plotting*.) The data objects appear in the ***X–Y* Data Manager**; you can copy them to an output database or perform any of the tasks that you can perform on other *X–Y* data in the Visualization module.

   In addition, Abaqus/CAE displays an informational dialog box containing the stability limits and coefficients for each hyperelastic strain energy potential. The dialog box also displays the viscoelastic material parameters if a viscoelastic evaluation was performed. Abaqus/CAE displays in the message area the path to the data (`.dat`) file that contains all the material evaluation information.
9. If desired, return to the Property module to edit the material data or to evaluate other materials.

   For example, if the **Strain energy potential** for the hyperelastic material was previously set to **Unknown**, you can use the evaluation results to complete the material definition using the optimal strain energy potential.

Additional information

• *Hyperelastic Behavior of Rubberlike Materials*

Displaying *X–Y* plots of viscoelastic material behavior

Abaqus/CAE allows you to evaluate viscoelastic material behavior by creating either relaxation and creep curves (for Prony series coefficients, relaxation test data, or creep test data) or shear and bulk modulus curves (for frequency data) based on the material definition.
When the curve fitting is complete, Abaqus/CAE opens the Visualization module and displays *X–Y* plots of the test results and a dialog box containing the material parameters. You can review the results and adjust the material as necessary. For more information, see *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*.

1. From the main menu bar, select **Material->Evaluate->*material name***. The material that you select must include time domain viscoelastic material data defined in conjunction with hyperelastic and/or elastic material data.

   Tip: You can also select the name of the material in the **Material Manager** and then click **Evaluate**.

   An **Evaluate Material** dialog box appears.

2. If you selected a viscoelastic material that also includes hyperelastic material data, click on the **Viscoelastic** tab; and toggle on **Perform viscoelastic evaluation** if it is not already selected.

   If desired, you can also evaluate the hyperelastic behavior of the material. For more information, see *Displaying X–Y plots of hyperelastic material behavior*.

3. In the **Available Input Data** field, do the following:

   a. Select the **Source** option of your choice:

   - Select **Test data** if you want Abaqus to calculate viscoelastic response using the experimental data specified in the material definition.
   - Select **Coefficients** if you want Abaqus to calculate viscoelastic response using the coefficients specified in the material definition. If the material was defined using a Prony series, relaxation test data, or creep test data for time, Abaqus uses the hyperelastic or elastic coefficient data. If the material was defined using frequency data for time, Abaqus uses the frequency coefficients specified in the viscoelastic material definition.

   b. If you selected **Test data** in the previous step, toggle on the test data type that you want Abaqus to use in calculating the material response. (Only data types for which you have specified data in the material definition appear in the list.)

      Note:

      Combined data cannot be selected at the same time as **Shear** or **Volumetric** data.

4. In the **Normalized Response Plots** field, toggle on **Stress Relaxation** and/or **Creep** to define the response modes that Abaqus will calculate; and enter the time period for the normalized response curves.

   If viscoelasticity is defined using frequency data in the time domain, the **Normalized Response Plots** field is not available. Instead, Abaqus produces shear and bulk modulus response curves on a logarithmic frequency scale.

   Note:

   When you evaluate a viscoelastic material using frequency data, Abaqus obtains expressions for the shear and bulk moduli by converting the Prony series terms from the time domain to the frequency domain. It is recommended that you independently verify the material model in the

domain in which the data will be used. For more information, see *Determination of Isotropic Viscoelastic Material Parameters*.

5. Click **OK** to begin the response calculations.

   If the evaluation fails during the extraction of material coefficients due to problems with nonlinear curve-fitting, Abaqus/CAE displays a dialog box containing the name of the data (`.dat`) file; the path to the data file is printed in the message area. The data file provides detailed information on each problem encountered. (For more information on the data file, see *About Output*.)

   If Abaqus completes the tests successfully, Abaqus/CAE enters the Visualization module and displays *X–Y* plots of the test results in new viewports. (For information on *X–Y* plots, see *X–Y plotting*.) The data objects appear in the ***X–Y* Data Manager**; you can copy them to an output database or perform any of the tasks that you can perform on other *X–Y* data in the Visualization module.

   In addition, Abaqus/CAE displays an informational dialog box containing the viscoelastic material parameters and the stability limits and coefficients for each hyperelastic strain energy potential if a hyperelastic evaluation was performed. Abaqus/CAE also displays in the message area the path to the data (`.dat`) file that contains all the material evaluation information.

6. If desired, return to the Property module to edit the material data or to evaluate other materials.

Additional information

• *Time Domain Viscoelasticity*

Displaying *X–Y* plots of hyperfoam material behavior

Abaqus/CAE allows you to evaluate hyperfoam material behavior by automatically creating response curves. When the curve fitting is complete, Abaqus/CAE opens the Visualization module and displays *X–Y* plots of the test results and a dialog box containing the stability limits for each strain energy potential.
You can review the results and adjust the material as necessary. For more information, see *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*.

1. From the main menu bar, select **Material->Evaluate->*material name***. The material that you select must include hyperfoam material data.

   Tip: You can also select the name of the material in the **Material Manager** and then click **Evaluate**.

   An **Evaluate Material** dialog box appears.

2. If you selected a hyperfoam material that also includes viscoelastic or hyperelastic material data, toggle on **Perform hyperfoam evaluation** if it is not already selected.

   If desired, you can also evaluate the viscoelastic behavior of the material. For more information, see *Displaying X–Y plots of viscoelastic material behavior*.

3. In the **Available Input Data** field, do the following:

   a. Select the **Source** option of your choice:

   - Select **Test data** if you want Abaqus to calculate the necessary coefficients from the experimental data specified in the material definition.
   - Select **Coefficients** if you want Abaqus to use the coefficients specified in the material definition.

   b. If you selected **Test data** in the previous step, specify the test data type or types that you want Abaqus to use in calculating the coefficients. (Only data types for which you have specified data in the material definition appear in the list.)

4. From the list of **Standard Tests**, select one or more tests for which you want response calculations using the data in the material definition.
5. For each test that you select, enter a minimum and maximum strain value that will be the upper and lower limits for the stress-strain response curves.
6. If the material that you are evaluating also includes viscoelastic material properties, click the **Viscoelastic** tab; you can either toggle off **Perform viscoelastic evaluation**, or select viscoelastic evaluation options. For more information, see *Displaying X–Y plots of viscoelastic material behavior*.
7. Click **OK** to begin the response calculations.

   If the evaluation fails during the extraction of material coefficients due to problems with nonlinear curve-fitting, Abaqus/CAE displays a dialog box containing the name of the data (`.dat`) file; the path to the data file is printed in the message area. The data file provides detailed information on each problem encountered. (For more information on the data file, see *About Output*.)

   If Abaqus completes the tests successfully, Abaqus/CAE enters the Visualization module and displays *X–Y* plots of the test results in new viewports. (For information on *X–Y* plots, see *X–Y plotting*.) The data objects appear in the ***X–Y* Data Manager**; you can copy them to an output database or perform any of the tasks that you can perform on other *X–Y* data in the Visualization module.

   In addition, Abaqus/CAE displays an informational dialog box containing the stability limits and coefficients. The dialog box also displays the viscoelastic material parameters if a viscoelastic evaluation was performed. Abaqus/CAE displays in the message area the path to the data (`.dat`) file that contains all the material evaluation information.

8. If desired, return to the Property module to edit the material data or to evaluate other materials.

Additional information

• *Hyperelastic Behavior in Elastomeric Foams*

Defining general material data

This section describes how you can specify general material data.

For more information, see the following sections:

- *Density*
- *About User Subroutines and Utilities*
- *Regularizing User-Defined Data in Abaqus/Explicit*
- *User-Defined Mechanical Material Behavior*
- *User-Defined Thermal Material Behavior*
- *USDFLD*
- *UVARM*

In this section:

- *Specifying material mass density*
- *Specifying solution-dependent state variables*
- *Regularizing user-defined material data in Abaqus/Explicit*
- *Defining constants for a user material*
- *Defining field variables at a material point*
- *Specifying the number of user variables*

Specifying material mass density

You can define density as a function of temperature and field variables. For acoustic, heat transfer, coupled temperature-displacement, and coupled thermal-electrical elements the density is continually updated to the value corresponding to the current temperature and field variables. However, for all other elements density is a function of the initial values of temperature and field variables and changes in volume only. Abaqus does not update density values if temperatures and field variables change during the analysis.

See *Density*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **General->Density**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:

   - Select **Uniform** to define density that is uniformly distributed throughout the material.
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying density. Only analytical fields and discrete fields that are valid for density are available in the selection list. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.

     Alternatively, you can click  to create a new discrete field.

3. Toggle on **Use temperature-dependent data** to define density as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.

4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the density depends.
5. Enter the following data in the **Data** table:

Mass Density

Mass density. (Units of ML^{-3}.)

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Specifying solution-dependent state variables

Solution-dependent state variables are values in user subroutines that you can define to evolve with the solution of an analysis.
If you refer to a material definition in a user subroutine, you can use the **Edit Material** dialog box to specify the number of solution-dependent variables required at the points or nodes to which that material is applied. See *About User Subroutines and Utilities* for more information.

1. From the menu bar in the **Edit Material** dialog box, select **General->Depvar**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrows to the right of the **Number of solution-dependent state variables** field to specify how many solution dependent state variables you want to allocate space for at each applicable integration point or contact secondary node.
3. If applicable, enter the state variable number controlling the element deletion flag in the field labeled **Variable number controlling element deletion**. For more information, see *Deleting Elements from a Mesh Using State Variables*.
4. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Regularizing user-defined material data in Abaqus/Explicit

Interpolating material data as a function of independent variables requires table lookups of the material data values during analysis. The table lookups occur frequently in Abaqus/Explicit and are most economical if the interpolation is from regular data.

If necessary, Abaqus/Explicit uses an error tolerance to regularize the input data. The number of intervals in the range of each independent variable is chosen such that the error between the piecewise linear regularized data and each of your defined points is less than the tolerance times the range of the dependent variable.

For more information, see *Regularizing User-Defined Data in Abaqus/Explicit*.

1. From the menu bar in the **Edit Material** dialog box, select **General->Regularization**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. In the **Rtol** field, enter the tolerance that you want Abaqus to use for regularizing the material data. The default is 0.03.
3. Specify how you want to interpolate strain rate–dependent data:

   - Select **Logarithmic** to interpolate strain rate data using logarithmic intervals rather than uniformly spaced intervals. This option generally provides a better match to typical strain rate–dependent curves.
   - Select **Linear** to use uniform intervals for interpolation of strain rate data.
4. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining constants for a user material

The following user subroutines allow you to create user-defined material models:

- *UMAT* for mechanical material models in Abaqus/Standard.
- *VUMAT* for mechanical material models in Abaqus/Explicit.
- *UMATHT* for thermal material models in Abaqus/Standard.
- *VUMATHT* for thermal material models in Abaqus/Explicit.

You can enter constants in the **Edit Material** dialog box for use in these subroutines. See the following sections for more information:

- *User-Defined Mechanical Material Behavior*
- *User-Defined Thermal Material Behavior*

1. From the menu bar in the **Edit Material** dialog box, select **General->User Material**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. If you are performing an Abaqus/Standard analysis, click the arrow to the right of the **User material type** field, and select the type of user material for which you are defining constants.
3. If you are performing an Abaqus/Standard analysis, toggle on **Use unsymmetric material stiffness matrix** if the material stiffness matrix, $\partial \Delta \boldsymbol{\sigma} / \partial \Delta \boldsymbol{\varepsilon}$, is not symmetric or, in the case of a thermal constitutive model, if $\partial \mathbf{f} / \partial (\partial \theta / \partial \mathbf{x})$ is not symmetric. This parameter causes Abaqus/Standard to use its unsymmetric equation solution procedures.
4. If you are defining a mechanical or thermomechanical material, enter the **Mechanical Constants** in the **Data** table.
   For detailed information on how to enter data, see *Entering tabular data*.
5. If you are defining a thermal or thermomechanical material, enter the **Thermal Constants** in the **Data** table.
   For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining field variables at a material point

In Abaqus/Standard you can introduce dependence on solution variables with user subroutine *USDFLD*. This subroutine allows you to define field variables at a material point as functions of time, of any of the available material point quantities listed in *Abaqus/Standard Output Variable Identifiers*, and of material directions. Material properties defined as functions of these field variables may, thus, be dependent on the solution.

User subroutine *USDFLD* is called at each point for which the material definition includes a reference to the user subroutine.

1. From the menu bar in the **Edit Material** dialog box, select **General->User Defined Field**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying the number of user variables

You can allocate space at each material calculation point for user-defined output variables defined in user subroutine *UVARM*.

1. From the menu bar in the **Edit Material** dialog box, select **General->User Output Variables**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrows to the right of the **Number of user-defined variables at each material point** field to specify how many user-defined variables you want to allocate space for at each material calculation point. Any number of user-defined output variables can be used.
3. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining mechanical material models

This section describes how you can specify mechanical material models and related material data.

For more information about mechanical material models, see the following sections:

- *Elastic Mechanical Properties*
- *Inelastic Mechanical Properties*
- *Progressive Damage and Failure*
- *Other Material Properties*

In this section:

- *Defining elasticity*
- *Defining plasticity*
- *Defining damage*
- *Defining other mechanical models*

Defining elasticity

You use the **Edit Material** dialog box to create an elastic material and to specify its elastic material properties.

You can create the following elastic material models:

- **Elastic**
- **Isotropic Hyperelastic**
- **Anisotropic Hyperelastic**
- **Hyperfoam**
- **Low-Density Foam**
- **Hypoelastic**
- **Porous Elastic**
- **Viscoelastic**

For more information, see *Elastic Behavior*.

In this section:

- *Creating a linear elastic material model*
- *Creating an isotropic hyperelastic material model*
- *Creating an anisotropic hyperelastic material model*
- *Creating a hyperfoam material model*
- *Creating a low-density foam material model*
- *Creating a hypoelastic material model*
- *Creating a porous elastic material model*
- *Creating a viscoelastic material model*

Creating a linear elastic material model

Linear elasticity is the simplest form of elasticity available in Abaqus.
The linear elastic model can define isotropic, orthotropic, or anisotropic material behavior and is valid for small elastic strains. See *Specifying elastic material properties* for details on how to define a linear elastic material model.

Failure theories are provided for use with linear elasticity. They can be used to obtain postprocessed output requests.

In this section:

- *Specifying elastic material properties*
- *Defining stress-based failure measures for an elastic model*
- *Defining strain-based failure measures for an elastic model*

Specifying elastic material properties

A linear elastic material model is valid for small elastic strains (normally less than 5%); can be isotropic, orthotropic, or fully anisotropic; and can have properties that depend on temperature and/or other field variables. For more information, see *Linear Elastic Behavior*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Elastic**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. From the **Type** field, choose the type of data you will supply to specify the elastic material properties.

   - Choose **Isotropic** to specify isotropic elastic properties, as described in *Defining Isotropic Elasticity*.
   - Choose **Engineering Constants** to specify orthotropic elastic properties by giving the engineering constants, as described in *Defining Orthotropic Elasticity by Specifying the Engineering Constants*.
   - Choose **Lamina** to specify orthotropic elastic properties in plane stress, as described in *Defining Orthotropic Elasticity in Plane Stress*.
   - Choose **Orthotropic** to specify orthotropic elastic properties directly, as described in *Defining Orthotropic Elasticity by Specifying the Terms in the Elastic Stiffness Matrix*.
   - Choose **Anisotropic** to specify anisotropic elastic properties, as described in *Defining Fully Anisotropic Elasticity*.
   - Choose **Traction** to specify orthotropic elastic properties for warping elements, as described in *Defining Orthotropic Elasticity for 1-DOF Warping Elements*, or to define uncoupled elastic properties for cohesive elements, as described in *Defining Elasticity in Terms of Tractions and Separations for Cohesive Elements*.
   - Choose **Coupled Traction** to specify coupled elastic properties for cohesive elements, as described in *Defining Elasticity in Terms of Tractions and Separations for Cohesive Elements*.
   - Choose **Shear** to specify a linear isotropic deviatoric material model. For more information, see *Deviatoric Behavior*.
   - Choose **Bilamina** to specify orthotropic elasticity in plane stress with different moduli in tension and compression, as described in *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*.

3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

5. If you are defining the elastic behavior of a viscoelastic material, click the arrow to the right of the **Moduli time scale (for viscoelasticity)** field to specify either long-term or instantaneous elastic response.
6. Toggle on **No compression** if you want to modify the elastic material response such that compressive stress cannot be generated. For details, see *No Compression or No Tension*.
7. Toggle on **No tension** if you want to modify the elastic material response such that tensile stress cannot be generated. For details, see *No Compression or No Tension*.
8. Enter the material properties in the **Data** table.

   - For **Isotropic** data, enter the Young's modulus, E, and Poisson's ratio, ν.

- For **Engineering Constants** data, enter the generalized Young's moduli in the principal directions, E_1, E_2, E_3; the Poisson's ratios in the principal directions, ν_{12}, ν_{13}, ν_{23}; and the shear moduli in the principal directions, G_{12}, G_{13}, G_{23}.
- For **Lamina** data, enter the Young's moduli, E_1, E_2; the Poisson's ratio, ν_{12}; and the shear moduli, G_{12}, G_{13}, G_{23}. The G_{13} and G_{23} shear moduli are needed to define transverse shear behavior in shells.
- For **Orthotropic** data, enter the 9 elastic stiffness parameters: D_{1111}, D_{1122}, etc. (units of FL^{-2}).
- For **Anisotropic** data, enter the 21 elastic stiffness parameters: D_{1111}, D_{1122}, etc. (units of FL^{-2}).
- For **Traction** data, your entries depend on the element type that you are modeling.
  - For solid cross-section Timoshenko beam elements modeled with warping elements, enter the Young's modulus, E, and the shear moduli in the material directions, G_1 and G_2.
  - For cohesive elements with uncoupled traction, enter the elastic modulus in the normal direction and the two local shear directions, E_{nn}, E_{ss}, and E_{tt}.
- For **Coupled Traction** data, enter the six elastic moduli: E_{nn}, E_{ss}, E_{tt}, E_{ns}, E_{nt}, and E_{st}.
- For **Shear** data, enter the **Shear Modulus**.
- For **Bilamina** data, enter the Tensile Young's moduli, E_{1+}, E_{2+}; the Tensile Poisson's ratio, ν_{12+}; the shear moduli, G_{12}, G_{13}; the Compressive Young's moduli, E_{1-}, E_{2-}; and the Compressive Poisson's ratio, ν_{12-}.

9. To define the plane stress orthotropic failure measures for the material, if desired, click **Suboptions**. For details, see the following sections:

- *Defining stress-based failure measures for an elastic model*"
- *Defining strain-based failure measures for an elastic model*"

10. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining stress-based failure measures for an elastic model

Use the **Suboption Editor** to define the stress limits for stress-based failure measures for an elastic material model. For more information, see *Plane Stress Orthotropic Failure Measures*.

1. Create a linear elastic material model as described in *Specifying elastic material properties*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Fail Stress**.
   The **Suboption** Editor appears.
3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.
   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
   Field variable columns appear in the **Data** table.
5. In the **Data** table, enter the stress limits:

   #### Ten Stress Fiber Dir

   Tensile stress limit in the fiber direction, X_t.

   #### Com Stress Fiber Dir

   Compressive stress limit in the fiber direction, X_c.

   #### Ten Stress Transv Dir

   Tensile stress limit in the transverse direction, Y_t.

   #### Com Stress Transv Dir

   Compressive stress limit in the transverse direction, Y_c.

   #### Shear Strength

   Shear strength in the *X*–*Y* plane, *S*.

   #### Cross-prod Term Coeff

   Cross product term coefficient, f^* ($-1.0 \leq f^* \leq 1.0$). This value is used only for the Tsai-Wu theory and is ignored if σ_{biax} is provided. The default is zero.

   #### Stress Limit

   Biaxial stress limit, σ_{biax}. This value is used only for the Tsai-Wu theory. If this entry is nonzero, f^* is ignored.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Defining strain-based failure measures for an elastic model

Use the **Suboption Editor** to define the strain limits for strain-based failure measures for an elastic material model. For more information, see *Plane Stress Orthotropic Failure Measures*.

1. Create a linear elastic material model as described in *Specifying elastic material properties*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Fail Strain**.
3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. In the **Data** table, enter the strain limits:

   ### Ten Strain Fiber Dir

   Tensile strain limit in the fiber direction, $X_{\varepsilon t}$.

   ### Com Strain Fiber Dir

   Compressive strain limit in the fiber direction, $X_{\varepsilon c}$.

   ### Ten Strain Transv Dir

   Tensile strain limit in the transverse direction, $Y_{\varepsilon t}$.

   ### Com Strain Transv Dir

   Compressive strain limit in the transverse direction, $Y_{\varepsilon c}$.

   ### Shear Strain

   Shear strain limit in the *X*–*Y* plane, S_{ε}.

   You may need to expand the dialog box to see all the columns in the **Data**. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Creating an isotropic hyperelastic material model

The isotropic hyperelastic model describes the behavior of nearly incompressible materials that exhibit instantaneous elastic response up to large strains.
For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

In this section:

- *Defining isotropic hyperelastic materials*
- *Entering material parameters to define an isotropic hyperelastic material*
- *Providing test data to define an isotropic hyperelastic material*
- *Specifying uniaxial test data for an isotropic hyperelastic material model*
- *Specifying biaxial test data for an isotropic hyperelastic material model*
- *Specifying planar test data for an isotropic hyperelastic material model*
- *Specifying volumetric test data for an isotropic hyperelastic material model*
- *Defining hysteretic behavior for an isotropic hyperelastic material model*

Defining isotropic hyperelastic materials

Isotropic hyperelastic materials are described in terms of a "strain energy potential," which defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the strain at that point in the material.
Several forms of strain energy potentials are available in Abaqus to model approximately incompressible isotropic elastomers. For more information on hyperelastic materials, see *Hyperelastic Behavior of Rubberlike Materials*.

When you define a isotropic hyperelastic material, you have the option of either specifying material parameters directly or allowing Abaqus to calculate them from test data that you provide. For detailed instructions, see the following sections:

- *Entering material parameters to define an isotropic hyperelastic material*
- *Providing test data to define an isotropic hyperelastic material*

Entering material parameters to define an isotropic hyperelastic material

You can provide the parameters of the hyperelastic strain energy potentials directly as functions of temperature.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hyperelastic**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Choose **Isotropic** as the material type.
3. Click the arrow to the right of the **Strain energy potential** field, and select the strain energy potential of your choice.

Arruda-Boyce

The Arruda-Boyce model is also known as the eight-chain model. For more information, see *Arruda-Boyce Form*.

Marlow

For more information, see *Marlow Form*.

Mooney-Rivlin

The Mooney-Rivlin model is equivalent to using the polynomial model with N=1. For more information, see *Mooney-Rivlin Form*.

Neo Hooke

The Neo Hookean model is equivalent to using the reduced polynomial model with N=1. For more information, see *Neo-Hookean Form*.

Ogden

For more information, see *Ogden Form*.

Polynomial

For more information, see *Polynomial Form*.

Reduced Polynomial

The reduced polynomial model is equivalent to using the polynomial model with $C_{ij} = 0$ for $j \neq 0$. For more information, see *Reduced Polynomial Form*.

User-defined

You can define the derivatives of the strain energy potential with respect to the strain invariants in user subroutine *UHYPER*. This method is valid only for Abaqus/Standard analyses. For more information, see *User Subroutine Specification in Abaqus/Standard*.

Van der Waals

The Van der Waals model is also known as the Kilian model. For more information, see *Van Der Waals Form*.

Yeoh

The Yeoh model is equivalent to using the reduced polynomial model with N=3. For more information, see *Yeoh Form*.

Unknown

If you define an isotropic hyperelastic material using experimental data, you also have the option of temporarily leaving the particular strain energy potential unspecified. You can use the **Evaluate** option to identify the optimal strain energy potential for the material data and display the material editor again to complete the material definition; see *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*, for more information.

4. Select **Coefficients** as the **Input source**. This **Input Source** option is invalid for the Marlow model or for an unknown strain energy potential.
5. If you are defining the hyperelastic behavior of a viscoelastic material, click the arrow to the right of the **Moduli time scale (for viscoelasticity)** field to specify either long-term or instantaneous elastic response.
6. If you selected **User-defined** as the strain energy potential, perform the following steps:
   - Toggle on **Include compressibility** to indicate that the material defined by user subroutine *UHYPER* is compressible. Otherwise, Abaqus assumes the material is incompressible.
   - Specify the **Number of property values** needed as data in user subroutine *UHYPER*.
7. If you selected **Ogden**, **Polynomial**, or **Reduced Polynomial** as the strain energy potential, click the arrows to the left of the **Strain energy potential order** field to select a value.
8. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
9. Enter the material properties in the **Data** table corresponding to the chosen strain energy potential.

Arruda-Boyce

Enter μ, λ_m, and D.

Mooney-Rivlin

Enter C_{10}, C_{01}, and D_1.

Neo Hooke

Enter C_{10} and D_1.

Ogden

Enter μ_i, α_i, and D_i, where *i* goes from 1 to N and N is the value specified for the **Strain energy potential order**.

Polynomial

Enter C_{ij}, where $i + j$ goes from 1 to N, and D_i, where *i* goes from 1 to N, and N is the value specified for the **Strain energy potential order**.

Reduced Polynomial

Enter C_{i0} and D_i, where *i* goes from 1 to N and N is the value specified for the **Strain energy potential order**.

Van der Waals

Enter μ, λ_m, a, β, and D.

Yeoh

Enter C_{10}, C_{20}, C_{30}, D_1, D_2, and D_3.

10. If desired, select **Hysteresis** from the **Suboptions** menu to define hysteretic behavior. See *Defining hysteretic behavior for an isotropic hyperelastic material model*" for details.
11. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Providing test data to define an isotropic hyperelastic material

Abaqus can calculate material parameters from test data that you enter in the **Test Data Editor**.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hyperelastic**. (For information on displaying the **Edit Material** dialog box, see *Creating and editing materials*.)
2. Choose **Isotropic** as the material type.
3. Click the arrow to the right of the **Strain energy potential** field, and select the strain energy potential of your choice.

Arruda-Boyce

The Arruda-Boyce model is also known as the eight-chain model. For more information, see *Arruda-Boyce Form*.

Marlow

For more information, see *Marlow Form*.

Mooney-Rivlin

The Mooney-Rivlin model is equivalent to using the polynomial model with N=1. For more information, see *Mooney-Rivlin Form*.

Neo Hooke

The Neo Hookean model is equivalent to using the reduced polynomial model with N=1. For more information, see *Neo-Hookean Form*.

Ogden

For more information, see *Ogden Form*.

Polynomial

For more information, see *Polynomial Form*.

Reduced Polynomial

The reduced polynomial model is equivalent to using the polynomial model with $C_{ij} = 0$ for $j \neq 0$. For more information, see *Reduced Polynomial Form*.

User-defined

You can define the derivatives of the strain energy potential with respect to the strain invariants in user subroutine *UHYPER*. This method is valid only for Abaqus/Standard analyses. For more information, see *User Subroutine Specification in Abaqus/Standard*.

Van der Waals

The Van der Waals model is also known as the Kilian model. For more information, see *Van Der Waals Form*.

Yeoh

The Yeoh model is equivalent to using the reduced polynomial model with N=3. For more information, see *Yeoh Form*.

Unknown

If you define an isotropic hyperelastic material using experimental data, you also have the option of temporarily leaving the particular strain energy potential unspecified. You can use the **Evaluate** option to identify the optimal strain energy potential for the material data and then display the material editor again to complete the material definition; see *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*, for more information.

4. Select **Test data** as the **Input source** to indicate that the material constants are to be computed from data taken from simple tests on a material specimen.
5. If you are defining the hyperelastic behavior of a viscoelastic material, click the arrow to the right of the **Moduli time scale (for viscoelasticity)** field to specify either long-term or instantaneous elastic response.
6. If you selected **Marlow** as the strain energy potential, select the **Data to define deviatoric response** and the **Data to define volumetric response** options of your choice.
   - The deviatoric response is defined by the **Uniaxial**, **Biaxial**, or **Planar** test data specified as described in Step 8.
   - The volumetric response is defined by one of the following methods:
     - **Ignore test data**: Abaqus/Standard assumes fully incompressible behavior, while Abaqus/Explicit assumes compressibility corresponding to a Poisson's ratio of 0.475.
     - **Volumetric test data**: The volumetric test data are specified directly, as described in Step 8.
     - **Poisson's ratio**: Specify a value for the Poisson's ratio of the isotropic hyperelastic material.
     - **Lateral nominal strain**: Lateral nominal strains are specified as part of the uniaxial, biaxial, or planar test data, as described in Step 8.
7. If you selected **Ogden**, **Polynomial**, or **Reduced Polynomial** as the strain energy potential, click the arrows to the left of the **Strain energy potential order** field to select a value.
8. If you selected **Van der Waals** as the strain energy potential, choose the method for specifying **Beta**:
   - Select **Fitted value** to determine the value of β from a nonlinear least-squares fit of the test data.
   - Select **Specify**, and enter a value to specify β directly. Allowable values are $\mathbf{0 \leq \beta \leq 1}$. It is recommended to set β=0 if only one type of test data is available.
9. You can specify the experimental stress-strain data for as many as four simple tests: uniaxial, equibiaxial, planar, and, if the material is compressible, a volumetric compression test. Use the **Test Data** menu to specify the experimental data. For details, see the following sections:
   - *Specifying uniaxial test data for an isotropic hyperelastic material model*"

- *Specifying biaxial test data for an isotropic hyperelastic material model*"
- *Specifying planar test data for an isotropic hyperelastic material model*"
- *Specifying volumetric test data for an isotropic hyperelastic material model*"

10. If desired, select **Hysteresis** from the **Suboptions** menu to define hysteretic behavior. See *Defining hysteretic behavior for an isotropic hyperelastic material model*" for details.
11. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying uniaxial test data for an isotropic hyperelastic material model

Use the **Test Data Editor** to specify uniaxial test data from which Abaqus can calibrate hyperelastic material coefficients. For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

1. Create an isotropic hyperelastic material model as described in *Providing test data to define an isotropic hyperelastic material*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Uniaxial Test Data**.

   A **Test Data Editor** appears.
3. Toggle on **Apply smoothing** if you want Abaqus to apply a smoothing filter to the stress-strain data. This option is particularly recommended if you are using the Marlow model.
4. If you have requested data smoothing, click the arrows to the right of the **Apply smoothing** field to specify the number of data points to the right and left of each data point within which Abaqus will fit a least-squares polynomial.
5. If you are defining a Marlow model, you can select the following options:
   - To include lateral nominal strain data, toggle on **Include lateral nominal strain**.

     A column labeled **Lateral Nominal Strain** appears in the **Data** table.
   - To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

     A column labeled **Temp** appears in the **Data** table.
   - To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

     Field variable columns appear in the **Data** table.
6. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, T_U.

   #### Nominal Strain

   Nominal strain, ϵ_U.

   #### Lateral Nominal Strain

   Nominal lateral strain, $\epsilon_2 = \epsilon_3$.

   #### Temp

   Temperature, θ.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Specifying biaxial test data for an isotropic hyperelastic material model

Use the **Test Data Editor** to specify biaxial test data from which Abaqus can calibrate hyperelastic material coefficients. For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

1. Create an isotropic hyperelastic material model as described in *Providing test data to define an isotropic hyperelastic material*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Biaxial Test Data**.

   A **Test Data Editor** appears.
3. Toggle on **Apply smoothing** if you want Abaqus to apply a smoothing filter to the stress-strain data. This option is particularly recommended if you are using a Marlow model.
4. If you have requested data smoothing, click the arrows to the right of the **Apply smoothing** field to specify the number of data points to the right and left of each data point within which Abaqus will fit a least-squares polynomial.
5. If you are defining a Marlow model, you can select the following options:
   - To include lateral nominal strain data, toggle on **Include lateral nominal strain**.

     A column labeled **Lateral Nominal Strain** appears in the **Data** table.
   - To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

     A column labeled **Temp** appears in the **Data** table.
   - To define behavior data that depend on field variables, click the arrow to the right of the **Number of field variables** field to increase or decrease the number of field variables.

     Field variable columns appear in the **Data** table.
6. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, $\boldsymbol{T_B}$.

   #### Nominal Strain

   Nominal strain, $\boldsymbol{\epsilon_B}$.

   #### Lateral Nominal Strain

   Nominal lateral strain, $\boldsymbol{\epsilon_3}$.

   #### Temp

   Temperature, $\boldsymbol{\theta}$.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Specifying planar test data for an isotropic hyperelastic material model

Use the **Test Data Editor** to specify planar test data from which Abaqus can calibrate hyperelastic material coefficients. For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

1. Create an isotropic hyperelastic material model as described in *Providing test data to define an isotropic hyperelastic material*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Planar Test Data**.

   A **Test Data Editor** appears.
3. Toggle on **Apply smoothing** if you want Abaqus to apply a smoothing filter to the stress-strain data. This option is particularly recommended if you are using a Marlow model.
4. If you have requested data smoothing, click the arrows to the right of the **Apply smoothing** field to specify the number of data points to the right and left of each data point within which Abaqus will fit a least-squares polynomial.
5. If you are defining a Marlow model, you can select the following options:
   - To include lateral nominal strain data, toggle on **Include lateral nominal strain**.

     A column labeled **Lateral Nominal Strain** appears in the **Data** table.
   - To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

     A column labeled **Temp** appears in the **Data** table.
   - To define behavior data that depend on field variables, click the arrow to the right of the **Number of field variables** field to increase or decrease the number of field variables.

     Field variable columns appear in the **Data** table.
6. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, T_S.

   #### Nominal Strain

   Nominal strain, ϵ_S.

   #### Lateral Nominal Strain

   Nominal lateral strain, ϵ_3.

   #### Temp

   Temperature, θ.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Specifying volumetric test data for an isotropic hyperelastic material model

Use the **Test Data Editor** to specify volumetric test data from which Abaqus can calibrate hyperelastic material coefficients. For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

1. Create an isotropic hyperelastic material model as described in *Providing test data to define an isotropic hyperelastic material*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Volumetric Test Data**.

   A **Test Data Editor** appears.
3. Toggle on **Apply smoothing** if you want Abaqus to apply a smoothing filter to the stress-strain data. This option is particularly recommended if you are using a Marlow model.
4. If you have requested data smoothing, click the arrows to the right of the **Apply smoothing** field to specify the number of data points to the right and left of each data point within which Abaqus will fit a least-squares polynomial.
5. If you are defining a Marlow model, you can select the following options:

   - To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

     A column labeled **Temp** appears in the **Data** table.
   - To define behavior data that depend on field variables, click the arrow to the right of the **Number of field variables** field to increase or decrease the number of field variables.

     Field variable columns appear in the **Data** table.
6. In the **Data** table, enter the test data:

   #### Pressure

   Total pressure stress, p.

   #### Volume Ratio

   Volume ratio (current volume/original volume), J.

   #### Temp

   Temperature, θ.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining hysteretic behavior for an isotropic hyperelastic material model

Use the **Suboption Editor** to define the strain-rate-dependent response of an isotropic hyperelastic material that exhibits pronounced hysteresis under cyclic loading. For more information, see *Hysteresis in Elastomers*.

1. Create an isotropic hyperelastic material model as described in *Entering material parameters to define an isotropic hyperelastic material*" or *Providing test data to define an isotropic hyperelastic material*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Hysteresis**.

   The **Suboption Editor** appears.

   In the **Data** table of the **Suboption Editor**, enter the creep behavior data:

   #### Stress Scaling Factor

   Stress scaling factor, S.

   #### Creep Parameter

   Creep parameter, A.

   #### Eff Stress Exponent

   Effective stress exponent, m.

   #### Creep Strain Exponent

   Creep strain exponent, C.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
3. Click **OK** to return to the **Edit Material** dialog box.

Creating an anisotropic hyperelastic material model

The anisotropic hyperelastic model provides a modeling capability for materials that exhibit highly anisotropic and nonlinear elastic behavior, such as biomedical soft tissues and fiber-reinforced elastomers. For more information, see *Anisotropic Hyperelastic Behavior*.

In Abaqus/CAE the local direction vectors of the material are orthogonal and align with the axes of the assigned material orientation. The best practice is to assign the orientation using discrete orientations in Abaqus/CAE. For information about defining discrete orientations, see *Using discrete orientations for material orientations and composite layup orientations*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hyperelastic**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Choose **Anisotropic** as the material type.
3. Click the arrow to the right of the **Strain energy potential** field, and select the strain energy potential of your choice.

Fung-Anisotropic

For the fully anisotropic strain-based Fung model, you must specify 21 independent components b_{ijkl}. For more information, see *Generalized Fung Form*.

Fung-Orthotropic

For the orthotropic strain-based Fung model, you must specify 9 independent components b_{ijkl}. For more information, see *Generalized Fung Form*.

Holzapfel

This form of invariant-based strain energy potential is used for modeling arterial layers with distributed collagen fiber orientations. For more information, see *Holzapfel-Gasser-Ogden Form*.

User

You can use a user subroutine to define the form of a strain-based or invariant-based strain energy potential directly. For more information, see *User-Defined Form: Strain-Based*, and *User-Defined Form: Invariant-Based*.

4. To define material parameters that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
   Field variable columns appear in the data table.
5. For the **Fung-Anisotropic**, **Fung-Orthotropic**, and **Holzapfel** forms of strain energy potential, toggle on **Use temperature-dependent coefficients** to define material parameters that depend on temperature.
   A column labeled **Temp** appears in the data table.
6. If you are defining the hyperelastic behavior of a viscoelastic material, click the arrow to the right of the **Moduli** field to specify either **Long-term** or **Instantaneous** elastic response. See *Viscoelasticity*, for more information.
7. For the Holzapfel strain energy potential, click the arrows to the right of the **Number of local directions** field to increase or decrease the number of preferred local directions (or fiber directions) in the material. The default (and minimum) is 1. See *Creating an anisotropic hyperelastic material model* below and *Holzapfel-Gasser-Ogden Form*, for more information.

8. For a user-defined strain energy potential, you must specify the following options:
   - Choose **Strain** or **Invariant** as the formulation defined by your user subroutine.
   - Choose **Incompressible** or **Compressible** as the type of material defined by your user subroutine. See *Compressibility*, for more information.
   - Specify the **Number of property values** needed as data in your user subroutine.

9. Enter the material parameters in the data table corresponding to the chosen strain energy potential.

Fung-Anisotropic

Enter b_{1111}, b_{1122}, b_{2222}, b_{1133}, b_{2233}, b_{3333}, b_{1112}, b_{2212}, b_{3312}, b_{1212}, b_{1113}, b_{2213}, b_{3313}, b_{1213}, b_{1313}, b_{1123}, b_{2223}, b_{3323}, b_{1223}, b_{1323}, b_{2323}, c (units of FL^{-2}), and D (units of $F^{-1}L^2$).

Fung-Orthotropic

Enter b_{1111}, b_{1122}, b_{2222}, b_{1133}, b_{2233}, b_{3333}, b_{1212}, b_{1313}, b_{2323}, c (units of FL^{-2}), and D (units of $F^{-1}L^2$).

Holzapfel

Enter C_{10} (units of FL^{-2}), D (units of $F^{-1}L^2$), k_1 (units of FL^{-2}), k_2, and the fiber dispersion parameter κ ($0 \le \kappa \le 1/3$).

You may need to expand the dialog box to see all the columns in the data table. For detailed information on how to enter data, see *Entering tabular data*.

10. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Creating a hyperfoam material model

You can create a hyperfoam material model to describe a cellular solid whose porosity permits very large volumetric changes.
See *Hyperelastic Behavior in Elastomeric Foams* for more information on hyperfoam materials.

In this section:

- *Defining hyperfoam materials*
- *Entering material parameters to define a hyperelastic foam material*
- *Providing test data to define a hyperelastic foam material model*
- *Specifying uniaxial test data for a hyperelastic foam material model*
- *Specifying biaxial test data for a hyperelastic foam material model*
- *Specifying simple shear test data for a hyperelastic foam material model*
- *Specifying planar test data for a hyperelastic foam material model*
- *Specifying volumetric test data for a hyperelastic foam material model*

Defining hyperfoam materials

When you define a hyperfoam material, you have the option of either specifying material parameters directly or allowing Abaqus to calculate them from test data that you provide.

For detailed instructions, see the following sections:

- *Entering material parameters to define a hyperelastic foam material*
- *Providing test data to define a hyperelastic foam material model*

See *Hyperelastic Behavior in Elastomeric Foams* for more information on hyperfoam materials.

Entering material parameters to define a hyperelastic foam material

You can provide the parameters of the hyperelastic foam strain energy potentials directly as functions of temperature.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hyperfoam**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrows to the right of the **Strain energy potential order** field to increase or decrease the order of the strain energy potential, N.
3. To specify material parameters that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. If you are defining the hyperfoam behavior of a viscoelastic material, click the arrow to the right of the **Moduli time scale (for viscoelasticity)** field to specify either long-term or instantaneous elastic response.
5. In the **Data** table, enter the material parameters: :

   #### mu*i*, alpha*i*, and nu*i*

   Material parameters μ_i, α_i, and ν_i.

   #### Temp

   Temperature.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Providing test data to define a hyperelastic foam material model

Abaqus can calculate material parameters from test data that you enter in the **Test Data Editor**.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hyperfoam**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use test data (Suboptions must be specified)**.
3. Click the arrows to the right of the **Strain energy potential order** field to increase or decrease the order of the strain energy potential, N.
4. Indicate how you want to define Poisson effects:
   - Toggle on **Use constant Poisson's ratio** if you want to enter a Poisson's ratio that will remain constant throughout the calculations.
   - Toggle off **Use constant Poisson's ratio** if you want Poisson effects defined from volumentric test data and/or lateral strains in other test data.
5. If you are defining the hyperfoam behavior of a viscoelastic material, click the arrow to the right of the **Moduli time scale (for viscoelasticity)** field to specify either long-term or instantaneous elastic response.
6. Use the **Test Data** menu to specify the experimental data from which Abaqus can calculate material parameters. For detailed instructions, see the following sections:
   - *Specifying uniaxial test data for a hyperelastic foam material model*"
   - *Specifying biaxial test data for a hyperelastic foam material model*"
   - *Specifying simple shear test data for a hyperelastic foam material model*"
   - *Specifying planar test data for a hyperelastic foam material model*"
   - *Specifying volumetric test data for a hyperelastic foam material model*"
7. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying uniaxial test data for a hyperelastic foam material model

Use the **Test Data Editor** to specify uniaxial test data from which Abaqus can calibrate hyperelastic foam material coefficients. For more information, see *Hyperelastic Behavior in Elastomeric Foams*.

1. Create a hyperelastic foam material model as described in *Providing test data to define a hyperelastic foam material model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Uniaxial Test Data**.

   A **Test Data Editor** appears.
3. In the **Data** table, enter the uniaxial test data:

   #### Nominal Stress

   Nominal stress, T_U.

   #### Nominal Strain

   Nominal strain, ϵ_U.

   #### Lateral Nominal Strain

   Nominal lateral strain, $\epsilon_2 = \epsilon_3$. This parameter is unnecessary if you have already entered a constant Poisson's ratio in the **Edit Material** dialog box.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Specifying biaxial test data for a hyperelastic foam material model

Use the **Test Data Editor** to specify biaxial test data from which Abaqus can calibrate hyperfoam material coefficients. For more information, see *Hyperelastic Behavior in Elastomeric Foams*.

1. Create a hyperelastic foam material model as described in *Providing test data to define a hyperelastic foam material model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Biaxial Test Data**.

   A **Test Data Editor** appears.
3. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, T_L.

   #### Nominal Strain

   Nominal strain, ϵ_B.

   #### Lateral Nominal Strain

   Nominal transverse strain, ϵ_3. This parameter is unnecessary if you have already entered a constant Poisson's ratio in the **Edit Material** dialog box.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Specifying simple shear test data for a hyperelastic foam material model

Use the **Test Data Editor** to specify simple shear test data from which Abaqus can calibrate hyperfoam material coefficients. For more information, see *Hyperelastic Behavior in Elastomeric Foams*.

1. Create a hyperelastic foam material model as described in *Providing test data to define a hyperelastic foam material model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Simple Shear Test Data**.

   A **Test Data Editor** appears.
3. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal shear stress, T_S.

   #### Nominal Strain

   Nominal shear strain, γ.

   #### Nominal transverse stress

   Nominal transverse stress, T_T (normal to edge with shear stress). This stress value is optional, but strongly recommended. If given, a more accurate material response will be obtained.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Specifying planar test data for a hyperelastic foam material model

Use the **Test Data Editor** to specify planar test data from which Abaqus can calibrate hyperfoam material coefficients. Planar test data alone is inadequate; you must also include uniaxial and/or biaxial test data in the material definition. See the following sections for more information:

- *Hyperelastic Behavior in Elastomeric Foams*
- *Specifying uniaxial test data for a hyperelastic foam material model*"
- *Specifying biaxial test data for a hyperelastic foam material model*"

1. Create a hyperelastic foam material model as described in *Providing test data to define a hyperelastic foam material model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Planar Test Data**.

   A **Test Data Editor** appears.
3. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, T_L.

   #### Nominal Strain

   Nominal strain in the direction of loading, ϵ_P.

   #### Nominal Lateral Strain

   Nominal transverse strain, ϵ_3. This parameter is unnecessary if you have already entered a constant Poisson's ratio in the **Edit Material** dialog box.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Specifying volumetric test data for a hyperelastic foam material model

Use the **Test Data Editor** to specify volumetric test data from which Abaqus can calibrate hyperfoam material coefficients. For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

1. Create a hyperelastic foam material model as described in *Providing test data to define a hyperelastic foam material model*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Volumetric Test Data**.

   A **Test Data Editor** appears.
3. In the **Data** table, enter the test data:

   #### Pressure

   Pressure, p.

   #### Volume Ratio

   Volume ratio (current volume/original volume), J.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Creating a low-density foam material model

You can create a material model to describe a low-density, highly compressible elastomeric foam with significant rate-sensitive behavior (such as polyurethane foam).
Abaqus calculates material parameters from test data that you enter in the **Test Data Editor**. You must provide uniaxial test data for both tension and compression. Your test data must specify the uniaxial stress-strain curve for different strain-rate values.

In this section:

- *Specifying low-density foam material properties*
- *Specifying uniaxial tension test data for a low-density foam material model*
- *Specifying uniaxial compression test data for a low-density foam material model*

Specifying low-density foam material properties

Enter the properties in the **Edit Material** dialog box, as described below. See *Low-Density Foams*, for more information on low-density foam materials.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Low Density Foam**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Select the **Strain rate measure**.

   - Choose **Volumetric** (default) to use the nominal volumetric strain rate that does not produce rate-sensitive behavior under volume-preserving deformation modes (e.g., simple shear).
   - Choose **Principal** to cause Abaqus to use the strain rate evaluated along each principal direction.

   See *Strain Rate in Abaqus/Explicit*, for more details.

3. Toggle on **Extrapolate stress-strain curve beyond maximum strain rate** to activate strain-rate extrapolation based on slope (with respect to the strain rate). See *Extrapolation of Stress-Strain Curves*, for more details.
4. Toggle on **Maximum allowable principal tensile stress** to enter a cutoff value that the foam material can sustain. The maximum principal tensile stresses computed by Abaqus will be forced to stay at or below this value. See *Tension Cutoff and Failure in Abaqus/Explicit*, for more details.
5. If you specified a value for the **Maximum allowable principal tensile stress**, toggle on **Remove elements exceeding maximum** to cause Abaqus to delete any elements in which the maximum principal tensile stress is reached. This is a simple method for modeling rupture.
6. You can accept the default values for **Relaxation coefficients** or enter new values for **mu0**, **mu1**, and **alpha**. See *Relaxation Coefficients in Abaqus/Explicit*, for a detailed description of these material parameters.
7. To define the uniaxial test data for the material, click **Uniaxial Test Data**. For details, see the following sections:

   - *Specifying uniaxial tension test data for a low-density foam material model*"
   - *Specifying uniaxial compression test data for a low-density foam material model*"

8. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying uniaxial tension test data for a low-density foam material model

Use the **Test Data Editor** to specify uniaxial test data for tension.

1. Create a low-density foam material model as described in *Specifying low-density foam material properties*."
2. From the **Uniaxial Test Data** menu in the **Edit Material** dialog box, select **Uniaxial Tension Test Data**.

   A **Test Data Editor** appears.
3. To specify test data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To specify test data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. In the **Data** table, enter the test data:

   #### Nominal Stress

   Nominal stress, T_U.

   #### Nominal Strain

   Nominal strain, ϵ_U.

   #### Nominal Strain Rate

   Nominal strain rate, $\dot{\epsilon}_U$. (Provide positive values to specify the loading response and negative values to specify unloading.)

   #### Temp

   Temperature, θ.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Specifying uniaxial compression test data for a low-density foam material model

Use the **Test Data Editor** to specify uniaxial test data for compression.

1. Create a low-density foam material model as described in *Specifying low-density foam material properties*."
2. From the **Uniaxial Test Data** menu in the **Edit Material** dialog box, select **Uniaxial Compression Test Data**.

   A **Test Data Editor** appears.
3. To specify test data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To specify test data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. In the **Data** table, enter the test data:

   ### Nominal Stress

   Enter the absolute value of nominal stress, T_U.

   ### Nominal Strain

   Enter the absolute value of nominal strain, ϵ_U.

   ### Nominal Strain Rate

   Nominal strain rate, $\dot{\epsilon}_U$. (Provide positive values to specify the loading response and negative values to specify unloading.)

   ### Temp

   Temperature, θ.

   ### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Creating a hypoelastic material model

You can create a hypoelastic material definition to describe a nonlinear, small-strain elastic material. You have the option of either entering hypoelastic material parameters directly in the material editor or defining material parameters in user subroutine *UHYPEL*. User subroutine *UHYPEL* allows you to specify temperature-dependent data.

For more information, see the following sections:

- *Hypoelastic Behavior*
- *UHYPEL*

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Hypoelastic**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Select one of the following options for specifying material parameters:
   - Toggle on **Use user subroutine UHYPEL**.
   - Enter material parameters in the **Data** area:

Young's Modulus

Instantaneous Young's modulus, *E*.

Poisson's Ratio

Instantaneous Poisson's ratio, ν.

I1

First strain invariant, I_1.

I2

Second strain invariant, I_2.

I3

Third strain invariant, I_3.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

3. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Creating a porous elastic material model

A porous elastic material model defines the elastic parameters for a porous material. For more information, see *Elastic Behavior of Porous Materials*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Porous Elastic**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Shear** field and select an option for defining deviatoric elastic behavior in the porous material:
   - Select **G** to specify a constant shear modulus. In this case the shear behavior is unaffected by compaction of the material.

     A **Shear Modulus** column appears in the **Data** table.
   - Select **Poisson** to allow Abaqus to calculate the instantaneous shear modulus from the bulk modulus and Poisson's ratio. In this case, the elastic shear stiffness increases as the material is compacted.

     A **Poisson's ratio** column appear in the **Data** table.
3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.
   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
   Field variable columns appear in the **Data** table.
5. In the **Data** table, enter the material parameters:

Log Bulk Modulus

Logarithmic bulk modulus, κ. (Dimensionless.)

G

Shear modulus, G. Enter this value if you selected **G** from the list of **Shear** options.

Poisson's ratio

Poisson's ratio, ν. Enter this value if you selected **Poisson** from the list of **Shear** options.

Tensile Limit

Elastic tensile limit, p_t^{el}. (This value cannot be negative.)

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Creating a viscoelastic material model

You can define viscoelastic behavior in a material as a function of frequency (for steady-state, small-vibration analyses) or as a function of reduced time (for time-dependent analyses).
For more information, see the following sections:

- *Time Domain Viscoelasticity*
- *Frequency Domain Viscoelasticity*

In this section:

- *Defining viscoelastic materials*
- *Defining time domain viscoelasticity*
- *Specifying thermorheologically simple (TRS) temperature dependence for time domain viscoelasticity*
- *Defining frequency domain viscoelasticity*
- *Specifying shear test data*
- *Specifying volumetric test data*
- *Specifying combined test data*

Defining viscoelastic materials

You can define viscoelasticity in the **Edit Material** dialog box.

For detailed instructions, see the following sections:

- *Defining time domain viscoelasticity*
- *Defining frequency domain viscoelasticity*

For more information on viscoelasticity, see *Time Domain Viscoelasticity* and *Defining frequency domain viscoelasticity*.

Defining time domain viscoelasticity

This type of material model describes the isotropic rate-dependent behavior of materials for which the dissipative losses caused by internal damping effects must be modeled in the time domain.

Abaqus assumes that the time domain viscoelasticity is defined by a Prony series expansion. You can specify the Prony series parameters directly for each term in the Prony series. Alternatively, Abaqus can calculate the terms in the Prony series using time-dependent creep test data, time-dependent relaxation test data, or frequency-dependent cyclic test data that you provide.

For more information on time domain viscoelasticity, see *Time Domain Viscoelasticity*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Viscoelastic**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Domain** field, and select **Time**.
3. Click the arrow to the right of the **Time** field, and select the option of your choice for determining viscoelastic material parameters:
   - Select **Prony** if you want to enter the Prony series parameters for each term directly.
   - Select **Creep test data** if you want Abaqus to calculate the parameters in the Prony series from creep test data that you provide. If you select this option, you must enter shear test data and/or volumetric test data in the **Test Data Editor**.
   - Select **Relaxation test data** if you want Abaqus to calculate the Prony series parameters from relaxation test data. If you select this option, you must enter shear test data and/or volumetric test data in the **Test Data Editor**.
   - Select **Frequency data** if you want Abaqus to calculate the Prony series parameters from frequency-dependent cyclic test data.
4. If you selected any of the test data options from the **Time** option list, you can specify two additional parameters related to the calibration of Prony series parameters:
   - Click the arrows to the right of the **Maximum number of terms in the Prony series** field to specify the maximum number of terms (*N*) in the Prony series. Abaqus will perform the least-squares fit from $N = 1$ to $N =$ NMAX until convergence is achieved for the lowest *N* with respect to the error tolerance.
   - In the **Allowable average root-mean-square error** field, enter the error tolerance for the data points in the least-squares fit.
5. If you selected **Prony** from the list of **Time** options, enter the Prony parameters in the **Data** table.

g_i Prony

Shear relaxation or shear traction relaxation modulus ratio, $\overline{g}_i^P$.

k_i Prony

Bulk relaxation or normal traction relaxation modulus ratio, $\overline{k}_i^P$.

tau_i Prony

Relaxation time, τ_i.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. If you selected **Frequency data** from the list of **Time** options, enter the frequency-dependent test data in the **Data** table.

Omega g* real

Real part of ωg^*.

Omega g* imag

Imaginary part of ωg^*.

Omega k* real

Real part of ωk^*.

Omega k* imag

Imaginary part of ωk^*.

Frequency

Frequency, *f*, in cycles per time.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If applicable, click **Test Data** to specify test data from which to define viscoelastic behavior. See the following sections for details:
   - *Specifying shear test data*"
   - *Specifying volumetric test data*"
   - *Specifying combined test data*"
8. To specify thermorheologically simple (TRS) temperature effects, use the **Suboptions** menu. For details, see *Specifying thermorheologically simple (TRS) temperature dependence for time domain viscoelasticity*.
9. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying thermorheologically simple (TRS) temperature dependence for time domain viscoelasticity

You can define the shift function either by entering parameters for the Williams-Landel-Ferry approximation or by using user subroutine *UTRS*. For more information on temperature effects in time domain viscoelasticity, see *Time Domain Viscoelasticity*.

1. Create a time domain viscoelastic material as described in *Defining time domain viscoelasticity*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Trs**.
   The **Suboption Editor** appears.
3. Click the arrow to the right of the **Shift function** field, and select the option of your choice.
   - Select **WLF** to define the shift function using the Williams-Landel-Ferry approximation.
   - Select **User Subroutine UTRS** to define the shift function using user subroutine *UTRS*.
4. If you selected **WLF** from the list of **Shift** function options, enter the required data in the **Data** table:

   #### Theta 0

   Reference temperature, θ_0.

   #### C1

   Calibration constant, C_1.

   #### C2

   Calibration constant, C_2.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Defining frequency domain viscoelasticity

This type of material model describes frequency-dependent material behavior in small, steady-state harmonic oscillations. In these cases the dissipative losses caused by internal damping effects must be modeled in the frequency domain.

The dissipative part of the material behavior is defined by giving the real and imaginary parts of g^* and k^* (for compressible materials) as functions of frequency. The moduli can be defined as functions of the frequency in one of three ways: by a power law, by tabular input, or by a Prony series expression for the shear and bulk relaxation moduli.

For more information on frequency domain viscoelasticity, see *Frequency Domain Viscoelasticity*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Elasticity->Viscoelastic**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Domain** field, and select **Frequency**.
3. Click the arrow to the right of the **Frequency** field, and select the option of your choice for determining viscoelastic material parameters:
   - Select **Formula** to define the frequency dependence by the power law formulae.
   - Select **Tabular** to define the frequency response in tabular form. You must provide the real and imaginary parts of ωg^* and ωk^*—where ω is the circular frequency—as functions of frequency in cycles per time.
   - Select **Prony** if you want Abaqus to calculate the frequency dependence from a time domain Prony series description of the dimensionless shear and bulk relaxation moduli.
   - Select **Creep test data** if you want Abaqus to calculate the parameters in the Prony series from creep test data that you provide. If you select this option, you must enter shear test data and/or volumetric test data in the **Test Data Editor**.
   - Select **Relaxation test data** if you want Abaqus to calculate the Prony series parameters from relaxation test data. If you select this option, you must enter shear test data and/or volumetric test data in the **Test Data Editor**.
4. If you selected **Creep test data** or **Relaxation test data** from the **Frequency** option list, you can specify two additional parameters related to the calibration of Prony series parameters:
   - Click the arrows to the right of the **Maximum number of terms in the Prony series** field to specify the maximum number of terms (*N*) in the Prony series. Abaqus will perform the least-squares fit from $N = 1$ to $N =$ NMAX until convergence is achieved for the lowest *N* with respect to the error tolerance.
   - In the **Allowable average root-mean-square error** field, enter the error tolerance of the data points in the least-squares fit.
5. If you selected **Tabular** from the **Frequency** option list, you can specify two additional options:

Type

This parameter specifies whether you are defining continuum material properties or effective thickness-direction gasket properties.

Select **Isotropic** if you are defining continuum material properties. This choice is appropriate when the viscoelasic material model is used for any continuum, structural, or special-purpose elements whose material response is modeled using continuum material properties.

Select **Traction** if you are defining effective thickness-direction gasket properties. This option is supported only for gasket elements whose behavior is modeled directly using a gasket behavior model.

Preload

This parameter specifies the nature of preload used for defining frequency domain viscoelastic material properties or effective thickness-direction gasket properties.

Select **Uniaxial** to specify that the material properties correspond to a uniaxial test.

Select **Volumetric** to specify that the material properties correspond to a volumetric test. This setting cannot be used to define effective thickness-direction gasket properties.

Select **Uniaxial and Volumetric** to specify that the material properties correspond to both types of tests. This setting cannot be used to define effective thickness-direction gasket properties.

Select **None** if you choose not to specify a preload parameter.

6. If you selected **Formula** from the list of **Frequency** options, enter the following data in the **Data** table:

g1*real

Real part of g_1^*.

g1*imag

Imaginary part of g_1^*.

a

Value of a.

k1*real

Real part of k_1^*.

k1*imag

Imaginary part of k_1^*. If the material is incompressible, this value is ignored.

b

Value of b. If the material is incompressible, this value is ignored.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected **Tabular** from the list of **Frequency** options, enter the data relevant to your **Type** and **Preload** selections (not all of the following parameters will apply):

Omega g* real

Real part of ωg^*. $(\omega \Re(g^*) = G_\ell / G_\infty)$

Omega g* imag

Imaginary part of ωg^*. $(\omega \mathfrak{I}(g^*) = 1 - G_s/G_\infty)$

Omega k* real

Real part of ωk^*. $(\omega \mathfrak{R}(k^*) = \mathbf{K}_\ell/\mathbf{K}_\infty)$ If the material is incompressible, this value is ignored.

Omega k* imag

Imaginary part of ωk^*. $(\omega \mathfrak{I}(k^*) = 1 - \mathbf{K}_s/\mathbf{K}_\infty)$ If the material is incompressible, this value is ignored.

Frequency

Frequency, *f*, in cycles per time.

Loss Modulus

Uniaxial or bulk loss modulus.

Storage Modulus

Uniaxial or bulk storage modulus.

Uniaxial Strain

Uniaxial nominal strain (defines the level of uniaxial preload).

Volume Ratio

Volume ratio, *J* (current volume/original volume; defines the level of volumetric preload).

Normalized Loss Modulus

Real part of ωk^*. $(\omega \mathfrak{R}(k^*) = \mathbf{K}_\ell/\mathbf{K}_\infty)$ for thickness-direction gasket behavior with no preload.

Normalized Shear Modulus

Imaginary part of ωk^*. $(\omega \mathfrak{I}(k^*) = 1 - \mathbf{K}_s/\mathbf{K}_\infty)$ for thickness-direction gasket behavior with no preload.

You may need to expand the dialog box to see all the columns. For detailed information on how to enter data, see *Entering tabular data*.

8. If you selected **Prony** from the list of **Frequency** options, enter the following data in the **Data** table:

g_i Prony

$\bar{g}_i^P$, the modulus ratio in the first term in the Prony series expansion of the shear relaxation modulus.

k_i Prony

$\bar{k}_i^P$, the modulus ratio in the first term in the Prony series expansion of the bulk relaxation modulus.

tau_i Prony

τ_i, the relaxation time for the first term in the Prony series expansion.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

9. If applicable, use the **Test Data** menu to specify test data from which to define viscoelastic behavior. See the following sections for details:
   - *Specifying shear test data*"
   - *Specifying volumetric test data*"
   - *Specifying combined test data*"

10. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying shear test data

You can use the **Test Data Editor** to specify the normalized shear creep compliance or relaxation modulus as a function of time. For information on using shear test data to define viscoelastic material behavior, see *Time Domain Viscoelasticity*, or *Frequency Domain Viscoelasticity*.

1. Create a viscoelastic material model as described in *Defining time domain viscoelasticity*" or *Defining frequency domain viscoelasticity*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Shear Test Data**.
   A **Test Data Editor** appears.
3. In the **Long-term normalized shear compliance or modulus** field, enter either the long-term, normalized shear compliance, $j_S(\infty)$, for creep test data or the long-term, normalized shear modulus, $g_R(\infty)$, for relaxation test data.
4. In the **Data** table, enter the following data:

   #### js or gR

   The normalized shear compliance, $j_S(t)$, ($(j_S(t) \geq 1)$) for creep test data or the normalized shear relaxation modulus, $g_R(t)$, ($(0 \leq g_R(t) \leq 1)$) for relaxation test data.

   #### Time

   Time, *t*. $(t > 0)$

   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Specifying volumetric test data

You can use the **Test Data Editor** to specify the normalized bulk creep compliance or relaxation modulus as a function of time. For information on using volumetric test data to define viscoelastic material behavior, see *Time Domain Viscoelasticity*, or *Frequency Domain Viscoelasticity*.

1. Create a viscoelastic material model as described in *Defining time domain viscoelasticity*" or *Defining frequency domain viscoelasticity*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Volumetric Test Data**.
   A **Test Data Editor** appears.
3. In the **Long-term normalized volumetric compliance or modulus** field, enter either the long-term, normalized volumetric compliance, $j_K(\infty)$, for creep test data or the long-term, normalized volumetric modulus, $k_R(\infty)$, for relaxation test data.
4. In the **Data** table, enter the following data:

   #### jK or kR

   The normalized volumetric compliance, $j_K(t), \quad (j_K(t) \geq 1)$, for creep test data or the normalized volumetric modulus, $k_R(t), \quad (0 \leq k_R(t) \leq 1)$, for relaxation test data.

   #### Time

   Time, *t*. $(t > 0)$

   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Specifying combined test data

You can use the **Test Data Editor** to specify simultaneously the normalized shear and bulk creep compliances or relaxation moduli as a function of time. For information on using combined test data to define viscoelastic material behavior, see *Time Domain Viscoelasticity*, or *Frequency Domain Viscoelasticity*.

1. Create a viscoelastic material model as described in *Defining time domain viscoelasticity*" or *Defining frequency domain viscoelasticity*."
2. From the **Test Data** menu in the **Edit Material** dialog box, select **Combined Test Data**.
   A **Test Data Editor** appears.
3. In the **Long-term normalized shear compliance or modulus** field, enter either the long-term, normalized shear compliance, $j_S(\infty)$, for creep test data or the long-term, normalized shear modulus, $g_R(\infty)$, for relaxation test data.
4. In the **Long-term normalized volumetric compliance or modulus** field, enter either the long-term, normalized volumetric compliance, $j_K(\infty)$, for creep test data or the long-term, normalized volumetric modulus, $k_R(\infty)$, for relaxation test data.
5. In the **Data** table, enter the following data:

   #### js or gR

   The normalized shear compliance, $j_S(t)$, ($(j_S(t) \geq 1)$) for creep test data or the normalized shear relaxation modulus, $g_R(t)$, ($(0 \leq g_R(t) \leq 1)$) for relaxation test data.

   #### jK or kR

   The normalized volumetric compliance, $j_K(t), \quad (j_K(t) \geq 1)$, for creep test data or the normalized volumetric modulus, $k_R(t), \quad (0 \leq k_R(t) \leq 1)$, for relaxation test data.

   #### Time

   Time, *t*. $(t > 0)$

   For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Defining plasticity

You use the **Edit Material** dialog box to create a plastic material and to specify its material properties.

You can create the following plastic material models:

- **Classical metal plasticity**
- **Cap plasticity**
- **Cast iron plasticity**
- **Clay plasticity**
- **Concrete damaged plasticity**
- **Concrete smeared cracking**
- **Crushable foam plasticity**
- **Drucker-Prager plasticity**
- **Mohr-Coulomb plasticity**
- **Porous metal plasticity**
- **Creep**
- **Swelling**
- **Viscous**

For more information, see *Inelastic Behavior*.

In this section:

- *Defining classical metal plasticity*
- *Defining cap plasticity*
- *Defining cast iron plasticity*
- *Defining clay plasticity*
- *Defining concrete damaged plasticity*
- *Defining concrete smeared cracking*
- *Defining crushable foam plasticity*
- *Defining Drucker-Prager plasticity*
- *Defining Mohr-Coulomb plasticity*
- *Defining porous metal plasticity*
- *Defining a creep law*
- *Defining swelling*
- *Defining the viscous component of a two-layer viscoplasticity model*

Defining classical metal plasticity

The classical metal plasticity model allows you to define the yield and inelastic flow of a metal at relatively low temperatures, where loading is relatively monotonic and creep effects are unimportant.
For more information, see *Classical Metal Plasticity*.

In this section:

- *Specify hardening behavior for classical metal plasticity*
- *Using an isotropic hardening model to define classical metal plasticity*
- *Using a linear kinematic cyclic hardening model to define classical metal plasticity*
- *Using the Johnson-Cook hardening model to define classical metal plasticity*
- *Specifying user subroutine UHARD to define classical metal plasticity*
- *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*
- *Defining the isotropic hardening component of a nonlinear isotropic/kinematic hardening model*
- *Defining rate-dependent yield with yield stress ratios*
- *Defining anisotropic yield and creep*
- *Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations*
- *Specifying cycled yield stress data for the ORNL model*
- *Specifying the annealing temperature of an elastic-plastic material*
- *Defining tensile failure*

Specify hardening behavior for classical metal plasticity

When you use the **Edit Material** dialog box to define the classical plastic behavior of a material, you must select an option for specifying the hardening behavior.

The following hardening options are available:

Option	Hardening behavior
Isotropic	Model hardening where the yield surface changes size uniformly in all directions such that the yield stress increases (or decreases) in all stress directions as plastic straining occurs
Kinematic	Model cyclic loading of a material with a constant rate of hardening
Johnson-Cook	Model isotropic hardening in Abaqus/Explicit, where the yield stress is provided as an analytical function of equivalent plastic strain, strain rate, and temperature
User	Describe yield stress for isotropic hardening through user subroutine *UHARD*
Combined	Model cyclic loading of a material with nonlinear isotropic/kinematic hardening

For general information on selecting a **Hardening** option, see *Hardening*.

- *Using an isotropic hardening model to define classical metal plasticity*
- *Using a linear kinematic cyclic hardening model to define classical metal plasticity*
- *Using the Johnson-Cook hardening model to define classical metal plasticity*
- *Specifying user subroutine UHARD to define classical metal plasticity*
- *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*

Using an isotropic hardening model to define classical metal plasticity

The isotropic hardening model is useful for cases involving gross plastic straining or for cases where the straining at each point is essentially in the same direction in strain space throughout the analysis.

1. From the menu in the **Edit Material** dialog box, select **Mechanical->Plasticity->Plastic**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select **Isotropic**.
3. Select **Use strain-rate-dependent data** if you want enter test data showing yield stress values versus equivalent plastic strain at different equivalent plastic strain rates.

   A **Rate** column appears in the **Data** table.

   Alternatively, if you want to define strain rate dependence using yield stress ratios, you must select **Rate Dependent** from the **Suboptions** menu. See *Defining rate-dependent yield with yield stress ratios*" for details. For background information on rate dependence, see *Rate-Dependent Yield*.
4. Select **Use temperature-dependent data** to define behavior data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
6. Enter the following data in the **Data** table:

Yield Stress

The stress at which yield is initiated.

Plastic Strain

Plastic strain.

Rate

Equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$, for which the stress-strain curve applies.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If desired, use the **Suboptions** menu to enter additional data.

   See the following sections for details:

   - *Defining rate-dependent yield with yield stress ratios*
   - *Specifying the annealing temperature of an elastic-plastic material*
   - *Defining tensile failure*

8. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (For more information, see *Browsing and modifying material behaviors*.)

Using a linear kinematic cyclic hardening model to define classical metal plasticity

Use the linear kinematic model to define a constant rate of cyclic hardening. See *Models for Metals Subjected to Cyclic Loading*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Plastic**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select **Kinematic**.
3. Toggle on **Use temperature-dependent data** to define behavior data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Enter the following data in the **Data** table:

   #### Yield Stress

   The stress at which yield is initiated.

   #### Plastic Strain

   Plastic strain.

   #### Temp

   Temperature.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
5. If desired, use the **Suboptions** menu to enter additional data. See the following sections for details:
   - *Defining anisotropic yield and creep*"
   - *Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations*"
   - *Specifying cycled yield stress data for the ORNL model*"
   - *Specifying the annealing temperature of an elastic-plastic material*"
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Using the Johnson-Cook hardening model to define classical metal plasticity

The Johnson-Cook plasticity model is particularly suited to model high-strain-rate deformation of metals. This model is a particular type of Mises plasticity that includes analytical forms of the hardening law and rate dependence. It is generally used in adiabatic transient dynamic analysis. For more information, see *Johnson-Cook Plasticity*.

1. From the menu in the **Edit Material** dialog box, select **Mechanical->Plasticity->Plastic**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select **Johnson-Cook**.
3. Enter the following data in the **Data** table:

   ### A, B, n, and m

   Material parameters measured at or below the transition temperature.

   ### Melting Temp

   Melting temperature, θ_{melt}, above which the material is melted and behaves like a fluid.

   ### Transition Temp

   Transition temperature, $\theta_{\text{transition}}$. There is no temperature dependence on the expression of the yield stress at or below the transition temperature.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. If desired, use the **Suboptions** menu to enter additional data.

   See the following sections for details:

   - *Defining rate-dependent yield with yield stress ratios*
   - *Specifying the annealing temperature of an elastic-plastic material*
   - *Defining tensile failure*
5. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying user subroutine UHARD to define classical metal plasticity

User subroutine *UHARD* allows you to define the yield surface size and hardening parameters for isotropic plasticity or combined hardening models.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Plastic**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select **User**.
3. In the **Data** table, enter the number of **Hardening Properties** needed as data in user subroutine *UHARD*.
4. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity

The evolution law of this model consists of two components:

- A nonlinear kinematic hardening component, which describes the translation of the yield surface in stress space through the backstress, $\boldsymbol{\alpha}$.
- An isotropic hardening component, which describes the change of the equivalent stress defining the size of the yield surface, σ^0, as a function of plastic deformation.

You can define the kinematic hardening component by selecting **Combined** from the list of **Hardening** options in the **Edit Material** dialog box and entering the required data.

You can define the isotropic hardening component by selecting **Cyclic Hardening** from the **Suboptions** menu and entering data in the **Suboption Editor** that appears.

For more information on cyclic hardening, see *Models for Metals Subjected to Cyclic Loading*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical**->**Plasticity**->**Plastic**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select **Combined**.
3. Click the arrow to the right of the **Data type** field, and specify how you want to define the kinematic hardening component of the model:
   - Select **Half Cycle** to provide stress-strain data obtained from the first half cycle of a unidirectional tension or compression experiment.
   - Select **Parameters** to specify the kinematic hardening parameters C_k and γ_k directly.
   - Select **Stabilized** to provide stress-strain data obtained from the stabilized cycle of a specimen that is subjected to symmetric strain cycles.
4. To specify the number of backstresses to include in the model, click the arrows to the right of the **Number of backstresses** field. The default number of backstresses is 1. The maximum number of backstresses allowed is 10.
   If you selected **Parameters** from the list of **Data type** options, additional columns appear in the table to specify the kinematic hardening parameters for multiple backstresses.
5. Toggle on **Use temperature-dependent data** to define behavior data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
6. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
7. If you selected **Stabilized** from the list of **Data type** options, you can choose to toggle on **Use strain-range-dependent data**. This option is useful if the shapes of the stress-strain curves are significantly different for different strain ranges.
8. In the **Data** table, enter the data relevant to your **Data type** selection (not all of the following parameters will apply):

Yield Stress

The stress at which yield is initiated.

Plastic Strain

Plastic strain.

Yield Stress At Zero Plastic Strain

Yield stress at zero plastic strain, $\sigma|_0$.

Kinematic Hard Parameter C1

Kinematic hardening parameter, C_1.

Gamma 1

Kinematic hardening parameter, γ_1.

Kinematic Hard Parameter C*k* and Gamma *k*

Kinematic hardening parameters C_k and γ_k for multiple backstresses.

Temp

Temperature.

Field *n*

Predefined field variables.

Strain Range

The strain range at which the stress-strain curve is obtained.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

9. To define the isotropic hardening component of the model, select **Cyclic Hardening** from the **Suboptions** menu, and enter the required data in the **Suboption Editor** that appears. See *Defining the isotropic hardening component of a nonlinear isotropic/kinematic hardening model*" for details.
10. If desired, use other options from the **Suboptions** menu to enter additional data. See the following sections for details:
    - *Defining anisotropic yield and creep*"
    - *Specifying the annealing temperature of an elastic-plastic material*"
11. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining the isotropic hardening component of a nonlinear isotropic/kinematic hardening model

The **Suboption Editor** allows you to define the evolution of the elastic domain for the nonlinear isotropic/kinematic hardening model. For more information, see the following sections:

- *Models for Metals Subjected to Cyclic Loading*
- *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*"

1. Create a material model as described in *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Cyclic Hardening**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. To define data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.
5. Specify how you want to define the isotropic hardening component:
   - Toggle on **Use parameters** if you want to enter the material parameters of the exponential law.
   - Toggle off **Use parameters** if you want to define the evolution of the yield surface size as a function of the equivalent plastic strain in tabular form.
6. If you toggled on **Use parameters**, enter the following data in the **Data** table:

Equiv Stress

Equivalent stress defining the size of the elastic range at zero plastic strain.

Q-infinity

Isotropic hardening parameter, Q_∞.

Hardening Param b

Isotropic hardening parameter, b.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If you toggled off **Use parameters**, enter the following data in the **Data** table:

Equiv Stress

Equivalent stress defining the size of the elastic range.

Equiv Plastic Strain

Equivalent plastic strain.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Click **OK** to return to the **Edit Material** dialog box.

Defining rate-dependent yield with yield stress ratios

Abaqus allows you to define a material's yield behavior accurately when the yield strength depends on the rate of straining and the anticipated strain rates are significant. You can define strain rate dependence in two ways:

- Enter hardening curves at different strain rates directly, as described in the following sections:
  - *Using an isotropic hardening model to define classical metal plasticity*"
  - *Defining Drucker-Prager hardening*"
- Define yield stress ratios to specify the rate dependence independently, as described in the procedure below.

For more information on strain rate dependence, see *Rate-Dependent Yield*.

1. Create a material model as described in described in one of the following sections:
   - *Using an isotropic hardening model to define classical metal plasticity*"
   - *Using the Johnson-Cook hardening model to define classical metal plasticity*"
   - *Defining Drucker-Prager hardening*"
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Rate Dependent**.
   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Hardening** field, and select a method for defining hardening dependencies:
   - Select **Power Law** to define yield stress ratios with the Cowper-Symonds overstress law.
   - Select **Tabular** to enter yield stress ratios directly in tabular form as a function of equivalent plastic strain rates.
   - Select **Johnson-Cook** to use an analytical Johnson-Cook form to define *R*.
4. If applicable, toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
5. If applicable, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. If you selected **Power Law** from the list of **Hardening** options, enter the following data in the **Data** table:

Mulitiplier

Material parameter, *D*.

Exponent

Material parameter, *n*.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected **Yield Ratio** from the list of **Hardening** options, enter the following data in the **Data** table:

Yld Stress Ratio

Yield stress ratio, $R = \overline{\sigma}/\sigma^0$.

Eq Plastic Strain Rate

Equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$ (or $\left|\dot{\varepsilon}^{pl}_{axial}\right|$, the absolute value of the axial plastic strain rate in uniaxial compression, for the crushable foam model).

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. If you selected **Johnson-Cook** from the list of **Hardening** options, enter the following data in the **Data** table:

C

Material constant, *C*, which is independent of temperature and field variables.

Epsilon dot zero

Material constant, $\dot{\varepsilon}_0$, which is independent of temperature and field variables.

For detailed information on how to enter data, see *Entering tabular data*.

9. Click **OK** to return to the **Edit Material** dialog box.

Defining anisotropic yield and creep

Abaqus provides an anisotropic yield and creep model for materials that exhibit different yield or creep behavior in different directions. You can define anisotropic yield or creep by specifying stress ratios that are applied in Hill's potential function. For more information, see *Hill Anisotropic Yield/Creep*.

1. Create a material model as described in one of the following sections
   - *Using an isotropic hardening model to define classical metal plasticity*"
   - *Using a linear kinematic cyclic hardening model to define classical metal plasticity*"
   - *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*"
   - *Defining a creep law*"
   - *Defining the viscous component of a two-layer viscoplasticity model*"
   - *Defining clay plasticity*"
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Potential**.
   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

R11, R22, R33, R12, R13, and R23

Yield or creep stress ratios.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations

The ORNL constitutive model in Abaqus/Standard applies to Types 304 and 316 stainless steel, as specified in nuclear standard *NE F9–5T(1981)*. The constitutive theory is uncoupled into a rate-independent plasticity response and a rate-dependent creep response, each of which is governed by a separate constitutive law. For more information, see *ORNL – Oak Ridge National Laboratory Constitutive Model*, and *ORNL constitutive theory*.

1. Create a material model as described in one of the following sections:
   - *Using a linear kinematic cyclic hardening model to define classical metal plasticity*"
   - *Defining a creep law*"
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Ornl**.

   A **Suboption Editor** appears.
3. In the **Saturation rates for kinematic shift** field, enter a value for A, the parameter equal to the saturation rates for kinematic shift caused by creep strain. This parameter is defined by Equation (15) of Section 4.3.3--3 of the Nuclear Standard. The default value is 0.3. Set A=0.0 to use the 1986 revision of the Standard.
4. In the **Rate of kinematic shift wrt creep strain** field, enter a value for H, the parameter equal to the rate of kinematic shift with respect to creep strain. This parameter is defined by Equation (7) of Section 4.3.2–1 of the Nuclear Standard. Set H=0.0 to use the 1986 revision of the Standard. If you omit this parameter, Abaqus determines the value of H according to Section 4.3.3–3 of the 1981 revision of the Standard.
5. If desired, toggle on **Invoke reset procedure** to invoke the optional α reset procedure described in Section 4.3.5 of the Nuclear Standard.
6. Click **OK** to return to the **Edit Material** dialog box.

Specifying cycled yield stress data for the ORNL model

You can use the **Suboption** editor to specify the tenth-cycle yield stress and hardening values for the ORNL constitutive model. This option is relevant only if you also follow the procedure described in *Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations*."

1. Create a material model as described in *Using a linear kinematic cyclic hardening model to define classical metal plasticity*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Cycled Plastic**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. In the **Data** table, enter values for the yield stress, plastic strain, and, if applicable, temperature.

   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Specifying the annealing temperature of an elastic-plastic material

When the temperature of a material point exceeds its annealing temperature, Abaqus assumes that the material point loses its hardening memory. You can specify an annealing temperature and, if desired, define it in terms of field variables. See *Annealing or Melting*, for more information.

1. Create a material model as described in one of the following sections
   - *Using an isotropic hardening model to define classical metal plasticity*"
   - *Using a linear kinematic cyclic hardening model to define classical metal plasticity*"
   - *Using a nonlinear isotropic/kinematic cyclic hardening model to define classical metal plasticity*"
   - *Using the Johnson-Cook hardening model to define classical metal plasticity*"
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Anneal Temperature**.
   A **Suboption Editor** appears.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the anneal temperature depends.
4. Enter the following data in the **Data** table:

   #### Anneal Temperature

   Value of the annealing temperature, θ.

   #### Field *n*

   Predefined field variables.

   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Defining tensile failure

The tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall (the crumbling of a material) or a pressure cutoff.
For more information, see *Tensile Failure Model*.

1. Create a material model as described in one of the following sections:
   - *Using an isotropic hardening model to define classical metal plasticity*
   - *Using the Johnson-Cook hardening model to define classical metal plasticity*
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Tensile Failure**.
   A **Suboption Editor** appears.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the hydrostatic cutoff stress depends.
4. Select **Use temperature-dependent data** to define the data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
5. Specify the failure choice for the failed material points.
   - Select **Allow element deletion when the failure criterion is met** to allow element removal.
   - Specify behavior for the pressure stress and deviatoric stresses.
     1. From the **Pressure** list, select the behavior for the pressure stress.
        - Select **BRITTLE** to require the pressure stress to be compressive when the failure criterion is met.
        - Select **DUCTILE** to limit the pressure stress by the hydrostatic cutoff stress when the failure criterion is met.
     2. From the **Shear** list, select the behavior for the deviatoric stresses.
        - Select **BRITTLE** to set the deviatoric stress to zero when the failure criterion is met.
        - Select **DUCTILE** to specify that the deviatoric stresses are unaffected when the failure criterion is met.
6. Enter the following data in the **Data** table:

   #### Hydrostatic cutoff stress

   Value of the hydrostatic cutoff stress, σ_{cutoff} (positive in tension).

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining cap plasticity

Abaqus allows you to define yield surface parameters for elastic-plastic materials that use the modified Drucker-Prager\Cap plasticity model.
See *Modified Drucker-Prager/Cap Model* for more information.

In this section:

- *Specifying cap plasticity behavior*
- *Defining hardening parameters for a cap plasticity model*
- *Defining creep parameters for a cap plasticity model*

Specifying cap plasticity behavior

You can use the modified Drucker-Prager/Cap plasticity model to simulate geological materials that exhibit pressure-dependent yield. The addition of a cap yield surface helps control volume dilatancy when the material yields in shear and provides an inelastic hardening mechanism to represent plastic compaction. You can also define inelastic time-dependent (creep) behavior coupled with the plastic behavior in an Abaqus/Standard analysis.

For more information on cap plasticity, see *Modified Drucker-Prager/Cap Model*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Cap Plasticity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
4. In the **Data** table, enter the following data:

Material Cohesion

Material cohesion, *d*, in the *p–t* plane (Abaqus/Standard) or in the *p–q* plane (Abaqus/Explicit). (Units of FL^{-2}.)

Angle of Friction

Material angle of friction, β, in the *p–t* plane (Abaqus/Standard) or in the *p–q* plane (Abaqus/Explicit). Enter the value in degrees.

Cap Eccentricity

Cap eccentricity parameter, *R*. Its value must be greater than zero (typically $0.0001 \leq R \leq 1000.0$).

Init Yld Surf Pos

Initial cap yield surface position, $\varepsilon_{vol}^{pl}\big|_0$.

Transition Surf Rad

Transition surface radius parameter, α. Its value should be a small number compared to unity. If you leave this field blank, the default is 0.0 (i.e., no transition surface). If you include creep properties in the material model, you must set α equal to zero.

Flow Stress Ratio

The ratio of the flow stress in triaxial tension to the flow stress in triaxial compression, *K*. The value of *K* should be such that $0.778 \leq K \leq 1.0$. If you leave this field blank or enter a value of 0.0, Abaqus uses a value of 1.0 by default. If you include creep properties in the material model, you should set *K* equal to 1.0. This parameter applies only to Abaqus/Standard analyses.

Temp

Temperature.

Field *n*

Predefined field variables.

5. To define the hardening part of the cap plasticity model, select **Cap Hardening** from the **Suboptions** menu. See *Defining hardening parameters for a cap plasticity model*" for detailed instructions.
6. If you want to specify cap creep behavior, select one of the following options from the **Suboptions** menu:

   - Select **Cap Creep Cohesion** to choose a cohesion creep mechanism that follows the type of plasticity active in the shear-failure plasticity.
   - Select **Cap Creep Consolidation** to choose a consolidation mechanism that follows the type of plasticity active in the cap plasticity region.

   See *Defining creep parameters for a cap plasticity model*" for detailed instructions. For more information on cap creep behavior, see *Creep Formulation*.
7. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining hardening parameters for a cap plasticity model

The hardening curve specified for this model interprets yielding in the hydrostatic pressure sense: the hydrostatic pressure yield stress is defined as a tabular function of the volumetric inelastic strain, and, if desired, a function of temperature and other predefined field variables. The range of values for which you define p_b should be sufficient to include all values of effective pressure stress that the material will be subjected to during the analysis.

See *Modified Drucker-Prager/Cap Model*, for more information.

1. Create a material model as described in *Specifying cap plasticity behavior*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Cap Hardening**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, enter the following data:

Yield Stress

Hydrostatic pressure yield stress. (The initial tabular value must be greater than zero, and values must increase with increasing volumetric inelastic strain.)

Vol Plas Strain

Absolute value of the corresponding volumetric inelastic strain.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining creep parameters for a cap plasticity model

The cap creep model has two possible mechanisms that are active in different loading regions: one is a cohesion mechanism, which follows the type of plasticity active in the shear-failure plasticity region, and the other is a consolidation mechanism, which follows the type of plasticity active in the cap plasticity region.

For more information, see *Creep Formulation*.

1. Create a material model as described in *Specifying cap plasticity behavior*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select either **Cap Creep Cohesion** or **Cap Creep Consolidation**. (See *Creep Formulation* for detailed information on the two creep mechanisms.)

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Law** field, and select the creep law of your choice. See *Specifying Creep Laws* for the data required for each creep law.
4. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
5. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
6. If you selected the **Time**, **SinghM**, or **Time Power** creep law, specify whether to use the total time or the creep time. See *Time-Dependent Behavior* for more information.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining cast iron plasticity

The cast iron plasticity model describes the mechanical behavior of gray cast iron, a material whose microstructure consisting of graphite flakes in a steel matrix. The model definition includes the plastic Poisson's ratio and information on hardening under compression and tension. See *Cast Iron Plasticity*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Cast Iron Plasticity**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Display the **Plasticity** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

      A column labeled **Temp** appears in the **Data** table.

   b. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
   c. Enter the following data in the **Data** table:

Plastic Poisson's Ratio

Value of the plastic "Poisson's ratio," ν_{pl}, where $-1.0 < \nu_{pl} \leq 0.5$. (Dimensionless.) The default value is 0.04.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

3. Display the **Compression Hardening** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

      A column labeled **Temp** appears in the **Data** table.

   b. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
   c. Enter the following data in the **Data** table:

Sigmac

Yield stress in compression, σ_c.

Epsilonc

Absolute value of the corresponding plastic strain. (The first tabular value entered must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

4. Display the **Tension Hardening** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

      A column labeled **Temp** appears in the **Data** table.
   b. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
   c. Enter the following data in the **Data** table:

Sigmat

Yield stress in uniaxial tension, σ_t.

Epsilont

Corresponding plastic strain. (The first tabular value entered must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining clay plasticity

The clay plasticity model allows you to specify the plastic part of the material behavior for elastic-plastic materials that use the extended Cam-clay plasticity model.
See *Critical State (Clay) Plasticity Model* for more details.

In this section:

- *Specifying clay plasticity behavior*
- *Defining compressive clay hardening for a clay plasticity model*
- *Defining tensile clay hardening for a clay plasticity model*
- *Specifying softening regularization for a clay plasticity model*

Specifying clay plasticity behavior

The Abaqus/Standard clay plasticity model describes the inelastic response of cohesionless soils. This model provides a reasonable match to the experimentally observed behavior of saturated clays. You can define the inelastic material behavior by a yield function that depends on the three stress invariants, an associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain.

For more information, see *Critical State (Clay) Plasticity Model*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Clay Plasticity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Hardening** field, and select the hardening law form of your choice:
   - Select **Exponential** to specify an exponential hardening/softening law.
   - Select **Tabular** to specify a piecewise linear hardening/softening relationship.

   See *Hardening Law*, for more information.
3. If you selected the **Exponential** form of the hardening law, you have the option of entering a value for **Intercept**. This parameter corresponds to e_1, the intercept of the virgin consolidation line with the void ratio axis in a plot of void ratio versus the logarithm of pressure stress.
   If you specify a value for **Intercept**, Abaqus ignores any value specified for the initial yield surface size, a_0, in the **Data** table.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. If you selected the **Exponential** form of the hardening law, enter the following data in the **Data** table:

Log Plas Bulk Mod

Logarithmic plastic bulk modulus, λ (dimensionless).

Stress Ratio

Stress ratio at critical state, M.

Init Yld Surf Size

Initial yield surface size, a_0 (units of FL^{-2}). Abaqus ignores this data item if you have specified a value for **Intercept**.

Wet Yld Surf Size

Parameter defining the size of the yield surface on the “wet” side of critical state, β. If this value is omitted or set to zero, a value of 1.0 is assumed.

Flow Stress Ratio

Ratio of the flow stress in triaxial tension to the flow stress in triaxial compression, K. $0.778 \le K \le 1.0$. If this value is left blank or set to zero, a value of 1.0 is assumed.

Temp

Temperature.

Field n

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected the **Tabular** form of the hardening law, enter the following data in the **Data** table:

Stress Ratio

Stress ratio at critical state, M.

Init Vol Plas Strain

Initial volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}\big|_0$ corresponding to $p_c|_0$.

Wet Yld Surf Size

Parameter defining the size of the yield surface on the "wet" side of critical state, β. If this value is omitted or set to zero, a value of 1.0 is assumed.

Flow Stress Ratio

Ratio of the flow stress in triaxial tension to the flow stress in triaxial compression, K. $0.778 \le K \le 1.0$. If this value is left blank or set to zero, a value of 1.0 is assumed.

Temp

Temperature.

Field n

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. If you selected the **Tabular** form of the hardening law, select **Compressive Clay Hardening** from the **Suboptions** menu to provide the hydrostatic compression yield stress as a function of volumetric plastic strain to define piecewise linear hardening/softening of the Cam-clay plasticity yield surface. See *Defining compressive clay hardening for a clay plasticity model*" for detailed instructions.

9. If you selected the **Tabular** form of the hardening law, select **Tensile Clay Hardening** from the **Suboptions** menu to provide the hydrostatic tension yield stress as a function of volumetric plastic strain to define piecewise linear hardening/softening of the Cam-clay plasticity yield surface. See *Defining tensile clay hardening for a clay plasticity model*" for detailed instructions.
10. If desired, select **Potential** from the **Suboptions** menu to specify anisotropic yield behavior. See *Defining anisotropic yield and creep*" for more information.
11. If desired, select **Softening Regularization** from the **Suboptions** menu to specify a regularization scheme to mitigate potential mesh independence of the results. See *Specifying softening regularization for a clay plasticity model*" for detailed instructions.
12. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining compressive clay hardening for a clay plasticity model

The **Suboption Editor** allows you to provide the hydrostatic compression yield stress as a function of volumetric plastic strain to define piecewise linear hardening/softening of the Cam-clay plasticity yield surface. For more information on this form of the hardening law, see *Piecewise Linear Form*.

1. Create a material model as described in *Specifying clay plasticity behavior*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Compressive Clay Hardening**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, enter the following data:

Yield Stress

Value of the hydrostatic pressure stress at yield, p_c. p_c is given as a positive value and must increase with increasing volumetric plastic strain.

Vol Plas Strain

Absolute value of the corresponding compressive volumetric plastic strain. (The first tabular value must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining tensile clay hardening for a clay plasticity model

The **Suboption Editor** allows you to provide the hydrostatic tensile yield stress as a function of volumetric plastic strain to define piecewise linear hardening/softening of the Cam-clay plasticity yield surface. For more information on this form of the hardening law, see *Piecewise Linear Form*.

1. Create a material model as described in *Specifying clay plasticity behavior*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Tensile Clay Hardening**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, enter the following data:

   #### Yield Stress

   Value of the hydrostatic pressure stress at yield, p_t. p_t can be zero or a negative value and must decrease with increasing volumetric plastic strain.

   #### Vol Plas Strain

   Absolute value of the corresponding compressive volumetric plastic strain. (The first tabular value must always be zero.)

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Specifying softening regularization for a clay plasticity model

The **Suboption Editor** allows you to specify a regularization scheme to mitigate potential mesh dependence of results in the Cam-clay plasticity model in the event that it exhibits strain localization with increasing plastic deformation For more information, see *Softening Regularization*.

1. Create a material model as described in *Specifying clay plasticity behavior*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Softening Regularization**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, enter the following data:

Crack Band Length

Crack band length, $l_c^{(m)}$. This value must be greater than zero.

Exponent

Exponent, n_r. This value must be greater than zero.

Bound

Bound on the magnitude of regularization, f_{max}. This value must be greater than zero.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining concrete damaged plasticity

The concrete damaged plasticity model provides a general capability for modeling concrete and other quasi-brittle materials in all types of structures.
This model uses concepts of isotropic damaged elasticity in combination with isotropic tensile and compressive plasticity to represent the inelastic behavior of concrete. See *Concrete Damaged Plasticity*, for more information.

In this section:

- *Defining a concrete damaged plasticity model*
- *Defining concrete compression damage*
- *Defining concrete tension damage*

Defining a concrete damaged plasticity model

The concrete damaged plasticity model is based on the assumption of scalar (isotropic) damage and is designed for applications in which the concrete is subjected to arbitrary loading conditions, including cyclic loading. The model takes into consideration the degradation of the elastic stiffness induced by plastic straining both in tension and compression. It also accounts for stiffness recovery effects under cyclic loading.

For more information, see *Concrete Damaged Plasticity*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Concrete Damaged Plasticity**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the **Plasticity** tab, if necessary, to display the **Plasticity** tabbed page.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

Dilation Angle

Dilation angle, ψ, in the *p*–*q* plane. Enter the value in degrees.

Eccentricity

Flow potential eccentricity, ϵ. The eccentricity is a small positive number that defines the rate at which the hyperbolic flow potential approaches its asymptote. The default is $\epsilon = 0.1$.

fb0/fc0

σ_{b0}/σ_{c0}, the ratio of initial equibiaxial compressive yield stress to initial uniaxial compressive yield stress. The default value is 1.16

K

K_c, the ratio of the second stress invariant on the tensile meridian, $q_{(\mathrm{TM})}$, to that on the compressive meridian, $q_{(\mathrm{CM})}$, at initial yield for any given value of the pressure invariant *p* such that the maximum principal stress is negative, $\widehat{\sigma}_{\max} < 0$. It must satisfy the condition $0.5 < K_c \leq 1.0$. The default value is $2/3$.

Viscosity Parameter

Viscosity parameter, μ, used for the visco-plastic regularization of the concrete constitutive equations in Abaqus/Standard analyses. This parameter is ignored in Abaqus/Explicit. The default value is 0.0. (Units of T.)

Temp

Temperature.

Field *n*

Predefined field variables.

6. Click the **Compressive Behavior** tab to display the **Compressive Behavior** tabbed page. (For information on compressive hardening, see *Defining Compressive Behavior*.)
7. Toggle on **Use strain-rate-dependent data** if the compressive stress data are a function of strain rate.
8. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
9. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
10. Enter the following data in the **Data** table:

Yield Stress

Yield stress in compression, σ_c. (Units of $\mathbf{FL^{-2}}$.)

Inelastic Strain

Inelastic (crushing) strain, $\tilde{\varepsilon}_c^{in}$.

Rate

Inelastic (crushing) strain rate, $\dot{\tilde{\varepsilon}}_c^{in}$. (Units of $\mathbf{T^{-1}}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

11. If desired, select **Compression Damage** from the **Suboptions** menu to specify damage in tabular form. (If you omit damage data, the model behaves as a plasticity model.) See *Defining concrete compression damage*" for details.
12. Click the **Tensile Behavior** tab to display the **Tensile Behavior** tabbed page. (For information on tension stiffening, see *Defining Tension Stiffening*.)
13. Click the arrow to the right of the **Type** field, and select a method for defining the postcracking behavior:
    - Select **Strain** to specify the postcracking behavior by entering the postfailure stress/cracking-strain relationship.
    - Select **Displacement** to define the postcracking behavior by entering the postfailure stress/cracking-displacement relationship.
    - Select **GFI** to define the postcracking behavior by entering the failure stress and the fracture energy.
14. Toggle on **Use strain-rate-dependent data** if the postcracking stress depends on strain rate.
15. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

    A column labeled **Temp** appears in the **Data** table.

16. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
17. In the **Data** table, enter the data relevant to your **Type** choice from Step 13 (not all of the following will apply):

Yield Stress

If you selected **Strain** or **Displacement** from the list of **Type** options, enter the remaining direct stress after cracking, σ_t. (Units of $\mathbf{FL^{-2}}$.)

If you selected **GFI** from the list of **Type** options, enter the failure stress, σ_{t0}. (Units of $\mathbf{FL^{-2}}$.)

Cracking Strain

Direct cracking strain, $\tilde{\varepsilon}_t^{ck}$.

Displacement

Direct cracking displacement, u_t^{ck}. (Units of L.)

Fracture Energy

Fracture energy, G_f. (Units of $\mathbf{FL^{-1}}$.)

Rate

If you selected **Strain** from the list of **Type** options, enter the direct cracking strain rate, $\dot{\tilde{\varepsilon}}_t^{ck}$. (Units of $\mathbf{T^{-1}}$.)

If you selected **Displacement** or **GFI** from the list of **Type** options, enter the direct cracking displacement rate, $\dot{u}_t^{ck}$. (Units of $\mathbf{LT^{-1}}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

18. If desired, select **Tension Damage** from the **Suboptions** menu to specify damage in tabular form. (If you omit damage data, the model behaves as a plasticity model.) See *Defining concrete tension damage*" for details.
19. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining concrete compression damage

You can define the uniaxial compression damage variable, d_c, as a tabular function of inelastic (crushing) strain. For more information, see *Defining Damage and Stiffness Recovery*.

1. Create a material model as described in *Defining a concrete damaged plasticity model*."
2. From the **Suboptions** menu on the **Compressive Behavior** tabbed page, select **Compression Damage**.

   A **Suboption Editor** appears.
3. In the **Tension recovery** field, enter a value for the stiffness recovery factor w_t, which determines the amount of tension stiffness that is recovered as the loading changes from compression to tension.

   If $w_t = 1$, the material fully recovers the tensile stiffness; if $w_t = 0$, there is no stiffness recovery.

   Intermediate values of w_t $(0 \le w_t \le 1)$ result in partial recovery of the tensile stiffness. The default value is 0.0.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. Enter the following data in the **Data** table:

   #### Damage Parameter

   Compressive damage variable, d_c.

   #### Inelastic Strain

   Inelastic (crushing) strain, $\tilde{\varepsilon}_c^{in}$.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining concrete tension damage

You can define the uniaxial tension damage variable, d_t, as a tabular function of either cracking strain or cracking displacement.For more information, see *Defining Damage and Stiffness Recovery*.

1. Create a material model as described in *Defining a concrete damaged plasticity model*."
2. From the **Suboptions** menu in the **Tensile Behavior** tabbed page, select **Tension Damage**.

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Type** field, and select a method for defining the tensile damage variable:
   - Select **Strain** to specify the tensile damage variable as a function of cracking strain.
   - Select **Displacement** to specify the tensile damage variable as a function of cracking displacement.
4. In the **Compression recovery** field, enter a value for the stiffness recovery factor, w_c, which determines the amount of compression stiffness that is recovered as the loading changes from tension to compression.

   If $w_c = 1$, the material fully recovers the compressive stiffness; if $w_c = 0$, there is no stiffness recovery.

   Intermediate values of w_c $(0 \le w_c \le 1)$ result in partial recovery of the compressive stiffness. The default value is **1.0**, which corresponds to the assumption that as cracks close the compressive stiffness is unaffected by tensile damage.
5. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
6. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
7. In the **Data** table, enter the data relevant to your **Type** selection from Step 3 (not all of the following will apply):

Damage Parameter

Tensile damage variable, d_t.

Cracking Strain

Direct cracking strain, $\tilde{\varepsilon}_t^{ck}$.

Displacement

Direct cracking displacement, u_t^{ck}. (Units of L.)

Temp

Temperature.

Field *n*

Predefined field variables.

8. Click **OK** to return to the **Edit Material** dialog box.

Defining concrete smeared cracking

You can use the concrete smeared cracking model to define the properties of plain concrete outside the elastic range in an Abaqus/Standard analysis.
For more information, see *Concrete Smeared Cracking*.

In this section:

- *Specifying a concrete smeared cracking model*
- *Defining tension stiffening for a concrete smeared cracking model*
- *Defining shear retention for a concrete smeared cracking model*
- *Defining the shape of the failure surface for a concrete smeared cracking model*

Specifying a concrete smeared cracking model

The concrete smeared cracking model allows you to define concrete behavior for relatively monotonic loadings under fairly low confining pressures. Abaqus assumes that cracking is the most important aspect of the behavior, and representation of cracking and postcracking behavior dominates the modeling. For more information, see *Concrete Smeared Cracking*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Concrete Smeared Cracking**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
4. Enter the following data in the **Data** table:

   #### Comp Stress

   Absolute value of compressive stress. (Units of FL^{-2}.)

   #### Plastic Strain

   Absolute value of plastic strain. The first stress-strain point given at each value of temperature and field variable must be at zero plastic strain and will define the initial yield point for that temperature and field variable.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
5. Select **Tension Stiffening** from the **Suboptions** menu to model the postfailure behavior for direct straining across cracks. See *Defining tension stiffening for a concrete smeared cracking model*" for details.
6. If desired, select **Shear Retention** from the **Suboptions** menu to define how shear stiffness diminishes as the concrete cracks. See *Defining shear retention for a concrete smeared cracking model*" for details.
7. If desired, select **Failure Ratios** from the **Suboptions** menu to define the shape of the failure surface. See *Defining the shape of the failure surface for a concrete smeared cracking model*" for details.
8. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining tension stiffening for a concrete smeared cracking model

You can model postfailure behavior for direct straining across cracks with tension stiffening, which allows you to define the strain-softening behavior for cracked concrete. This behavior also allows the effects of the reinforcement interaction with concrete to be simulated in a simple manner.

You can specify tension stiffening by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion. See *Tension Stiffening*, for more information.

Tension stiffening information is required for the concrete smeared cracking model.

1. Create a material model as described in *Specifying a concrete smeared cracking model*."
2. From the **Suboptions** menu, select **Tension Stiffening**.

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Type** field, and select a method for defining postcracking behavior:
   - Select **Displacement** to enter the displacement at which a linear loss of strength after cracking gives zero stress.
   - Select **Strain** to enter the postfailure stress-strain relationship directly.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. In the **Data** table, enter the data relevant to your **Type** choice from Step 3 (not all of the following will apply):

   #### Disp

   Displacement, u_0, at which a linear loss of strength after cracking gives zero stress. (Units of L.)

   #### sigma/sigma_c

   Fraction of remaining stress to stress at cracking.

   #### epsilon-epsilon_c

   Absolute value of the direct strain minus the direct strain at cracking.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining shear retention for a concrete smeared cracking model

As concrete cracks, its shear stiffness is diminished. You can define this effect by specifying the reduction in the shear modulus as a function of the opening strain across the crack. You can also specify a reduced shear modulus for closed cracks. See *Cracked Shear Retention*, for more information.

If you do not define shear retention for a concrete smeared cracking model, Abaqus/Standard automatically assumes that the shear response is unaffected by cracking (full shear retention). This assumption is often reasonable: in many cases, the overall response is not strongly dependent on the amount of shear retention.

1. Create a material model as described in *Specifying a concrete smeared cracking model*."
2. From the **Suboptions** menu, select **Shear Retention**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

   #### Rho_close

   The multiplying factor, ϱ^{close}, that defines the modulus for shearing of closed cracks as a fraction of the elastic shear modulus of the uncracked concrete. The default value is 1.0.

   #### Eps_max

   The maximum direct strain across the crack, ε^{max}. The default value is a very large number (full shear retention).

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Defining the shape of the failure surface for a concrete smeared cracking model

You can specify failure ratios to define the shape of the failure surface. If you do not define the shape of the failure surface, Abaqus uses the default values listed below.

1. Create a material model as described in *Specifying a concrete smeared cracking model*."
2. From the **Suboptions** menu, select **Failure Ratios**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

Ratio 1

Ratio of the ultimate biaxial compressive stress to the uniaxial compressive ultimate stress. The default value is 1.16.

Ratio 2

Absolute value of the ratio of uniaxial tensile stress at failure to the uniaxial compressive stress at failure. The default value is 0.09.

Ratio 3

Ratio of the magnitude of a principal component of plastic strain at ultimate stress in biaxial compression to the plastic strain at ultimate stress in uniaxial compression. The default value is 1.28.

Ratio 4

Ratio of the tensile principal stress value at cracking in plane stress, when the other nonzero principal stress component is at the ultimate compressive stress value, to the tensile cracking stress under uniaxial tension. The default value is 1/3.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining crushable foam plasticity

The crushable foam model allows you to define not only crushable foams that are typically used as energy absorption structures, but also other crushable materials other than foams.
See *Crushable Foam Plasticity Models* for more information.

In this section:

- *Specifying a crushable foam model*
- *Defining crushable foam hardening*
- *Defining rate dependence for a crushable foam plasticity model*

Specifying a crushable foam model

You can use the crushable foam model to model the enhanced ability of a foam material to deform in compression due to cell wall buckling processes.
This model is based on the assumption that the resulting deformation is not recoverable instantaneously and can be idealized as plastic for short duration events. See *Crushable Foam Plasticity Models*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Crushable Foam**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Click the arrow to the right of the **Hardening** field, and select a hardening model:

   - Select **Volumetric** to specify a model that assumes that the yield surface is controlled by the volumetric compacting plastic strain experienced by the material.
   - Select **Isotropic** to specify a model that uses a yield surface that is an ellipse centered at the origin in the *p–q* stress plane. The yield surface evolves in a self-similar manner, and the evolution is governed by the equivalent plastic strain.

3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.

4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, enter the data relevant to your **Hardening** choice from Step 2 (not all of the following will apply):

Compression Yield Stress Ratio

Yield stress ratio for compression loading, $k = \sigma_c^0/p_c^0$; $0 < k < 3$. Enter the ratio of initial yield stress in uniaxial compression to initial yield stress in hydrostatic compression.

Hydrostatic Yield Stress Ratio

Yield stress ratio for hydrostatic loading, $k_t = p_t/p_c^0$; $k_t \geq 0$. Enter the ratio of yield stress in hydrostatic tension to initial yield stress in hydrostatic compression, provided as a positive value. The default value is 1.0.

Plastic Poisson's Ratio

Plastic Poisson's ratio, ν_p; $-1 < \nu_p \leq 0.5$.

Temp

Temperature.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Select **Foam Hardening** from the **Suboptions** menu to define hardening data for the crushable foam model. See *Defining crushable foam hardening*" for details.
7. If desired, select **Rate Dependent** from the **Suboptions** menu to specify strain-rate-dependent material behavior. See *Defining rate dependence for a crushable foam plasticity model*" for details.
8. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining crushable foam hardening

You must provide crushable foam hardening data to complete a crushable foam plasticity definition. See *Crushable Foam Plasticity Models*, for more information.

1. Create a material model as described in *Specifying a crushable foam model*."
2. From the **Suboptions** menu, select **Crushable Foam Hardening**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

Yield Stress

Yield stress in uniaxial compression, σ_c, provided as a positive value.

Vol Plastic Strain

Absolute value of the corresponding plastic strain. (The first tabular value entered must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining rate dependence for a crushable foam plasticity model

As strain rates increase, many materials show an increase in the yield stress. For many crushable foam materials this increase in yield stress becomes important when the strain rates are in the range of 0.1–1 per second and can be very important if the strain rates are in the range of 10–100 per second, as commonly occurs in high-energy dynamic events.

For more information on strain rate dependence, see *Crushable Foam Plasticity Models*, and *Rate-Dependent Yield*.

1. Create a material model as described in *Specifying a crushable foam model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Rate Dependent**.

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Hardening** field, and select a method for defining hardening dependencies:

   - Select **Power Law** to define yield stress ratios with the Cowper-Symonds overstress law.
   - Select **Tabular** to enter yield stress ratios directly in tabular form as a function of equivalent plastic strain rates.
4. If applicable, toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. If applicable, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. If you selected **Power Law** from the list of **Hardening** options, enter the following data in the **Data** table:

Mulitiplier

Material parameter, *D*.

Exponent

Material parameter, *n*.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected **Yield Ratio** from the list of **Hardening** options, enter the following data in the **Data** table:

Yld Stress Ratio

Yield stress ratio, $R = \overline{\sigma}/\sigma^0$.

Eq Plastic Strain Rate

Equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$ (or $\left|\dot{\varepsilon}^{pl}_{axial}\right|$, the absolute value of the axial plastic strain rate in uniaxial compression, for the crushable foam model).

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Click **OK** to return to the **Edit Material** dialog box.

Defining Drucker-Prager plasticity

You can define a Drucker-Prager model to model frictional materials, which are typically granular-like soils and rock and exhibit pressure-dependent yield.
See *Extended Drucker-Prager Models* for more information.

In this section:

- *Defining a Drucker-Prager model*
- *Defining Drucker-Prager hardening*
- *Defining Drucker-Prager creep*
- *Specifying triaxial test data for a Drucker-Prager material model*

Defining a Drucker-Prager model

The extended Drucker-Prager family of plasticity models describes the behavior of granular materials or polymers in which the yield behavior depends on the equivalent pressure stress. The inelastic deformation may sometimes be associated with frictional mechanisms such as sliding of particles across each other. See *Extended Drucker-Prager Models*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Drucker Prager**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Shear criterion** field, and specify which yield criterion you want to define. See the following sections for more information:
   - *Linear Drucker-Prager Model*
   - *Hyperbolic and General Exponent Models*
3. If you are performing an Abaqus/Standard analysis, enter a value for the **Flow potential eccentricity**, ϵ.

   The eccentricity is a small positive number that defines the rate at which the hyperbolic flow potential approaches its asymptote. The default is $\epsilon = 0.1$ for the exponent model, and if $\psi = \beta$, it is set to $\epsilon = (d'|_0 - p_t|_0 \tan\beta) / (\overline{\sigma}|_0 \tan\beta)$ for the hyperbolic model to ensure associated flow. See *Extended Drucker-Prager Models*, for more information.
4. If you selected **Exponent Form** from the list of **Shear criterion** options, you can toggle on **Use Suboption Triaxial Test Data** to request that Abaqus compute the material constants from triaxial test data at different levels of confining pressure. See *General Exponent Model*, for more information.
5. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
6. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
7. In the **Data** table, enter the data relevant to your **Shear criterion** choice (not all of the following will apply):

Angle of Friction

If you selected **Linear** from the list of **Shear criterion** options, enter the material angle of friction, β, in the *p–t* plane.

If you selected **Hyperbolic** from the list of **Shear criterion** options, enter the material angle of friction, β, at high confining pressure in the *p–q* plane.

Enter the value in degrees.

Flow Stress Ratio

The ratio of the flow stress in triaxial tension to the flow stress in triaxial compression, *K*.

$0.778 \le K \le 1.0$. If you leave this field blank or enter a value of 0.0 is entered, Abaqus uses a default of 1.0. If you plan to define creep behavior, set *K* to 1.0.

Dilation Angle

If you selected **Linear** from the list of **Shear criterion** options, enter the dilation angle, ψ, in the p–t plane.

If you selected **Hyperbolic** or **Exponent Form** from the list of **Shear criterion** options, enter the dilation angle, ψ, at high confining pressure in the p–q plane.

Enter the value in degrees.

Init Tension

Initial hydrostatic tension strength, $p_t|_0$. (Units of FL^{-2}.)

a

Material constant a.

b

Exponent b. To ensure a convex yield surface, $b \geq 1$.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Select **Drucker Prager Hardening** from the **Suboptions** menu to specify hardening data for the Drucker-Prager model. See *Defining Drucker-Prager hardening*" for details.
9. If desired, select **Drucker Prager Creep** from the **Suboptions** menu to specify creep data for the Drucker-Prager model. This option is valid only if you have selected **Linear** from the list of **Shear criterion** options and are performing an Abaqus/Standard analysis. See *Defining Drucker-Prager creep*" for details.
10. If you toggled on **Use Suboption Triaxial Test Data** in Step 4, select **Triaxial Test Data** from the **Suboptions** menu to enter the triaxial test data. See *Specifying triaxial test data for a Drucker-Prager material model*" for details.
11. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining Drucker-Prager hardening

Use the **Suboption Editor** to specify hardening data for a Drucker-Prager model. For more information on Drucker-Prager hardening, see *Hardening and Rate Dependence*.

1. Create a material model as described in *Defining a Drucker-Prager model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Drucker Prager Hardening**.

   A **Suboption Editor** appears.
3. Select the **Hardening behavior type** of your choice.

   The evolution of the yield surface with plastic deformation is described in terms of the equivalent stress $\bar{\sigma}$, which you can choose to be either the uniaxial **Compression** yield stress, the uniaxial **Tension** yield stress, or the **Shear** (cohesion) yield stress
4. Toggle on **Use strain-rate-dependent data** if you want enter data showing yield stress values versus equivalent plastic strain at different equivalent plastic strain rates.

   A **Rate** column appears in the **Data** table.

   Alternatively, if you want to define strain rate dependence using yield stress ratios, you must select **Rate Dependent** from the **Suboptions** menu in the **Edit Material** dialog box. See *Defining rate-dependent yield with yield stress ratios*" for details. For background information on rate dependence, see *Rate-Dependent Yield*.
5. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
6. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
7. Enter the following data in the **Data** table:

   #### Yield Stress

   Yield stress.

   #### Abs Plastic Strain

   Absolute value of the corresponding plastic strain. (The first tabular value entered must always be zero.)

   #### Rate

   Equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$, for which this hardening curve applies.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
8. Click **OK** to return to the **Edit Material** dialog box.

Defining Drucker-Prager creep

You can define classical "creep" behavior of materials that exhibit plasticity according to the extended Drucker-Prager models in Abaqus/Standard analyses. The creep behavior in such materials is intimately tied to the plasticity behavior (through the definitions of creep flow potentials and definitions of test data), so you must include both Drucker-Prager plasticity and Drucker-Prager hardening data in the material definition.

The creep data that you enter must be consistent with the **Hardening behavior type** that you select when defining Drucker-Prager hardening (see *Defining Drucker-Prager hardening*" for details).

For more information, see *Creep Models for the Linear Drucker-Prager Model*.

1. Create a material model as described in *Defining a Drucker-Prager model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Drucker Prager Creep**.

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Law** field, and select the creep law option of your choice:
   - Select **Strain** to choose a strain hardening power law.
   - Select **Time** to choose a time hardening power law
   - Select **SinghM** to choose a Singh-Mitchell type law.
   - Select **User** to specify the creep law with user subroutine *CREEP*.

   For more information, see *Specifying a Creep Law*.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. If you selected either the **Strain** or the **Time** creep law option, enter the following data in the **Data** table:

   ## A, n, and m

   Creep material parameters.

   ## Temp

   Temperature.

   ## Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. If you selected the **SinghM** creep law option, enter the following data in the **Data** table:

   ## A, alpha, m, and t1

   Creep material parameters A, α, t_1, and m.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Click **OK** to return to the **Edit Material** dialog box.

Specifying triaxial test data for a Drucker-Prager material model

Abaqus can use triaxial test data to calibrate the material parameters that define the **Exponent Form** of Drucker-Prager plasticity. See *General Exponent Model*, for more information.

1. Create a material model as described in *Defining a Drucker-Prager model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Triaxial Test Data**.

   A **Suboption Editor** appears.
3. Enter values for **Material constant a**, **Material constant b**, and **Material constant pt** if they are known and held fixed at the input value. Alternatively, you can leave one or more of the fields blank if you want Abaqus to calibrate the values from the triaxial test data.
4. Enter the following data in the **Data** table:

   ## Confining Stress

   Sign and magnitude of the confining stress, $\sigma_1 = \sigma_2$.

   ## Loading Dirn Stress

   Sign and magnitude of the stress in the loading direction, σ_3.

   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to return to the **Edit Material** dialog box.

Defining Mohr-Coulomb plasticity

You can use the Mohr-Coulomb plasticity model for geotechnical engineering design applications. The model uses the classical Mohr-Coloumb yield criterion: a straight line in the meridional plane and an irregular hexagonal section in the deviatoric plane. However, the Abaqus Mohr-Coulomb model has a completely smooth flow potential instead of the classical hexagonal pyramid: the flow potential is a hyperbola in the meridional plane, and it uses the smooth deviatoric section proposed by Menétrey and Willam. See *Mohr-Coulomb Plasticity*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Mohr Coulomb Plasticity**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the **Plasticity** tab, if necessary, to display the **Plasticity** tabbed page.
3. Choose how you want to define **Deviatoric eccentricity**, *e*:
   - Select **Calulated default** to allow Abaqus to calculate the deviatoric eccentricity as $e = (3 - \sin\phi) / (3 + \sin\phi)$, where ϕ is the Mohr-Coulomb **Friction Angle** that you specify in the **Data** table.
   - Select **Specify**, and enter a value for deviatoric eccentricity in the field provided. The range of values *e* can have is $1/2 < e \leq 1$.
4. Enter a value for **Meridional eccentricity**, ϵ.

   The meridional eccentricity is a small positive number that defines the rate at which the flow potential approaches its asymptote.
5. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
6. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
7. Enter the following data in the **Data** table:

   #### Friction Angle

   Friction angle, ϕ, at high confining pressure in the p–$R_{mc}q$ plane. Enter the value in degrees.

   #### Dilation Angle

   Dilation angle, ψ, at high confining pressure in the p–$R_{mw}q$ plane. Enter the value in degrees.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
8. Click the **Cohesion** tab to display the **Cohesion** tabbed page.
9. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

A column labeled **Temp** appears in the **Data** table.

10. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
11. Enter the following data in the **Data** table:

Cohesion Yield Stress

Cohesion yield stress.

Abs Plastic Strain

Absolute value of the corresponding plastic strain. (The first tabular value entered must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

12. If desired, toggle on **Specify tension cutoff** and click the **Tension Cutoff** tab to specify tension cutoff stress data to limit the load-carrying capacity of the material model near the tensile region.
13. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

    A column labeled **Temp** appears in the **Data** table.
14. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
15. Enter the following data in the **Data** table:

Tension Cutoff Stress

Yield stress in uniaxial tension, σ_t.

Tensile Plastic Strain

Corresponding plastic strain. (The first tabular value entered must always be zero.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

16. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining porous metal plasticity

You can use the porous metal plasticity model for materials with a dilute concentration of voids in which the relative density is greater than 0.9.
See *Porous Metal Plasticity* for more information.

In this section:

- *Defining a porous metal plasticity model*
- *Defining porous material failure criteria*
- *Defining void nucleation in a porous material*

Defining a porous metal plasticity model

The porous metal plasticity model describes materials that exhibit damage in the form of void initiation and growth. You can also use this model for some powder metal process simulations at high relative densities (relative density is defined as the ratio of the volume of solid material to the total volume of the material). The model is based on Gurson's porous metal plasticity theory with void nucleation and is intended for use with materials that have a relative density that is greater than 0.9. The model is adequate for relatively monotonic loading.

See *Porous Metal Plasticity* for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Porous Metal Plasticity**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter a value for the initial **Relative density** of the material.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

q1, q2, and q3

Material parameters q_1, q_2, q_3.

For typical metals the ranges of the parameters reported in the literature are q_1 = 1.0 to 1.5, q_2 = 1.0, and $q_3 = q_1^2$ = 1.0 to 2.25 (see *Necking of a round tensile bar*). The original Gurson model is recovered when $q_1 = q_2 = q_3$ = 1.0.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. If desired, select **Porous Failure Criteria** from the **Suboption** menu to specify material failure criteria for an Abaqus/Explicit analysis. See *Defining porous material failure criteria* for details.
7. If desired, select **Void Nucleation** from the **Suboption** menu to define the nucleation of voids in the porous material. See *Defining void nucleation in a porous material* for details.
8. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Defining porous material failure criteria

You use the **Suboption Editor** to define failure in a porous metal plasticity model. See *Failure Criteria in Abaqus/Explicit* for more information.

1. Create a material model as described in *Defining a porous metal plasticity model*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Porous Failure Criteria**.
   A **Suboption Editor** appears.
3. Enter a value for the **Total volume void fraction at total failure**, $f_F > 0$. The default is 1.
4. Enter a value for the **Critical void volume fraction** (threshold of rapid loss of stress carrying capacity), $f_c \geq 0$. The default is f_F.
5. Click **OK** to return to the **Edit Material** dialog box.

Defining void nucleation in a porous material

You use the **Suboption Editor** to define the nucleation of voids in a porous metal plasticity model. See *Void Growth and Nucleation* for more information.

1. Create a material model as described in *Defining a porous metal plasticity model*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Void Nucleation**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

Mean

Mean value of the nucleation-strain normal distribution, ε_N.

Standard Deviation

Standard deviation of the nucleation-strain normal distribution, s_N.

Volume Fraction

Volume fraction of nucleating voids, f_N.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining a creep law

In an Abaqus/Standard analysis, you can define classical deviatoric metal creep behavior in a user subroutine or by providing parameters for the creep laws.
You can define creep behavior in user subroutine *CREEP*.

See *Rate-Dependent Plasticity: Creep and Swelling* for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Creep**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Law** field, and select the creep law of your choice. See *Creep Behavior* for the data required for each creep law.
3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. If desired, select **Ornl** from the **Suboptions** menu to implement the creep rules specified by the Oak Ridge National Laboratory constitutive model. See *Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations* for more information.
6. If desired, select **Potential** from the **Suboptions** menu to specify anisotropic creep behavior. See *Defining anisotropic yield and creep* for more information.
7. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Defining swelling

You can enter swelling data in tabular form.
In addition, user subroutine *CREEP* (*CREEP*) provides a very general capability for implementing viscoplastic models such as creep and swelling models.

See *Volumetric Swelling Behavior* for more information.

In this section:

- *Defining a volumetric swelling model*
- *Defining anisotropic swelling*

Defining a volumetric swelling model

As with the creep laws, volumetric swelling laws are usually complex and are most conveniently specified in user subroutine *CREEP*. However, you can also enter tabular swelling data in the **Edit Material** dialog box. See *Volumetric Swelling Behavior*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Swelling**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Law** field, and select an option for specifying swelling data:
   - Select **Input** to enter tabular data in the **Edit Material** dialog box.
   - Select **User-defined** to define the swelling behavior in user subroutine *CREEP*
3. If you selected **Input** in Step 2, toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
4. If you selected **Input** in Step 2, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. If you selected **Input** in Step 2, enter the following data in the **Data** table:

Strain Rate

Volumetric swelling strain rate.

Temp

Temperature.

Field *n*

Predefined field variables.

6. If desired, select **Ratios** from the **Suboptions** menu to define anisotropic swelling. See *Defining anisotropic swelling*" for details.
7. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining anisotropic swelling

You can specify ratios in the **Suboptions Editor** to define the swelling rate in each material direction. See *Volumetric Swelling Behavior*, for more information.

1. Create a material model as described in *Defining a volumetric swelling model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Ratios**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

r11, r22, and r33

Anisotropic swelling ratios, r_{11}., r_{22}., and r_{33}..

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining the viscous component of a two-layer viscoplasticity model

The two-layer viscoplasticity model in Abaqus/Standard is useful for modeling materials in which significant time-dependent behavior as well as plasticity is observed. For metals this typically occurs at elevated temperatures.

This model consists of three parts: elastic, plastic, and viscous. You can define the viscous behavior of the material by selecting a creep law and entering viscosity parameters. See *Two-Layer Viscoplasticity* for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity->Viscous**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Law** field, and select the creep law of your choice. See *Viscous Behavior* and *Creep Behavior* for the data required for each creep law.
3. To define behavior data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. If desired, select **Potential** from the **Suboptions** menu to define anisotropic viscosity. See *Defining anisotropic yield and creep* for details.
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Defining damage

You use the **Edit Material** dialog box to specify material damage initiation criteria and associated damage evolution.
Once an initiation criterion is met, Abaqus applies the associated damage evolution law to determine the material degradation. You can specify the following types of damage in Abaqus/CAE:

- **Ductile**
- **FLD**
- **FLSD**
- **Johnson-Cook**
- **Maxe** or **Quade**
- **Maxs** or **Quads**
- **Maxps** or **Maxpe**
- **M-K**
- **MSFLD**
- **Shear**
- **Hosford-Coulomb**
- **Hashin**
- **LaRC05**
- **Mullins Effect**

You can define multiple damage initiation criteria and damage evolution models to accurately represent the behavior of a material. When a damage initiation criterion is met, material damage has begun. Abaqus uses the damage evolution definition associated with the initiation criterion to evaluate the extent of the damage. If you do not define damage evolution, Abaqus continues to evaluate the damage initiation criterion to provide an indication of the extent to which the analysis has exceeded the initiation point.

In addition to damage initiation and evolution, Abaqus/Standard uses a viscous regularization scheme to improve convergence as fiber-reinforced composite materials (Hashin damage model) are damaged.

For more information on material damage, see *Progressive Damage and Failure*.

In this section:

- *Ductile damage*
- *Forming limit diagram (FLD) damage*
- *Forming limit stress diagram (FLSD) damage*
- *Johnson-Cook damage*
- *Maximum or quadratic nominal strain damage*
- *Maximum or quadratic nominal stress damage*
- *Maximum principal stress or strain damage*
- *Marciniak-Kuczynski (M-K) damage*
- *Müschenborn-Sonne forming limit diagram (MSFLD) damage*
- *Shear damage*
- *Hosford-Coulomb damage*
- *Hashin damage*

- *LaRC05 damage*
- *Mullins effect*
- *Damage evolution*
- *Damage stabilization*

Ductile damage

The **Ductile** damage initiation criterion is a model for predicting the onset of damage due to nucleation, growth, and coalescence of voids in ductile metals. The model assumes that the equivalent plastic strain at the onset of damage is a function of stress triaxiality and strain rate. The ductile criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->Ductile Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the material parameter, **Accumulation power**, equal to the value of the power coefficient, *N*.
3. To define material damage data that depend on the lode angle, toggle on **Use lode-dependent data**.

   A column labeled **lode-dependent** appears in the **Data** table.
4. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
5. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
6. Enter damage parameters in the **Data** table:

Fracture Strain

Equivalent fracture strain at damage initiation.

Stress Triaxiality

The stress triaxiality is defined as $\eta = -p/q$, where p is the pressure stress and q is the Mises equivalent stress.

Strain Rate

The equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

8. Click **OK** to exit the material editor.

• *Damage Initiation for Ductile Metals*

Forming limit diagram (FLD) damage

A forming limit diagram (FLD) is a plot of the forming limit strains in the space of principal (in-plane) logarithmic strains. The **FLD** damage initiation criterion is intended to predict the onset of necking instability in sheet metal forming. The maximum strains that a sheet material can sustain prior to the onset of necking are referred to as the forming limit strains.

Damage due to bending deformation cannot be evaluated using this model. For more information, see *Damage Initiation for Ductile Metals*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->FLD Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

3. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

4. Enter damage parameters in the **Data** table:

Major Principal Strain

The maximum value of the in-plane principal limit strain.

Minor Principal Strain

The minimum value of the in-plane principal limit strain.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

6. Click **OK** to exit the material editor.

Forming limit stress diagram (FLSD) damage

The **FLSD** damage initiation criterion is intended to predict the onset of necking instability in sheet metal forming. The strain-based forming limit curves (as used in the *FLD criterion*) are converted to stress-based curves to reduce the dependence on the strain path. This improves the performance of the FLSD damage model under conditions of arbitrary loading.

Similar to the FLD criterion, damage due to bending deformation cannot be evaluated using this model. For more information, see *Damage Initiation for Ductile Metals*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->FLSD Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

3. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

4. Enter damage parameters in the **Data** table:

Major Principal Stress

The maximum value of the in-plane principal limit stress.

Minor Principal Stress

The minimum value of the in-plane principal limit stress.

Temp

Temperature, θ.

Field n

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

6. Click **OK** to exit the material editor.

Johnson-Cook damage

The **Johnson-Cook** damage initiation criterion is a special case of the ductile damage criterion model for predicting the onset of damage due to nucleation, growth, and coalescence of voids in ductile metals.
The model assumes that the equivalent plastic strain at the onset of damage is a function of stress triaxiality and strain rate. The Johnson-Cook criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->Johnson-Cook Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the material parameter, **Accumulation power**, equal to the value of the power coefficient, *N*.
3. Enter the damage parameters in the **Data** table:

   d_1–d_5

   Failure parameters.

   #### Melting temperature

   θ_{melt}.

   #### Transition temperature

   $\theta_{\text{transition}}$.

   #### Reference strain rate

   $\dot{\varepsilon}_0$

   You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
4. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*.
5. Click **OK** to exit the material editor.

- *Ductile damage*
- *Damage Initiation for Ductile Metals*

Maximum or quadratic nominal strain damage

The **Maxe** and **Quade** damage initiation criteria are used to predict damage initiation in cohesive elements where the cohesive layers are defined in terms of traction-separation.
Both forms evaluate the strain ratios between a given strain value and the peak nominal strain value in each of three directions. ε_n^o, ε_s^o, and ε_t^o represent the peak values of the nominal strain when the deformation is either purely normal to the interface or purely in the first or the second shear direction, respectively. The **Maxe** criterion is based on the maximum value of the three ratios, whereas the **Quade** criterion is based on a quadratic combination of all three ratios.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Traction Separation Laws->Maxe Damage** or **Quade Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. If you are using the extended finite element method (XFEM) to model fracture, you can specify the crack propagation direction when the damage initiation criterion is satisfied. The crack can extend at a direction **Normal** to the local 1-direction (default) or **Parallel** to the local 1-direction.
3. Enter the **Tolerance**. The value should be equal to the tolerance within which the damage initiation criterion must be satisfied.
4. To define the separate tolerance used for crack growth, toggle on **Growth Tolerance** and enter the **Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **default**.
5. To define the separate tolerance used for unstable crack growth, toggle on **Unstable Growth Tolerance** and enter the **Unstable Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **infinite**.
6. Select the arrow to the right of the **Position** field, and select the method for computing the stress/strain fields ahead of the crack tip to determine if the damage initiation criterion is satisfied and to determine the crack propagation direction (if needed):

   - Select **Centroid** to use the stress/strain at the element centroid.
   - Select **Crack tip** to use the stress/strain extrapolated to the crack tip.
   - Select **Combined** to use the stress/strain extrapolated to the crack tip to determine if the damage initiation criterion is satisfied and to use the stress/strain at the element centroid to determine the crack propagation direction (if needed).

7. To define the material damage data that depend on the effective rate of separation, toggle on **Use rate-dependent data**.

   A column labeled **rate-dependent** appears in the **Data** table.
8. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
9. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
10. Enter damage parameters in the **Data** table:

Nominal Strain Normal-only Mode

Nominal strain at damage initiation in a normal-only mode.

Nominal Strain Shear-only mode First Direction

Nominal strain at damage initiation in a shear-only mode that involves separation only along the first shear direction.

Nominal Strain Shear-only mode Second Direction

Nominal strain at damage initiation in a shear-only mode that involves separation only along the second shear direction.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

11. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

    For more information, see *Damage evolution*."

12. Select **Suboptions->Damage Stabilization Cohesive** to enter viscous coefficients and improve the model convergence.

    For more information, see *Damage stabilization*."

13. Click **OK** to exit the material editor.

• *Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*

Maximum or quadratic nominal stress damage

The **Maxs** and **Quads** damage initiation criteria are used to predict damage initiation in cohesive elements where the cohesive layers are defined in terms of traction-separation. Both forms evaluate the stress ratios between a given stress value and the peak nominal stress value in each of three directions. t_n^o, t_s^o, and t_t^o represent the peak values of the nominal stress when the deformation is either purely normal to the interface or purely in the first or the second shear direction, respectively. The **Maxs** criterion is based on the maximum value of the three ratios, whereas the **Quads** criterion is based on a quadratic combination of all three ratios.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Traction Separation Laws->Maxs Damage** or **Quads Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. If you are using the extended finite element method (XFEM) to model fracture, you can choose the crack propagation direction when the damage initiation criterion is satisfied. The crack can extend at a direction normal to the element local 1-direction (default) or parallel to the element local 1-direction.
3. Enter the **Tolerance**. The value should be equal to the tolerance within which the damage initiation criterion must be satisfied.
4. To define the separate tolerance used for crack growth, toggle on **Growth Tolerance** and enter the **Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **default**.

5. To define the separate tolerance used for unstable crack growth, toggle on **Unstable Growth Tolerance** and enter the **Unstable Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **infinite**.

6. Select the arrow to the right of the **Position** field, and select the method for computing the stress/strain fields ahead of the crack tip to determine if the damage initiation criterion is satisfied and to determine the crack propagation direction (if needed):

   - Select **Centroid** to use the stress/strain at the element centroid.
   - Select **Crack tip** to use the stress/strain extrapolated to the crack tip.
   - Select **Combined** to use the stress/strain extrapolated to the crack tip to determine if the damage initiation criterion is satisfied and to use the stress/strain at the element centroid to determine the crack propagation direction (if needed).

7. To define the material damage data that depend on effective rate of separation, toggle on **Use rate-dependent data**.

   A column labeled **rate-dependent** appears in the **Data** table.

8. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

9. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

10. Enter damage parameters in the **Data** table:

Maximum Nominal Stress Normal-only Mode

Nominal stress at damage initiation in a normal-only mode.

Maximum Nominal Stress Shear-only mode First Direction

Nominal stress at damage initiation in a shear-only mode that involves separation only along the first shear direction.

Maximum Nominal Stress Shear-only mode Second Direction

Nominal stress at damage initiation in a shear-only mode that involves separation only along the second shear direction.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

11. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

    For more information, see *Damage evolution*."

12. Select **Suboptions->Damage Stabilization Cohesive** to enter viscous coefficients and improve the model convergence.

    For more information, see *Damage stabilization*."

13. Click **OK** to exit the material editor.

• *Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*

Maximum principal stress or strain damage

The **Maxps** and **Maxpe** damage initiation criteria are used to predict damage initiation in the XFEM enriched region.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Traction Separation Laws->Maxps Damage** or **Maxpe Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Enter the **Tolerance**. The value should be equal to the tolerance within which the damage initiation criterion must be satisfied.
3. To define the separate tolerance used for crack growth, toggle on **Growth Tolerance** and enter the **Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **default**.

4. To define the separate tolerance used for unstable crack growth, toggle on **Unstable Growth Tolerance** and enter the **Unstable Growth Tolerance**.

   If this parameter is included but no value is specified, keep the toggle on for **infinite**.

5. Select the arrow to the right of the **Position** field, and select the method for computing the stress/strain fields ahead of the crack tip to determine if the damage initiation criterion is satisfied and to determine the crack propagation direction (if needed):

   - Select **Centroid** to use the stress/strain at the element centroid.
   - Select **Crack tip** to use the stress/strain extrapolated to the crack tip.
   - Select **Combined** to use the stress/strain extrapolated to the crack tip to determine if the damage initiation criterion is satisfied and to use the stress/strain at the element centroid to determine the crack propagation direction (if needed).
   - Select **Nonlocal** to use the stress/strain extrapolated to the crack tip to determine if the damage initiation criterion is satisfied and to use the stress/strain averaged over a group of elements around the crack tip in the enriched region to determine the crack propagation direction (if needed).

6. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

7. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

8. Enter the **Angsmooth**. The value should be equal to the maximum allowed difference (in degrees) below which the normals of the crack facets are included in the moving least-squares approximation to smooth out the crack normals to obtain the crack propagation direction along the crack front, and it can be used only in conjunction with **Position=Nonlocal** in enriched elements.
9. Enter the **R Crack Direction**. The value should be equal to the radius around the crack tip within which the elements are included for calculating the averaged stress/strain and for smoothing out the normals of the individual crack facets along the crack front used to obtain the crack propagation direction, and it can be used only in conjunction with **Position=Nonlocal**.
10. Select the arrow to the right of the **IniSmooth** field, and it can be used only in conjunction with **Position=Nonlocal** in enriched elements:

    - Select **Yes** to indicate the normals of pre-existing crack facets in elements are included in the moving least-squares approximation to smooth out the crack normals to obtain the crack propagation direction along the crack front.

- Select **No** to Indicate the normals of pre-existing crack facets in elements are excluded from the moving least-squares approximation to smooth out the crack normals to obtain the crack propagation direction along the crack front..

11. Select the arrow to the right of the **Smoothing** field, and it can be used only in conjunction with **Position=Nonlocal**:

- Select **None** to use the stress/strain at the integration points directly for averaging.
- Select **Nodal** to assemble the stress/strain to element nodes and then interpolate to the integration points for averaging.

12. Select the arrow to the right of the **Npoly** field, and is used to specify the number of terms in the polynomial used for the moving least-squares approximation to smooth out the crack normals, only in conjunction with **Position=Nonlocal** in enriched elements:

- Select **Least Square** to set NPOLY=0 and to suppress the moving least-squares approximation.
- Select **Linear** to set NPOLY=4 and to use a linear polynomial approximation.
- Select **Quadratic** to set NPOLY=7 and to use a quadratic polynomial approximation.
- Select **Cubic** to set NPOLY=10 and to use a cubic polynomial approximation.

13. Select the arrow to the right of the **Weighting Method** field, and it can be used only in conjunction with **Position=Nonlocal**:

- Select **Uniform** to average the stress/strain with a uniform weight function.
- Select **Gauss** to average the stress/strain with a Gaussian weight function.
- Select **Cubic Spline** to average the stress/strain with a cubic spline weight function.
- Select **User** to average the stress/strain with a user-defined weight function.

14. Enter damage parameters in the **Data** table:

Maximum Principal Stress or Maximum Principal Strain

Maximum principal stress or strain at damage initiation.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

15. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

For more information, see *Damage evolution*."

16. Select **Suboptions->Damage Stabilization Cohesive** to enter the viscous coefficient and improve the model convergence.

For more information, see *Damage stabilization*."

17. Click **OK** to exit the material editor.

• *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*

Marciniak-Kuczynski (M-K) damage

The **M-K** damage initiation criterion is used to predict sheet metal forming limits for arbitrary loading paths. The model introduces thickness imperfections, in the form of grooves, in the sheet material to simulate defects. Damage occurs when the ratio of deformation in the grooves relative to deformation in the original sheet thickness exceeds a critical value. By default, Abaqus evaluates four grooves at equally spaced angles of 0°, 45°, 90°, and 135° with respect to the local 1-direction of the material at each time increment and uses the worst result to determine damage initiation. The M-K criterion can be used in conjunction with the Mises and Johnson-Cook plasticity models, including kinematic hardening.

For more information, see *Damage Initiation for Ductile Metals*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->M-K Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

3. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

4. If desired, modify the critical deformation severity factors F_{eq}, F_{nn}, and F_{nt}.

   The severity factors each have a default value of 10 and are related to the ratios of equivalent plastic, normal, and tangential strain in the groove area compared to the nominal thickness area. Abaqus/Explicit will ignore severity factors that are set to 0. If all of these parameters are set equal to zero, the M-K criterion is based solely on nonconvergence of the equilibrium and compatibility equations.

5. Select the **Frequency**—the number of increments between calculating the M-K criterion.

   Using the default frequency, 1, can be expensive since Abaqus evaluates each groove at every increment.

6. Select the **Number of imperfections**—the number of angular groove positions to evaluate.

   The groove positions are equally spaced, starting at 0° and ending at $180\frac{(n-1)}{n}$ with respect to the local 1-direction of the material.

7. Enter damage parameters in the **Data** table:

Groove Size

The ratio of the thickness at the groove to the nominal material thickness.

Angle

The starting angle (in degrees) with respect to the 1-direction of the local material orientation.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Select **Suboptions**->**Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

9. Click **OK** to exit the material editor.

Müschenborn-Sonne forming limit diagram (MSFLD) damage

The **MSFLD** damage initiation criterion is used to predict sheet metal forming limits for arbitrary loading paths. The model works on the basis of equivalent plastic strain and assumes that the forming limit curve represents the sum of the highest attainable equivalent plastic strains. The approach requires transforming the original forming limit curve (without predeformation effects) from the space of major versus minor strains to the space of equivalent plastic strain, $\bar{\varepsilon}^{pl}$, versus the ratio of principal strain rates, $\alpha = \dot{\varepsilon}_{\text{minor}} / \dot{\varepsilon}_{\text{major}}$.

Damage due to bending deformation cannot be evaluated using this model. For more information, see *Damage Initiation for Ductile Metals*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->MSFLD Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Choose one of the following **Definitions**:

   - Select **MSFLD** to enter data in the form of equivalent plastic strains and ratios of minor to major strain rates.
   - Select **FLD** to enter data in the form of major and minor principal strains and the equivalent plastic strain and to have Abaqus transform the data into the Müschenborn-Sonne form.

3. Enter the material parameter, **Accumulation power**, equal to the value of the power coefficient, *N*.
4. If desired, you can change the value of ω ($0 < \omega \leq 1$). Omega is used to filter the ratio of principal strains rates, preventing the ratio from jumping to a higher value due to sudden changes in the strain direction (deformation path); the default value is $\omega = 1$.
5. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

6. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

7. Enter damage parameters in the **Data** area (parameters followed by **MSFLD** or **FLD** are used only for that definition type):

Plastic Strain at Initiation (MSFLD)

Equivalent plastic strain at initiation of localized necking.

Ratio of Principal Strains (MSFLD)

Ratio of minor to major principal strains, α.

Major Principal Strain (FLD)

Major principal strain at damage initiation.

Minor Principal Strain (FLD)

Minor principal strain at damage initiation.

Plastic Strain Rate

Equivalent plastic strain rate.

Temp

Temperature, θ.

Field n

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

9. Click **OK** to exit the material editor.

• *Damage Initiation for Ductile Metals*

Shear damage

The **Shear** damage initiation criterion is a model for predicting the onset of damage due to shear band localization. The model assumes that the equivalent plastic strain at the onset of damage is a function of the shear stress ratio and strain rate. The shear criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->Shear Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Enter the material parameter, k_s.
3. Enter the material parameter, **Accumulation power**, equal to the value of the power coefficient, *N*.
4. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

5. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

6. Enter the damage parameters in the **Data** table:

Fracture Strain

Equivalent fracture strain at damage initiation.

Shear Stress Ratio

The shear stress ratio is defined as $\theta_s = (q + k_s p)/\tau_{max}$, where q is the Mises equivalent stress, p is the pressure stress, and τ_{max} is the maximum shear stress.

Strain Rate

The equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

8. Click **OK** to exit the material editor.

• *Damage Initiation for Ductile Metals*

Hosford-Coulomb damage

The **Hosford-Coulomb** damage initiation criterion is a special case of the ductile criterion in which the equivalent plastic strain at the onset of damage is a function of the stress triaxiality, material parameters, and strain rate.

The Hosford-Coulomb criterion can be used in conjunction with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models. It can also be specified together with any other initiation criteria; each initiation criterion is treated independently.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Ductile Metals->Hosford-Coulomb Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the material parameter, **Accumulation power**, equal to the value of the power coefficient, *N*.
3. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.
4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.
5. Enter the damage parameters in the **Data** table:

a

Material parameter a.

b

Material parameter b.

c

Material parameter c.

n

Material parameter n.

d

Material parameter d.

$\dot{\varepsilon}_0$

Reference strain rate.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might have to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*.

7. Click **OK** to exit the material editor.

• *Damage Initiation for Ductile Metals*

Hashin damage

The **Hashin** damage model predicts anisotropic damage in elastic-brittle materials. It is primarily intended for use with fiber-reinforced composite materials and takes into account four different failure modes: fiber tension, fiber compression, matrix tension, and matrix compression.

For more information, see *Damage Initiation for Fiber-Reinforced Composites*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Fiber-Reinforced Composites->Hashin Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Select $\alpha = 0$ to use the model proposed in 1973 and $\alpha = 1$ to use the 1980 model. (For more information, see *Damage Initiation for Fiber-Reinforced Composites*.)
3. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

5. Enter damage parameters in the **Data** table:

Longitudinal Tensile Strength

Longitudinal tensile strength.

Longitudinal Compressive Strength

Longitudinal compressive strength.

Transverse Tensile Strength

Transverse tensile strength.

Transverse Compressive Strength

Transverse compressive strength.

Longitudinal Shear Strength

Longitudinal shear strength.

Transverse Shear Strength

Transverse shear strength.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*."

7. Select **Suboptions->Damage Stabilization** to enter viscous coefficients and improve the model convergence.

   For more information, see *Damage stabilization*."

8. Click **OK** to exit the material editor.

LaRC05 damage

The **LaRC05** damage 3D model predicts polymer-matrix fiber-reinforced composites materials.
It is primarily intended for use with unidirectional fiber-reinforced composite materials and takes into account four different failure modes: matrix cracking, fiber kinking, fiber splitting, and fiber tension.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Fiber-Reinforced Composites->LaRC05 Damage**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Enter the **Tolerance**.

   The value should be equal to the tolerance within which the damage initiation criterion must be satisfied.

3. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Data** table.

4. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

   Field variable columns appear in the **Data** table.

5. Enter the damage parameters in the **Data** table:

Longitudinal Tensile Strength

Longitudinal tensile strength.

Longitudinal Compressive Strength

Longitudinal compressive strength.

Transverse Tensile Strength

Transverse tensile strength.

Transverse Compressive Strength

Transverse compressive strength.

In Plane Shear Strength

In Plane shear strength.

Fracture plane angle for pure compression

Fracture plane angle for pure compression.

Misalignment angle at failure for pure compression

Misalignment angle at failure for pure compression.

Transverse Shear Strength

Transverse shear strength.

Longitudinal shear friction coefficient

Longitudinal shear friction coefficient.

Transverse shear friction coefficient

Transverse shear friction coefficient.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

You might have to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Select **Suboptions->Damage Evolution** to define the material degradation that takes place once damage begins.

   For more information, see *Damage evolution*.

7. Select **Suboptions->Damage Stabilization** to enter viscous coefficients and improve the model convergence.

   For more information, see *Damage stabilization*.

8. Click **OK** to exit the material editor.

• *Damage Initiation for Fiber-Reinforced Composites*

Mullins effect

The Mullins effect material behavior models stress softening of filled rubber elastomers under quasi-static cyclic loading. Abaqus provides three methods to define the Mullins effect in a material:

- Specify the Mullins effect parameters directly as functions of temperature and/or field variables.
- Use experimental unloading-reloading test data to calibrate the Mullins effect parameters.
- Use user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit.

For more information about the Mullins effect, including the meaning of the Mullins coefficients r, m, and β, see *Stress Softening in Elastomers*.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damage for Elastomers->Mullins Effect**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. To define the Mullins data using material constants only, perform the following steps:
   a. From the **Definition** field, select **Constants**.
   b. To define material damage data that depend on temperature, toggle on **Use temperature-dependent data**.

      A column labeled **Temp** appears in the **Data** table.

   c. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

      Field variable columns appear in the **Data** table.

   d. Enter damage parameters in the **Data** table:

r

Value of the r coefficient in the Mullins effect model. r must be greater than 1.

m

Value of the m coefficient in the Mullins effect model. m must be greater than or equal to zero, and the values of m and β cannot both be zero.

beta

Value of the β coefficient in the Mullins effect model. β must be greater than or equal to zero, and the values of m and β cannot both be zero.

Temp

Temperature, θ.

Field *n*

Predefined field variables.

If you include temperature values in your data, you can specify multiple rows of material data. You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

3. To specify experimental unloading-reloading test data to calibrate the parameters for the Mullins effect, perform the following steps:
   a. From the **Definition** field, select **Test Data Input**.
   b. If desired, enter values for one or two of the damage parameters r, m, and β from the **Define Parameters** options. For this type of Mullins effect definition, Abaqus/CAE uses test data you provide to calculate the remaining damage parameters.

      To provide a value for a damage parameter, toggle it on and specify its value in the appropriate field:

      r

      Value of the r coefficient in the Mullins effect model. r must be greater than 1.

      m

      Value of the m coefficient in the Mullins effect model. m must be greater than or equal to zero, and the values of m and β cannot both be zero.

      beta

      Value of the β coefficient in the Mullins effect model. β must be greater than or equal to zero, and the values of m and β cannot both be zero.

   c. From the **Add Test** menu in the **Edit Material** dialog box, select **Biaxial Test**, **Planar Test**, or **Uniaxial Test** to add the selected type of unloading-reloading curve to the material model.

      You can add multiple versions of each type of test to the material model. Abaqus/CAE names each test you create according to its type and the order in which it was created, so the first two uniaxial material tests you define would be named **Uniaxial Test 1** and **Uniaxial Test 2**.
   d. In the **Test data** table, enter the test data for the selected test:

      Nominal Stress

      Nominal stress, T_U.

      Nominal Strain

      Nominal strain, ϵ_U.

      For detailed information on how to enter tabular data, see *Entering tabular data*.
   e. Repeat the previous two steps to specify additional data calibration tests.
   f. If you want to delete a calibration test, highlight its name in the **Tests** list and click **Delete Test**. When you delete a test, Abaqus/CAE removes the test from the list and renames the existing tests so that the test numbering remains consecutive. For example, if you create three biaxial tests and delete the first one (**Biaxial Test 1**), **Biaxial Test 2** is renamed **Biaxial Test 1**, and **Biaxial Test 3** is renamed **Biaxial Test 2**.
4. To define the Mullins effect by specifying the damage variable in the user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit, perform the following steps:
   a. From the **Definition** field, select **User Defined**.

b. In the **Mullins Properties** field, enter values to specify an array of material properties for this user-defined hyperelastic material. Abaqus/CAE uses this array to populate the variable `PROPS` passed to user subroutine *UMULLINS* and *VUMULLINS*.

See the following sections for more information:

- *Specifying general job settings*
- *UMULLINS*
- *VUMULLINS*

5. Click **OK** to exit the material editor.

Damage evolution

The damage evolution definition defines how the material degrades after one or more damage initiation criteria are met.
Multiple forms of damage evolution might act on a material at the same time—one for each damage initiation criterion that was defined.

The procedure below includes data entries for every type of damage evolution available in the Property module. The selections vary with the current damage initiation form.

1. When you create a damage initiation criterion in the **Edit Material** dialog box, select **Suboptions->Damage Evolution** to specify the associated damage evolution parameters.

   (For information on entering damage initiation criteria, see *Defining damage*.)

2. Select the **Type** of damage evolution:

Displacement

Displacement damage evolution defines damage as a function of the total (for elastic materials in cohesive elements) or the plastic (for bulk elastic-plastic materials) displacement after damage initiation. This type corresponds to the **Displacement at Failure** field in the **Data** table.

Energy

Energy damage evolution defines damage in terms of the energy required for failure (fracture energy) after the initiation of damage. This type corresponds to the **Fracture Energy** field in the **Data** table.

3. Select the **Softening** method:

Linear

Linear softening specifies a linear softening stress-strain response for linear elastic materials or a linear evolution of the damage variable with deformation for elastic-plastic materials. Linear softening is the default method.

Exponential

Exponential softening specifies an exponential softening stress-strain response for linear elastic materials or an exponential evolution of the damage variable with deformation for elastic-plastic materials.

Tabular

Tabular softening specifies the evolution of the damage variable with deformation in tabular form and is available only when you select **Displacement** for the type. The **Displacement at Failure** field in the **Data** table is replaced by a **Damage Variable** field and a **Displacement** field, and you can add additional rows to define the displacements.

4. Select the **Mixed mode behavior** (for materials associated with cohesive elements only):

Mode-Independent

Mode-independent is the default selection.

Tabular

Tabular mixed mode behavior specifies the fracture energy or displacement (total or plastic) directly as a function of the shear-normal mode mix for cohesive elements. This method must be used when you select the **Displacement** type with cohesive elements.

Power Law

Power law mixed mode behavior specifies the fracture energy as a function of the mode mix by means of a power law mixed mode fracture criterion; it is available only when you select the **Energy** type with cohesive elements. The **Fracture Energy** field in the **Data** table is replaced by fracture energy in the normal mode and first direction and second direction shear mode components.

BK

The BK mixed mode behavior specifies the fracture energy as a function of the mode mix by means of the Benzeggagh-Kenane mixed mode fracture criterion. The **Data** table entries are the same as those for the **Power Law**.

5. Select the **Degradation** to determine how Abaqus combines damage evolution when multiple forms are active:

Maximum

The maximum degradation form indicates that the current damage evolution mechanism will interact with other damage evolution mechanisms in a maximum sense to determine the total damage from multiple mechanisms. Maximum is the default selection.

Multiplicative

The multiplicative degradation form indicates that the current damage evolution mechanism will interact in a multiplicative manner with other damage evolution mechanisms defined using this form to determine the total damage from multiple mechanisms. Other damage evolution mechanisms defined using the maximum degradation will interact with the combination of those using the multiplicative form.

6. Select the **Mode Mix Ratio** to use in conjunction with the **Mixed mode behavior** definition (for cohesive elements):

Energy

The energy mixed mode ratio defines the mode mix in terms of a ratio of fracture energy in the different modes. This definition is the default, and it must be used when you select **Power Law** or **BK** for the **Mixed mode behavior**.

Traction

The traction mixed mode ratio defines the mode mix in terms of a ratio of traction components.

7. When you select **Power Law** or **BK** for the **Mixed mode behavior** for cohesive elements, toggle on **Power** and enter the exponent in the power law or the Benzeggagh-Kenane criterion that defines the variation of fracture energy with mode mix for cohesive elements.

8. For the Hashin damage evolution model, the **Data** table contains the following fields:
   - **Fiber Tensile Fracture Energy**
   - **Fiber Compressive Fracture Energy**
   - **Matrix Tensile Fracture Energy**
   - **Matrix Compressive Fracture Energy**

   For more information, see *Damage Evolution and Element Removal for Fiber-Reinforced Composites*.
9. To define the damage evolution data that depend on the effective rate of separation, toggle on **Use rate-dependent data**.

   A column labeled **rate-dependent** appears in the **Data** table.
10. To define damage evolution data that depend on temperature, toggle on **Use temperature-dependent data**.

    A column labeled **Temp** appears in the **Data** table.
11. To define damage evolution data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

    Field variable columns appear in the **Data** table.
12. Enter damage evolution parameters in the **Data** table.

    You might need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
13. Click **OK** to save the damage evolution data and return to the material editor.

- *Damage Evolution and Element Removal for Ductile Metals*
- *Damage Evolution and Element Removal for Fiber-Reinforced Composites*
- *Connector Damage Behavior*

Damage stabilization

This option is used to specify viscosity coefficients used in the viscous regularization scheme for the damage model for traction separation laws and fiber-reinforced materials. Viscous regularization is intended to improve convergence as the material fails.

Defining damage stabilization for traction separation laws

Damage stabilization is available for use with the following damage models for traction separation laws:

- Maxe and Quade damage models. (For information on entering the Maxe and Quade damage initiation criterion in Abaqus/CAE, see *Maximum or quadratic nominal strain damage*.)
- Maxs and Quads damage models. (For information on entering the Maxs and Quads damage initiation criterion in Abaqus/CAE, see *Maximum or quadratic nominal stress damage*.)
- Maxps and Maxpe damage models. (For information on entering the Maxps and Maxpe damage initiation criterion in Abaqus/CAE, see *Maximum principal stress or strain damage*.)

You define damage stabilization for traction separation laws by entering the viscosity coefficient.

Defining damage stabilization for fiber-reinforced materials

Damage stabilization is available for use with the following damage model for fiber-reinforced materials:

- Hashin damage model. (For information on entering the Hashin damage initiation criterion in Abaqus/CAE, see *Hashin damage*.)

For more information, see *Viscous Regularization*.

You define damage stabilization for fiber-reinforced materials by entering viscosity coefficients for each of the potential failure modes:

- Fiber tensile failure
- Fiber compressive failure
- Matrix tensile failure
- Matrix compressive failure

Each of the viscous coefficients should be small compared to the increment size.

Defining other mechanical models

You can model deformation plasticity, damping, expansion, brittle cracking, equations of state, viscosity, superelasticity and crush stress.

In this section:

- *Defining deformation plasticity*
- *Defining damping*
- *Defining thermal expansion*
- *Defining brittle cracking*
- *Defining equations of state*
- *Defining viscosity*
- *Defining superelasticity*
- *Defining crush stress*
- *Defining plasticity correction*

Defining deformation plasticity

Abaqus/Standard provides a deformation theory Ramberg-Osgood plasticity model for use in developing fully plastic solutions for fracture mechanics applications in ductile metals. The model is most commonly applied in static loading with small-displacement analysis for which the fully plastic solution must be developed in a part of the model.

See *Deformation Plasticity*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Deformation Plasticity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Enter the following data in the **Data** table:

Young's Modulus

Young's modulus, E, defined as the slope of the stress-strain curve at zero stress.

Poisson's Ratio

Poisson's ratio, ν.

Yield Stress

Yield stress, σ^0.

Exponent

Hardening exponent, n, for the plastic (nonlinear term).

Yield Offset

Yield offset, α.

Temp

Temperature.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

4. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining damping

You can define damping for mode-based analyses and for direct-integration dynamic analysis in Abaqus/Standard and for explicit dynamic analysis in Abaqus/Explicit. See *About Dynamic Analysis Procedures*, and *Material Damping*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Damping**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. In the **Alpha** field, enter a value for the α_R factor to create Rayleigh mass proportional damping. The default is 0. (Units of T^{-1}.)
3. In the **Beta** field, enter a value for the β_R factor to create Rayleigh stiffness proportional damping. The default is 0. (Units of T.)
4. In the **Composite** field, enter a value for the fraction of critical damping to be used with this material in calculating composite damping factors for the modes. The default is 0. (This value applies only to Abaqus/Standard analyses.)
5. In the **Structural** field, enter a value for the s factor to create imaginary stiffness proportional damping. The default is 0.
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining thermal expansion

You can define thermal expansion either by entering thermal expansion coefficients in the **Edit Material** dialog box or, if the thermal strains are complicated functions of field and state variables, with user subroutine *UEXPAN*. See *Thermal Expansion*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Expansion**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the thermal expansion.
3. Toggle on **Use user subroutine UEXPAN** if you want to define the increments of thermal strain in user subroutine *UEXPAN*.
4. If you toggled on **Use user subroutine UEXPAN**, click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

   If you choose to specify thermal expansion coefficients directly in the **Edit Material** dialog box, perform the remaining steps in this procedure.
5. If the thermal expansion is temperature- or field-variable-dependent, enter a value for the **Reference temperature**, θ^0.
6. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
7. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
8. Enter the applicable data in the **Data** table:

Expansion Coeff alpha

Isotropic thermal expansion coefficient, α. (Units of θ^{-1}.)

alpha11, alpha22, and alpha33

Three values to define orthotropic thermal expansion, α_{11}, α_{22}, and α_{33}. (Units of θ^{-1}.)

alpha11, alpha22, alpha33, alpha12, alpha13, and alpha23

Six values to define anisotropic thermal expansion, α_{11}, α_{22}, α_{33}, α_{12}, α_{13}, and α_{23}. (Units of θ^{-1}.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

9. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining brittle cracking

You can use the brittle cracking model in Abaqus/Explicit for applications in which the concrete behavior is dominated by tensile cracking and compressive failure is unimportant.
See *Cracking Model for Concrete* for more information.

In this section:

- *Defining a brittle cracking model*
- *Defining brittle shear*
- *Defining brittle failure*

Defining a brittle cracking model

The brittle cracking model in Abaqus/Explicit is most accurate in applications where the brittle behavior dominates and it is adequate to assume that the material is linear elastic in compression. You can use this model for plain concrete and for other materials such as ceramics or brittle rocks, but it is primarily intended for the analysis of reinforced concrete structures. See *Cracking Model for Concrete*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Brittle Cracking**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and select a method for defining the postcracking behavior:
   - Select **Strain** to specify the postcracking behavior by entering the postfailure stress-strain relationship directly.
   - Select **Displacement** to define the postcracking behavior by entering the postfailure stress/displacement relationship directly.
   - Select **GFI** to define the postcracking behavior by entering the failure stress and the Mode I fracture energy.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. If you selected **Strain** or **Displacement** from the list of **Type** options, enter the following data in the **Data** table:

Direct stress after cracking

Remaining direct stress after cracking, σ_t^I. (Units of $\mathbf{FL^{-2}}$.)

Direct cracking strain

Direct cracking strain, e_{nn}^{ck}. (Enter this value if you selected **Strain** from the list of **Type** options.)

Direct cracking displacement

Direct cracking displacement, u_n^{ck}. (Units of L.) (Enter this value if you selected **Displacement** from the list of **Type** options.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. If you selected **GFI** from the list of **Type** options, enter the following data in the **Data** table:

Failure stress

Failure stress, σ_{tu}^{I}. (Units of $\mathbf{FL^{-2}}$.)

Mode I fracture energy

Mode I fracture energy, G_f^I. (Units of $\mathbf{FL^{-1}}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. Select **Brittle Shear** from the **Suboptions** menu to define the postcracking shear behavior of the material. See *Defining brittle shear*" for details.
8. If desired, select **Brittle Failure** from the **Suboptions** menu to specify the brittle failure criterion. See *Defining brittle failure*" for details.
9. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining brittle shear

An important feature of the cracking model is that, while crack initiation is based on Mode I fracture only, postcracked behavior includes Mode II as well as Mode I.

Mode II shear behavior is based on the common observation that the shear behavior depends on the amount of crack opening. More specifically, the cracked shear modulus is reduced as the crack opens. Therefore, Abaqus/Explicit offers a shear retention model in which the postcracked shear stiffness is defined as a function of the opening strain across the crack.

You must provide postcracking shear data to complete a brittle cracking model definition. See *Shear Retention Model*, for more information.

1. Create a material model as described in *Defining a brittle cracking model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Brittle Shear**.

   A **Suboption Editor** appears.
3. Click the arrow to the right of the **Type** field, and select a method for specifying postcracking shear behavior:
   - Select **Retention Factor** to specify the postcracking shear behavior by entering the shear retention factor-crack opening strain relationship directly.
   - Select **Power Law** to specify the postcracking shear behavior by entering the mateiral parameters for the power law shear retention model.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. If you selected **Retention Factor** from the list of **Type** options, enter the following data in the **Data** table:

   ### Shear retention factor

   Shear retention factor, ρ.

   ### Crack opening strain

   Crack opening strain, $\bar{\epsilon}_{nn}^{ck}$.

   ### Temp

   Temperature.

   ### Field *n*

   Predefined field variables.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
7. If you selected **Power Law** from the list of **Type** options, enter the following data in the **Data** table:

e and p

Material parameters $\tilde{\varepsilon}^{ck}_{max}$ and p.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Click **OK** to return to the **Edit Material** dialog box.

Defining brittle failure

When one, two, or all three local direct cracking strain or displacement components at a material point reach the value defined as the failure strain or displacement, the material point fails and all the stress components are set to zero. If all of the material points in an element fail, the element is removed from the mesh. See *Brittle Failure Criterion*, for more information.

1. Create a material model as described in *Defining a brittle cracking model*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Brittle Failure**.

   A **Suboption Editor** appears.
3. Select an option from the list of **Failure criteria**:
   - Select **Unidirectional** to indicate that an element will be removed when any local direct cracking strain (or displacement) component reaches the failure value.
   - Select **Bidirectional** to indicate that an element will be removed when any two direct cracking strain (or displacement) components reach the failure value.
   - Select **Tridirectional** to indicate that an element will be removed when all three possible direct cracking strain (or displacement) components reach the failure value.
4. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
6. Enter the following data in the **Data** table:

Direct cracking failure strain or displacement

The value you enter depends on the method you chose in the **Edit Material** dialog box for specifying postcracking behavior (as described in *Defining a brittle cracking model*").

If you selected **Strain**, enter the direct cracking failure strain, $\left(e_{nn}^{ck}\right)_f$.

If you selected **Displacement** or **GFI**, enter the direct cracking failure displacement, $\left(u_n^{ck}\right)_f$. (Units of L.)

Temp

Temperature.

Field *n*

Predefined field variables.

7. Click **OK** to return to the **Edit Material** dialog box.

Defining equations of state

The **Edit Material** dialog box allows you to define a hydrodynamic model in the form of an equation of state.

In this section:

- *Defining an equation of state*
- *Defining an ignition and growth equation of state*
- *Defining plastic compaction behavior for an equation of state*
- *Defining detonation points for an explosive material*
- *Defining tensile failure for an equation of state*

Defining an equation of state

You can use the **Edit Material** dialog box to define a hydrodynamic material model in which the material's volumetric strength is determined by an equation of state.
For more information, see *Equation of State*.

1. From the menu in the **Edit Material** dialog box, select **Mechanical->Eos**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and select the type of equation of state that you want to define:
   - Select **Ideal Gas** to define an ideal gas equation of state. For more information, see *Ideal Gas Equation of State*.
   - Select **JWL** to define a Jones-Wilkins-Lee explosive equation of state. For more information, see *JWL High Explosive Equation of State*.
   - Select **Us-Up** to define a linear $U_s - U_p$ equation of state. For more information, see *Mie-Grüneisen Equations of State*.
   - Select **Ignition and growth** to define an equation of state that models shock initiation and detonation wave propagation. For more information, see *Ignition and Growth Equation of State*.
   - Select **Tabular** to define a tabulated equation of state linear in energy. For more information, see *Tabulated Equation of State*.
3. If you selected **Ideal Gas** in Step 2, enter the following data in the **Data** table:

Gas Constant

Gas constant, R. (Units of $JM^{-1}K^{-1}$.)

Ambient Pressure

The ambient pressure, p_A (Units of FL^{-2}).

For detailed information on how to enter data, see *Entering tabular data*.

4. If you selected **JWL** in Step 2, enter the following data in the **Data** table:

Detonation Wave Speed

Detonation wave speed, C_d. (Units of LT^{-1}.)

A and B

Material constants A, and B. (Units of FL^{-2}.)

omega, R1, and R2

Material constants ω, R_1, and R_2. (Dimensionless.)

Detonation Energy Dens

Detonation energy density, E_0. (Units of JM^{-1}.)

Pre-deton Bulk Modulus

Pre-detonation bulk modulus, K_{pd}. (Units of FL^{-2}.)

For detailed information on how to enter data, see *Entering tabular data*.

5. If you selected **Us-Up** in Step 2, enter the following data in the **Data** table:

c0

Reference sound speed, c_0. (Units of LT^{-1}.)

s

Slope of the $U_s - U_p$ curve, s. (Dimensionless.)

Gamma0

Grüneisen ratio, Γ_0. (Dimensionless.)

For detailed information on how to enter data, see *Entering tabular data*.

6. If you selected **Tabular** in Step 2, enter the following data in the **Data** table. The volumetric strain values must be arranged in descending order (that is, from the most tensile to the most compressive states).

f1

f_1. (Units of FL^{-2}.)

f2

f_2. (Dimensionless.)

epsilon_vol

Volumetric strain ε_{vol}. (Dimensionless.)

For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected **Ignition and growth** in Step 2, see *Defining an ignition and growth equation of state* for detailed instructions.
8. If you selected **JWL** in Step 2, select **Detonation Point** from the **Suboptions** menu to define detonation points for the explosive material. For detailed instructions, see *Defining detonation points for an explosive material*.
9. If you selected **Us-Up** or **Tabular** in Step 2, you can select **Eos Compaction** from the **Suboptions** menu to specify plastic compaction behavior for a ductile porous material. For detailed instructions, see *Defining plastic compaction behavior for an equation of state*.
10. If you selected **JWL**, **Us-Up**, or **Tabular** in Step 2, you can select **Tensile Failure** from the **Suboptions** menu to specify a tensile failure model and criterion. For detailed instructions, see *Defining tensile failure for an equation of state*.

11. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining an ignition and growth equation of state

The ignition and growth equation of state models shock initiation and detonation wave propagation for a solid high-explosive material.

For more information, see *Ignition and Growth Equation of State*.

1. Define an ignition and growth equation of state, as described in *Defining an equation of state*.
2. In the **Detonation energy** field, enter a value for E_d. The default value is 0.
3. On the **Solid Phase** tabbed page, enter the following material constants for the unreacted solid explosive in the **Data** table:

A and B

Material constants A and B. (Units of FL^{-2}.)

omega, R1, and R2

Material constants ω, R_1, and R_2. (Dimensionless.)

For detailed information on how to enter data, see *Entering tabular data*.

4. On the **Gas Phase** tabbed page, enter the following material constants for the reacted gas product in the **Data** table:

A and B

Material constants A and B. (Units of FL^{-2}.)

omega, R1, and R2

Material constants ω, R_1, and R_2. (Dimensionless.)

For detailed information on how to enter data, see *Entering tabular data*.

5. On the **Reaction Rate** tabbed page, enter the following data in the **Data** table:

I

Constant. (Units of T^{-1}.)

a

Product covolume. (Dimensionless.)

b

Exponent on the unreacted fraction (ignition term). (Dimensionless.)

x

Exponent (ignition term). (Dimensionless.)

G1

First burn rate coefficient. (Units depend on units of pressure.)

c

Exponent on the unreacted fraction (growth term). (Dimensionless.)

d

Exponent on the reacted fraction (growth term). (Dimensionless.)

y

Pressure exponent (growth term). (Dimensionless.)

G2

Second burn rate coefficient. (Units depend on units of pressure.)

e

Exponent on the unreacted fraction (completion term). (Dimensionless.)

g

Exponent on the reacted fraction (completion term). (Dimensionless.)

z

Pressure exponent (completion term). (Dimensionless.)

Fig(max)

Initial reacted fraction, F_{ig}^{max}. (Dimensionless.)

FG1(max)

Maximum reacted fraction for the growth term, F_{G1}^{max}. (Dimensionless.)

FG2(min)

Minimum reacted fraction for the completion term, F_{G2}^{min}. (Dimensionless.)

For detailed information on how to enter data, see *Entering tabular data*.

6. On the **Gas Specific** tabbed page, enter specific heat data for the reacted gas product.
   a. Toggle on **Use temperature-dependent data** to define data that depend on temperature. A column labeled **Temp** appears in the **Data** table.
   b. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.

c. Enter the following data in the **Data** table:

Specific Heat

Specific heat of the reacted gas product, per unit mass. (Units of $JM^{-1}\theta^{-1}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining plastic compaction behavior for an equation of state

If you define a linear $U_s - U_p$ or tabulated equation of state (as described in *Defining an equation of state*"), you can use the **Suboption Editor** to specify plastic compaction behavior for a ductile porous material. For more information, see *P–alpha Equation of State*.

1. Define a $U_s - U_p$ or tabulated equation of state, as described in *Defining an equation of state*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Eos Compaction**.

   A **Suboption Editor** appears.
3. In the **Reference sound speed in the porous material** field, enter a value for c_e. (Units of LT^{-1}.)
4. In the **Value of the porosity of the unloaded material** field, enter a value for n_0. (Dimensionless.)
5. In the **Pressure required to initialize plastic behavior** field, enter a value for p_e. (Units of FL^{-2}.)
6. In the **Compaction pressure at which all pores are crushed** field, enter a value for p_s. (Units of FL^{-2}.)
7. Click **OK** to return to the **Edit Material** dialog box.

Defining detonation points for an explosive material

You can define any number of detonation points for an explosive material. Coordinates of the points must be defined along with a detonation delay time. Each material point responds to the first detonation point that it sees. For more information, see *JWL High Explosive Equation of State*.

1. Define a Jones-Wilkins-Lee (JWL) equation of state, as described in *Defining an equation of state*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Detonation Point**.
   A **Suboption Editor** appears.
3. Enter the following data in the **Data** table:

   X

   Coordinate 1 of the detonation point.

   Y

   Coordinate 2 of the detonation point.

   Z

   Coordinate 3 of the detonation point.

   Detonation Delay Time

   Detonation delay time (total time, as defined in *Conventions*). The default is 0.

   For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Defining tensile failure for an equation of state

The tensile failure model for an equation of state uses the hydrostatic pressure stress as a failure measure to model dynamic spall (the crumbling of a material) or pressure cutoff.
For more information, see *Tensile Failure Model*.

1. Define a Jones-Wilkins-Lee (JWL), Us-Up, or tabular equation of state, as described in *Defining an equation of state*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Tensile Failure**.

   A **Suboption Editor** appears.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the hydrostatic cutoff stress depends.
4. Select **Use temperature-dependent data** to define the data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Specify the failure choice for the failed material points.

   - Select **Allow element deletion when the failure criterion is met** to allow element removal.
   - Specify behavior for the pressure stress and deviatoric stresses.

     1. From the **Pressure** list, select the behavior for the pressure stress.

        - Select **BRITTLE** to require the pressure stress to be compressive when the failure criterion is met.
        - Select **DUCTILE** to limit the pressure stress by the hydrostatic cutoff stress when the failure criterion is met.

     2. From the **Shear** list, select the behavior for the deviatoric stresses.

        - Select **BRITTLE** to set the deviatoric stress to zero when the failure criterion is met.
        - Select **DUCTILE** to specify that the deviatoric stresses are unaffected when the failure criterion is met.
6. Enter the following data in the **Data** table:

   #### Hydrostatic cutoff stress

   Value of the hydrostatic cutoff stress, σ_{cutoff} (positive in tension).

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to return to the **Edit Material** dialog box.

Defining viscosity

You can define Newtonian viscosity for a material. See *Viscosity*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Viscosity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
4. Enter the following data in the **Data** table:

Dynamic Viscosity

Dynamic viscosity.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining superelasticity

You can use the superelastic material model to model Nitinol-type materials that undergo solid-solid, martensitic phase transformation and exhibit superelastic response.
This model can also be used to model the stress-induced transformation of austenite to single variant martensite, the stress-induced transformation of martensite to austenite, reorientation of martensite, and the shape setting process.

In this section:

- *Defining a superelasticity model*
- *Defining superelastic hardening*
- *Defining superelastic hardening modifications*

Defining a superelasticity model

The superelastic model and the plasticity model for superelastic materials are based on the uniaxial stress-strain response of phase transforming materials.
Such materials (e.g., Nitinol) are in the austenite phase under no loading conditions. Austenite is assumed to follow isotropic linear elasticity. On loading the material, the austenite phase starts transforming into martensite beyond a certain stress.

In the superelastic model martensite is also assumed to follow isotropic linear elasticity. During the phase transformation, elastic properties are calculated from the elastic constants of austenite and martensite.

In the plasticity model for superelastic materials martensite is assumed to follow an elastoplastic response, with elasticity characterized by the linear elastic model and the plastic behavior represented by the Drucker-Prager model. Martensite exhibits plastic behavior after full transformation.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Superelasticity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Nonassociated** to specify the volumetric transformation strain, ε_V^L. Otherwise, Abaqus assumes the volumetric transformation strain is equal to the uniaxial transformation strain, ε^L.
3. Enter the following data in the **Data** table:

Young's Modulus (Martensite)

Young's modulus of martensite, E_M.

Poisson's Ratio (Martensite)

Poisson's ratio of martensite, ν_M.

Transformation Strain

Uniaxial transformation strain, ε^L.

Start of Transformation (Loading)

Stress at which the transformation begins during loading in tension, σ_{tL}^S.

End of Transformation (Loading)

Stress at which the transformation ends during loading in tension, σ_{tL}^E.

Start of Transformation (Unloading)

Stress at which the reverse transformation begins during unloading in tension, σ_{tU}^S.

End of Transformation (Unloading)

Stress at which the reverse transformation ends during unloading in tension, σ_{tU}^E.

Start of Transformation in Compression (Loading)

Stress at which the transformation begins during loading in compression, as a positive value, σ_{cL}^{S}.

Reference Temperature

Reference temperature, T_0.

Loading

Slope of the stress versus temperature curve for loading, $\left(\frac{\delta\sigma}{\delta T}\right)_L$.

Unloading

Slope of the stress versus temperature curve for unloading, $\left(\frac{\delta\sigma}{\delta T}\right)_U$.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

4. Select **Superelastic Hardening** from the **Suboptions** menu to define piecewise linear hardening of martensite. See *Defining superelastic hardening* for details.
5. Select **Superelastic Hardening Modifications** from the **Suboptions** menu to specify the variation of the transformation stress levels of a superelastic material as a function of the plastic strain. See *Defining superelastic hardening modifications* for details.
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

- *Superelasticity*
- *Plasticity Model for Superelastic Materials*

Defining superelastic hardening

You specify the dependence of the yield stress on the total strain to define the piecewise linear hardening of martensite.

1. Create a material model as described in *Defining a superelasticity model*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Superelastic Hardening**.

   A **Suboption Editor** appears.
3. Enter the following data in the **Data** table:

   Yield Stress

   Yield stress.

   Total Strain

   Total strain.

   For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

• *Plasticity Model for Superelastic Materials*

Defining superelastic hardening modifications

You specify the variation of the transformation stress levels of a superelastic material as a function of the plastic strain. You can specify the data describing the change in transformation stress levels as a function of the plastic strain, or you can use a user subroutine to define this dependency.

1. Create a material model as described in *Defining a superelasticity model*.
2. Define superelastic hardening as described in *Defining superelastic hardening*.
3. From the **Suboptions** menu in the **Edit Material** dialog box, select **Superelastic Hardening Modifications**.

   A **Suboption Editor** appears.

4. To specify the data describing the change in transformation stress levels as a function of the plastic strain, enter the following data in the **Data** table:

   ### Start of Transformation (Loading)

   Stress at which the transformation begins during loading in tension, σ_{tL}^{S}.

   ### End of Transformation (Loading)

   Stress at which the transformation ends during loading in tension, σ_{tL}^{E}.

   ### Start of Transformation (Unloading)

   Stress at which the reverse transformation begins during unloading in tension, σ_{tU}^{S}.

   ### End of Transformation (Unloading)

   Stress at which the reverse transformation ends during unloading in tension, σ_{tU}^{E}.

   ### Plastic Strain

   Plastic strain.

   You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Toggle on **User** to define the dependence of the transformation stress levels on plastic strain in user subroutine *USUPERELASHARDMOD* (for Abaqus/Standard) or *VUSUPERELASHARDMOD* (for Abaqus/Explicit), and do the following:
   a. Click the arrows to the right of the **Number of properties** field to increase or decrease the number of hardening properties you want to specify in the data table.
   b. In the **Data** table, enter the hardening properties.

      You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to return to the **Edit Material** dialog box.

• *Plasticity Model for Superelastic Materials*

Defining crush stress

You can use the crush stress material model to specify the compressive crush stress magnitude of a material at an active crush front for the CZone capability.
See *CZone Crush Stress* for more information.

In this section:

- *Defining a crush stress model*
- *Defining crush stress velocity factor*

Defining a crush stress model

The crush stress can be thought of as the stress of the crushing body at the crushing interface. The crush stress is specified as a material property but does not directly influence the constitutive calculations associated with element integration points; rather, the crush stress limits the contact pressure magnitude at an actively crushing interface. See *CZone Analysis* for more information.

In the crush stress model, the composite material can be a constant or can vary as a function of the angle, θ, between the local x-direction of the crushable element and the contact normal direction projected to the plane of the element.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Crush Stress**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
4. Enter the following data in the **Data** table:

Crush Stress

Crush stress.

Material angle (degrees)

Material angle in degrees.

Temp

Temperature.

Field *n*

Predefined field variables.

You can expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

5. Select **Velocity Factor** from the **Suboptions** menu to define how the approach velocity at a crushing interface influences a material's resistance to crushing (see *Defining crush stress velocity factor* for details).
6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining crush stress velocity factor

You can specify the influence of the approach velocity at a crushing interface on a material's resistance to crushing.

1. Create a material model as described in *Defining a crush stress model*.
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Velocity Factor**.

   A **Suboption Editor** appears.
3. Enter the following data in the **Data** table:

   #### Scaling factor

   Scaling factor.

   #### Relative velocity

   Relative velocity between the two surfaces.

   For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Defining plasticity correction

You can use the plasticity correction material model to estimate the elastic-plastic response of the material with linear elasticity.

In this section:

- *Defining a Ramberg-Osgood model*
- *Defining a Tabular model*

Defining a Ramberg-Osgood model

The Ramberg-Osgood model is based on the observation that the stress versus plastic strain response is linear when plotted in the logarithmic scale.
The total strain is decomposed into the elastic and plastic component.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity Correction**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Select the **Definition** as **Ramberg-Osgood** to define this plastic response for the evaluation of the plasticity correction.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

K, n

Material constants K,n.

Temp

Temperature.

Field *n*

Predefined field variables.

You can expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to create the material and to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Defining a Tabular model

The tabular model of plasticity correction allows you to specify the hardening curve directly by providing the yield stress as a function of the equivalent plastic strain.

1. From the menu bar in the **Edit Material** dialog box, select **Mechanical->Plasticity Correction**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Select the **Definition** as **Tabular** to define this plastic response for the evaluation of the plasticity correction.
3. Toggle on **Use temperature-dependent data** to define data that depend on temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. Enter the following data in the **Data** table:

   #### Yield Stress

   Yield stress.

   #### Plastic Strain

   Plastic strain.

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   You can expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to create the material and to close the **Edit Material** dialog box.
7. **Optional:** Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors* for more information).

Defining thermal material models

This section describes how you can define a thermal material model.

In this section:

- *Specifying thermal conductivity*
- *Including volumetric heat flux due to internal heat generation*
- *Specifying an inelastic heat fraction*
- *Specifying a joule heat fraction*
- *Specifying latent heat data*
- *Defining specific heat*
- *Defining thermal conductance for gasket/cohesive elements*
- *Defining thermal radiation for cohesive elements*

Specifying thermal conductivity

You can use the **Edit Material** dialog box to specify isotropic, orthotropic, or fully anisotropic thermal conductivity. For more information, see *Conductivity*.

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Conductivity**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the thermal conductivity.
3. Toggle on **Use temperature-dependent data** to define conductivity as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which conductivity depends.
5. Enter the applicable data in the **Data** table:

Conductivity

Isotropic conductivity, k. (Units of $JT^{-1}L^{-1}\theta^{-1}$.)

k11, k22, and k33

Three values for orthotropic conductivity, k_{11}, k_{22}, and k_{33}. (Units of $JT^{-1}L^{-1}\theta^{-1}$.)

k11, k12, k22, k13, k23, and k33

Six values for anisotropic conductivity, k_{11}, k_{12}, k_{22}, k_{13}, k_{23} and k_{33}. (Units of $JT^{-1}L^{-1}\theta^{-1}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Including volumetric heat flux due to internal heat generation

You can define a heat flux due to internal heat generation in conjunction with user subroutines.

User subroutine *HETVAL* allows you to define a heat flux due to internal heat generation for heat transfer, coupled thermal-electric, or coupled temperature-displacement analyses. User subroutine *VHETVAL* allows you to define a heat flux due to internal heat generation for coupled temperature-displacement analyses.

Abaqus/Standard calls user subroutine *HETVAL* and Abaqus/Explicit calls *VHETVAL* at each point for which the material definition includes a reference to the user subroutine.

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Heat Generation**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (for more information, see *Browsing and modifying material behaviors*).
3. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
4. In the job editor, click the **General** tab, and specify the file containing user subroutine *HETVAL* or *VHETVAL*. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

- *HETVAL*
- *VHETVAL*

Specifying an inelastic heat fraction

You can specify an inelastic heat fraction to provide for inelastic energy dissipation as a heat source in an adiabatic or fully coupled thermal-stress analysis.

Inelastic heat fractions are typically used in the simulation of high-speed manufacturing processes involving large amounts of inelastic strain, where the heating of the material caused by its deformation significantly influences temperature-dependent material properties. The generated heat is treated as a volumetric heat flux source term in the heat balance equation.

See the following sections for more information:

- *Inelastic Energy Dissipation as a Heat Source*
- *Adiabatic Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Inelastic Heat Fraction**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. In the **Fraction** field, enter the fraction of inelastic dissipation rate that appears as a heat flux per unit volume. The fraction may include a unit conversion factor if required. The default value is 0.9.
3. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying a joule heat fraction

Joule heating arises when the energy dissipated by an electrical current flowing through a conductor is converted into thermal energy. A joule heat fraction is the fraction of dissipated electrical energy released as heat in coupled thermal-electrical problems.

See the following sections for more information:

- *Coupled Thermal-Electrical Analysis*
- *Specifying the Amount of Thermal Energy Generated due to Electrical Current*

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Joule Heat Fraction**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. In the **Fraction** field, enter the fraction of electrical energy released as heat, including any unit conversion factor. The default value is 1.0.
3. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying latent heat data

A material's latent heat models large changes in internal energy during phase change of the material. The latent heat is assumed to be released over a range of temperatures from a lower (solidus) temperature to an upper (liquidus) temperature.

If the phase change occurs within a known temperature range, you can enter the solidus and liquidus temperatures directly in the **Edit Material** dialog box. Otherwise you must model the effect with a user subroutine. See *Latent Heat*, for more information.

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Latent Heat**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the following data in the **Data** table:

Latent Heat

Latent heat per unit mass. (Units of JM^{-1}.)

Solidus Temp

Low end of the temperature range within which the phase changes occurs.

Liquidus Temp

High end of the temperature range within which the phase changes occurs.

For detailed information on how to enter data, see *Entering tabular data*.

3. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining specific heat

You can define the specific heat per unit mass of a material as a function of temperature and field variables. When possible, you should model large changes in internal energy during a phase change with a latent heat definition rather than a specific heat definition. See the following sections for more information:

- *Specific Heat*
- *Specifying latent heat data*

1. From the menu bar in the **Edit Material** dialog box, select **Thermal->Specific Heat**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Choose the specific heat **Type**:
   - Use **Constant Volume** for a structural model in Abaqus/Standard or Abaqus/Explicit.
3. Toggle on **Use temperature-dependent data** to define specific heat as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the specific heat depends.
5. Enter the following data in the **Data** table:

Specific Heat

Specific heat per unit mass. (Units of $JM^{-1}\theta^{-1}$.)

Temp

Temperature

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining thermal conductance for gasket/cohesive elements

You can use the **Edit Material** dialog box to define thermal interaction between the top and bottom surfaces of a coupled temperature-displacement gasket element and coupled temperature-displacement cohesive elements.

For more information, see:

- *Defining a Thermal Interaction for a Cohesive Element*
- *Defining a Thermal Interaction for a Gasket Element*

1. From the menu in the **Edit Material** dialog box, select **Thermal->Gap Conductance**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use pressure-dependent data** to define the gap conductance parameters as a function of pressure.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the gap conductance depends.
4. Enter the following data in the **Data** table:

Gap Conductance

Gap conductance or cohesive separation, *k*.

Gap Clearance, Gap Pressure, or Closure

Gap clearance, gap pressure (if **Use pressure-dependent data** is toggled on), or closure (for coupled temperature-displacement gasket elements).

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

5. Toggle on **User Defined** to define the gap conductance in user subroutine *GAPCON*. Other parameter settings and any data lines are ignored.
6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining thermal radiation for cohesive elements

You can use the **Edit Material** dialog box to define radiative heat transfer between the top and bottom surfaces when the surfaces are separated by a narrow gap. For more information, see *Defining a Thermal Interaction for a Cohesive Element*.

1. From the menu in the **Edit Material** dialog box, select **Thermal->Gap Radiation**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter values for **Secondary Surface Emissivity** and **Main Surface Emissivity**.
3. Enter the following data in the **Data** table:

Effective View Factor

Effective view factor.

Gap Clearance

Gap clearance.

Repeat this data line as often as necessary to define the dependence of the view factor on gap clearance.

For detailed information on how to enter data, see *Entering tabular data*.

4. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining electrical and magnetic material models

This section describes how you can define an electrical or magnetic material model.

In this section:

- *Defining electrical conductivity*
- *Defining dielectric material properties*
- *Defining piezoelectric properties*
- *Defining magnetic permeability*

Defining electrical conductivity

If you are using a material in a coupled thermal-electrical or coupled thermal-electrical-structural analysis, you must define the material's electrical conductivity. For a time-harmonic eddy current analysis, the material's electrical conductivity must be used to define the electromagnetic response of a conductor. You can specify electrical conductivity that is isotropic, orthotropic, or fully anisotropic. See the following sections for more information:

- *Electrical Conductivity*
- *Coupled Thermal-Electrical Analysis*
- *Fully Coupled Thermal-Electrical-Structural Analysis*
- *Eddy Current Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Electrical/Magnetic->Electrical Conductivity**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the electrical conductivity.
3. Toggle on **Use frequency-dependent data** to define electrical conductivity that varies with frequency.

   A column labeled **Frequency** appears in the **Data** table.
4. Toggle on **Use temperature-dependent data** to define electrical conductivity as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which electrical conductivity depends.
6. Enter the applicable data in the **Data** table:

Conductivity

Isotropic electrical conductivity. (Units of $CT^{-1}L^{-1}\varphi^{-1}$.)

s11(E), s22(E), and s33(E)

Three values for orthotropic electrical conductivity, σ^E_{11}, σ^E_{22}, and σ^E_{33}. (Units of $CT^{-1}L^{-1}\varphi^{-1}$.)

s11(E), s12(E), s22(E), s13(E), s23(E), and s33(E)

Six values for anisotropic electrical conductivity, σ^E_{11}, σ^E_{12}, σ^E_{22}, σ^E_{13}, σ^E_{23}, and σ^E_{33}. (Units of $CT^{-1}L^{-1}\varphi^{-1}$.)

Frequency

Frequency in cycles/time.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining dielectric material properties

The **Edit Material** dialog box allows you to define the dielectric properties (electrical permittivity) of a fully constrained material for use in a coupled piezoelectric analysis. For more information, see *Piezoelectric Behavior*.

1. From the menu bar in the **Edit Material** dialog box, select **Electrical/Magnetic->Dielectric (Electrical Permittivity)**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the dielectric constant.
3. Toggle on **Use temperature-dependent data** to define the dielectric constant as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the dielectric constant depends.
5. Enter the applicable data in the **Data** table:

Dielectric constant

Dielectric constant for isotropic behavior. (Units of $C\varphi^{-1}L^{-1}$.)

D11, D22, and D33

Three values to define orthotropic behavior, $D_{11}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, and $D_{33}^{\varphi(\varepsilon)}$. (Units of $C\varphi^{-1}L^{-1}$.)

D11, D12, D22, D13, D23, D33

Six values to define anisotropic behavior, $D_{11}^{\varphi(\varepsilon)}$, $D_{12}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, $D_{13}^{\varphi(\varepsilon)}$, $D_{23}^{\varphi(\varepsilon)}$, $D_{33}^{\varphi(\varepsilon)}$. (Units of $C\varphi^{-1}L^{-1}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining piezoelectric properties

You can define piezoelectric material properties by providing the stress coefficients, e^{φ}_{mij}, or the strain coefficients, d^{φ}_{mkl}. For more information, see *Piezoelectric Behavior*.

1. From the menu bar in the **Edit Material** dialog box, select **Electrical/Magnetic->Piezoelectric**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define piezoelectric properties as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the piezoelectric properties depend.
4. In the **Data** table, enter the piezoelectric stress or strain coefficient matrix. For more information, see *Specifying Piezoelectric Material Properties*. Enter values for temperature and field variables if applicable.
   For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining magnetic permeability

For a time-harmonic eddy current analysis, you must define the material's magnetic permeability. You can specify magnetic permeability that is isotropic, orthotropic, or fully anisotropic. See the following sections for more information:

- *Magnetic Permeability*
- *Eddy Current Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Electrical/Magnetic->Magnetic Permeability**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Specify using nonlinear B-H curve** to define nonlinear magnetic permeability.

   Columns labeled **B** and **H** replace the **Magnetic Permeability** column used for linear data in the **Data** table.
3. Click the arrow to the right of the **Type** field, and specify the directional dependence of the magnetic permeability.
4. Toggle on **Use frequency-dependent data** to define magnetic permeability that varies with frequency.

   A column labeled **Frequency** appears in the **Data** table.

   Note:

   Frequency dependence is not available for nonlinear magnetic permeability.
5. Toggle on **Use temperature-dependent data** to define magnetic permeability as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
6. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which magnetic permeability depends.
7. Enter the applicable data in the **Data** table:

B

Strength of the magnetic flux density vector.

You must provide a single set of values for isotropic nonlinear magnetic permeability and a set of values for each of the three principal directions for orthotropic nonlinear magnetic permeability.

H

Strength of the magnetic field vector.

You must provide a single set of values for isotropic nonlinear magnetic permeability and a set of values for each of the three principal directions for orthotropic nonlinear magnetic permeability.

Magnetic Permeability

Isotropic linear magnetic permeability. (Units of FA^{-2}.)

mu11(E), mu22(E), and mu33(E)

Three values for orthotropic linear magnetic permeability, μ_{11}, μ_{22}, and μ_{33}. (Units of FA^{-2}.)

mu11(E), mu12(E), mu22(E), mu13(E), mu23(E), and mu33(E)

Six values for anisotropic linear magnetic permeability, μ_{11}, μ_{12}, μ_{22}, μ_{13}, μ_{23}, and μ_{33}. (Units of FA^{-2}.)

Frequency

Frequency in cycles/time.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

8. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining other types of material models

This section describes how you can define additional types of material models.

In this section:

- *Defining an acoustic medium*
- *Defining mass diffusion*
- *Defining a fluid-filled porous material*
- *Defining gasket behavior*

Defining an acoustic medium

You can define an acoustic medium to model sound propagation problems in a purely acoustic analysis or in a coupled acoustic-structural analysis. For more information, see *Acoustic Medium*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Acoustic Medium**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Display the **Bulk Modulus** tabbed page.
3. Toggle on **Use temperature-dependent data** to define the bulk modulus as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the bulk modulus depends.
5. Enter the following data in the **Data** table:

Bulk Modulus

Bulk modulus, K_f. (Units of FL^{-2}.) For more information, see *Defining an Acoustic Medium*.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

6. If desired, display the **Volumetric Drag** tabbed page, and toggle on **Include volumetric drag**. For more information, see *Volumetric Drag*.
   If you want to omit volumetric drag data, skip to Step 11.
7. Toggle on **Use temperature-dependent data** to define the volumetric drag data as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
8. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the volumetric drag data depend.
9. Enter the following data in the **Data** table:

Volumetric Drag

Volumetric drag coefficient. (Units of FTL^{-4}.)

Frequency

Frequency. (Cycles/time.) Frequency dependence is active only during frequency domain procedures in Abaqus/Standard.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

10. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining mass diffusion

You can use the **Edit Material** dialog box to define certain aspects of mass diffusion.

In this section:

- *Defining diffusivity*
- *Defining general temperature-driven mass diffusion*
- *Defining pressure-driven mass diffusion*
- *Defining solubility*

Defining diffusivity

Diffusivity defines the diffusion, or movement, of one material through another. The governing equations for mass diffusion are an extension of Fick's equations: they allow for nonuniform solubility of the diffusing substance in the base material and for mass diffusion driven by gradients of temperature and pressure. See the following sections for more information:

- *Diffusivity*
- *Mass Diffusion Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Other->Mass Diffusion->Diffusivity**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the diffusivity.
3. Select a **Law** option to specify how you want to define diffusivity behavior:
   - Select **General** to choose general mass diffusion behavior.
   - Select **Fick** to choose Fick's diffusion law.

   For more information, see *Defining Diffusivity*.
4. Toggle on **Use temperature-dependent data** to define diffusivity data as a function of temperature. A column labeled **Temp** appears in the **Data** table.
5. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the diffusivity data depend.
6. In the **Data** table, enter the applicable data:

D

Isotropic diffusivity. (Units of L^2T^{-1}.)

D11, D22, and D33

Orthotropic diffusivity terms. (Units of L^2T^{-1}.)

D11, D12, D22, D13, D23, D33

Anisotropic diffusivity terms. (Units of L^2T^{-1}.)

Concentration

Mass concentration of the diffusing material.

Temp

Temperature.

Field *n*

Predefined field variables.

You may need to expand the dialog box to see all the columns in the **Data** table. For detailed information on how to enter data, see *Entering tabular data*.

7. To describe temperature-driven diffusion, select **Soret Effect** from the **Suboptions** menu. (This option is valid only if you selected **General** in Step 3.) See *Defining general temperature-driven mass diffusion*" for detailed instructions.
8. To describe pressure-driven mass diffusion, select **Pressure Effect** from the **Suboptions** menu. See *Defining pressure-driven mass diffusion*" for detailed instructions.
9. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining general temperature-driven mass diffusion

The Soret effect factor, κ_s, governs temperature-driven mass diffusion. You can define the Soret effect factor as a function of concentration, temperature, and/or field variables. See *Diffusivity*, for more information.



Note:

You can specify the Soret effect factor only if you select general mass diffusion behavior in the diffusivity definition. (If you select Fick's diffusion law, the Soret effect factor is calculated automatically.) For more information, see *Fick's Law*.

1. Define diffusivity as described in *Defining diffusivity*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Soret Effect**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define the Soret effect factor as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables included in the definition of the Soret effect factor.
5. Enter the following data in the **Data** table:

kappa_s

Soret effect factor, κ_s. (Units of $F^{1/2}L^{-1}$.)

Concentration

Mass concentration of the diffusing material.

Temp

Temperature.

Field *n*

Predefined field variables.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining pressure-driven mass diffusion

The pressure stress factor, κ_p, governs mass diffusion driven by the gradient of the equivalent pressure stress. You can define the pressure stress factor as a function of concentration, temperature, and/or field variables. See *Diffusivity*, for more information.

1. Define diffusivity as described in *Defining diffusivity*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Pressure Effect**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define the pressure stress factor as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables included in the definition of the pressure stress factor.
5. Enter the following data in the **Data** table:

kappa_p

Pressure stress factor, κ_p. (Units of $LF^{-1/2}$.)

Concentration

Mass concentration of the diffusing material.

Temp

Temperature.

Field *n*

Predefined field variables.

6. Click **OK** to return to the **Edit Material** dialog box.

Defining solubility

Solubility, s, is used to define the "normalized concentration," ϕ, of the diffusing phase in a mass diffusion process:

$$\phi = c/s,$$

where c is the concentration. The normalized concentration is often also referred to as the "activity" of the diffusing material, and the gradients of the normalized concentration, along with gradients of temperature and pressure stress, drive the diffusion process. For more information, see the following sections:

- *Diffusivity*
- *Mass Diffusion Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Other->Mass Diffusion->Solubility**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define solubility as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the solubility depends.
4. Enter the following data in the **Data** table:

Solubility

Solubility. (Units of $\text{PLF}^{-1/2}$.)

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

5. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining a fluid-filled porous material

You can define specific properties for a fluid-filled porous material.
This type of porous medium is considered in a coupled pore fluid diffusion/stress analysis (*Coupled Pore Fluid Diffusion and Stress Analysis*).

In this section:

- *Defining a swelling gel*
- *Defining moisture swelling*
- *Defining anisotropic swelling*
- *Defining permeability*
- *Defining pore fluid expansion*
- *Defining porous bulk moduli*
- *Defining sorption*
- *Defining normal flow across gap surfaces*
- *Defining tangential flow across gap surfaces*
- *Defining thermal convection across gap surfaces*

Defining a swelling gel

You can model the growth of gel particles that swell and trap wetting liquid in a partially saturated porous medium. For more information, see *Swelling Gel*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Gel**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the following data in the **Data** table:

r_a(dry)

Radius of gel particles when completely dry, r_a^{dry}. (Units of L.)

r_a(f)

Fully swollen radius of gel particles, r_a^f. (Units of L.)

k_a

Number of gel particles per unit volume, k_a. (Units of L^{-3}.)

tau_1

Relaxation time constant for long-term swelling of gel particles, τ_1. (Units of T.)

3. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining moisture swelling

The **Edit material** dialog box allows you to define the saturation-driven volumetric swelling of the solid skeleton of a porous medium in partially saturated flow conditions. You can use this type of material definition in the analysis of coupled wetting liquid flow and porous medium stress. See the following sections for more information:

- *Moisture Swelling*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

1. From the menu bar in the **Edit Material** dialog box, select **Other**->**Pore Fluid**->**Moisture Swelling**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Enter the following data in the **Data** table:

Strain

Volumetric moisture swelling strain, ε^{ms}.

Saturation

Saturation, *s*. This value must lie in the range $0.0 \leq s \leq 1.0$.

For detailed information on how to enter data, see *Entering tabular data*.

3. If you want to define anisotropic swelling, select **Ratios** from the **Suboptions** menu. See *Defining anisotropic swelling*" for detailed instructions.
4. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining anisotropic swelling

You can include anisotropy in moisture swelling behavior by defining the ratios r_{11}, r_{22}, and r_{33} such that two or more of the three ratios differ. The orientation of the moisture swelling strain directions depends on the user-specified local orientation (see *Orientations*).

1. Define moisture swelling behavior, as described in *Defining moisture swelling*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Ratios**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define the anisotropy ratios as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables included in the definition of the anisotropy ratios.
5. In the **Data** table, enter the anisotropy ratios r_{11}, r_{22}, and r_{33}. Enter values for temperature and field variables if applicable.

   For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Defining permeability

Permeability is the relationship between the volumetric flow rate per unit area of a particular wetting liquid through a porous medium and the gradient of the effective fluid pressure.

You can use the **Edit Material** dialog box to define certain aspects of permeability.

For more information, see *Permeability*.

In this section:

- *Defining permeability in an Abaqus/Standard analysis*
- *Defining the saturation dependence of permeability*
- *Defining the velocity dependence of permeability*

Defining permeability in an Abaqus/Standard analysis

You must specify permeability for a wetting liquid in Abaqus/Standard for an effective stress/wetting liquid diffusion analysis. For more information, see *Permeability*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Permeability**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify the directional dependence of the permeability.
3. Enter a value for **Specific weight of wetting liquid**.
4. Toggle on **Use temperature-dependent data** to define permeability as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
5. Enter the applicable data in the **Data** table:

k

Isotropic permeability of the fully saturated medium, k. (Units of LT^{-1}.)

k11, k22, and k33

Three values for orthotropic permeability, k_{11}, k_{22}, and k_{33}. (Units of LT^{-1}.)

k11, k12, k22, k13, k23, and k33

Six values for anisotropic permeability, k_{11}, k_{12}, k_{22}, k_{13}, k_{23}, and k_{33}. (Units of LT^{-1}.)

Void Ratio

Void ratio, e.

Temp

Temperature.

For detailed information on how to enter data, see *Entering tabular data*.

6. If you want to define the dependence of permeability on saturation, select **Saturation Dependence** from the **Suboptions** menu. See *Defining the saturation dependence of permeability*" for detailed instructions.
7. If you want to define the dependence of permeability on velocity, select **Velocity Dependence** from the **Suboptions** menu. See *Defining the velocity dependence of permeability*" for detailed instructions.
8. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Additional information

- *Permeability*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining the saturation dependence of permeability

You can define the dependence of permeability, $\overline{\mathbf{k}}$, on saturation, *s*, by specifying k_s. Abaqus/Standard assumes by default that $k_s = s^3$ for $s < 1.0$; $k_s = 1.0$ for $s \geq 1.0$. The tabular definition of $k_s\ (s)$ must specify $k_s = 1.0$ for $s \geq 1.0$.

For more information, see *Permeability*.

1. Define permeability, as described in *Defining permeability in an Abaqus/Standard analysis*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Saturation Dependence**. A **Suboption Editor** appears.
3. Enter the following data in the **Data** table:

   #### k_s

   The dependence of permeability on saturation of the wetting liquid.

   #### Saturation

   The fluid saturation ($s = 1$ for a fully saturated medium; $s = 0$ for a completely dry medium). For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Defining the velocity dependence of permeability

Permeability is defined, in general, by Forchheimer's law, which accounts for changes in permeability as a function of fluid flow velocity. The **Suboption Editor** allows you define the velocity coefficient as a function of the void ratio of the material. For more information, see *Permeability*.

1. Define permeability, as described in *Defining permeability in an Abaqus/Standard analysis*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Velocity Dependence**.

   A **Suboption Editor** appears.
3. Enter the following data in the **Data** table:

   #### Beta

   Velocity coefficient, $\beta(e)$.

   #### Void ratio

   Void ratio, e.

   For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to return to the **Edit Material** dialog box.

Defining pore fluid expansion

You can use the **Edit Material** dialog box to define thermal expansion for the pore fluid in a porous medium. For more information, see *Thermal Expansion*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Pore Fluid Expansion**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. In the **Reference temperature** field, enter a value for reference temperature if the coefficient of thermal expansion is a function of temperature or field variables
3. Toggle on **Use temperature-dependent data** to define the expansion coefficient as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the expansion coefficient depends.
5. Enter the following data in the **Data** table:

   #### Expansion Coeff

   Thermal expansion coefficient, α. (Units of θ^{-1}.)

   #### Temp

   Temperature.

   #### Field *n*

   Predefined field variables.

   For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining porous bulk moduli

You can define the bulk moduli of solid grains and a permeating fluid such that their compressibility can be considered in the analysis of a porous medium. For more information, see *Porous Bulk Moduli*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Porous Bulk Moduli**. (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Toggle on **Use temperature-dependent data** to define the bulk moduli as a function of temperature. A column labeled **Temp** appears in the **Data** table.
3. Enter the following data in the **Data** table:

Bulk mod of grains

Bulk modulus of solid grains. (Units of FL^{-2}.)

Bulk mod of fluids

Bulk modulus of permeating fluid. (Units of FL^{-2}.)

Temp

Temperature.

For detailed information on how to enter data, see *Entering tabular data*.

4. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining sorption

You can define absorption and exsorption behaviors of a partially saturated porous medium in the analysis of coupled wetting liquid flow and porous medium stress. For more information, see *Sorption*.

1. From the menu bar in the **Edit Material** dialog box, select **Other**->**Pore Fluid**->**Sorption**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Display the **Absorption** tabbed page.
3. Click the arrow to the right of the **Law** field, and specify how you want to define absorption behavior:
   - Select **Log** to define absorption behavior by the analytical logarithmic form.
   - Select **Tabular** to define absorption behavior in tabulated form.
4. If you selected **Log** in the previous step, enter the following data in the **Data** table:

   #### A

   A. This value must be positive. (Dimensionless.)

   #### B

   B. This value must be positive. (Units of L^2F^{-1}.)

   #### s0

   s_0. This value must lie in the range $0.01 \leq s_0 < s_1 < 0.9$. The default is 0.01.

   #### s1

   s_1. This value must lie in the range $0.01 \leq s_0 < s_1 < 0.9$. The default is 0.01 plus a very small positive number (since s_1 cannot be equal to s_0).
5. If you selected **Tabular** in Step 3, enter the following data in the **Data** table:

   #### Pore Pressure

   Pore pressure, u_w, with the condition $u_w \leq 0.0$. (Units of FL^{-2}.)

   #### Saturation

   Saturation, *s*. This value must lie in the range $0.01 \leq s \leq 1.0$
6. If you want to include exsorption behavior in the sorption definition, display the **Exsorption** tabbed page, and toggle on **Include exsorption**. Otherwise, skip to Step 12.
7. Click the arrow to the right of the **Law** field, and specify how you want to define exsorption behavior:
   - Select **Log** to define exsorption behavior by the analytical logarithmic form.
   - Select **Tabular** to define exsorption behavior in tabulated form.

8. Toggle on **Include scanning** to define the behavior between absorption and exsorption by a scanning line of constant slope, $(du_w/ds)|_s$. This slope should be larger than the slope of any segment of the absorption or exsorption behaviors.

   If you define sorption behavior without specifying a scanning line slope, Abaqus uses a slope of 1.05 times the largest value of du_w/ds provided in the absorption and exsorption behavior definitions.

9. If you toggled on **Include scanning**, enter a value for the slope of the scanning line in the **Slope** field.
10. If you selected **Log** in Step 7, enter the following data in the **Data** table:

A

A. This value must be positive. (Dimensionless.)

B

B. This value must be positive. (Units of L^2F^{-1}.)

s0

s_0. This value must lie in the range $0.01 \le s_0 < s_1 < 0.9$. The default is 0.01.

s1

s_1. This value must lie in the range $0.01 \le s_0 < s_1 < 0.9$. The default is 0.01 plus a very small positive number (since s_1 cannot be equal to s_0).

11. If you selected **Tabular** in Step 7, enter the following data in the **Data** table:

Pore Pressure

Pore pressure, u_w, with the condition $u_w \le 0.0$. (Units of FL^{-2}.)

Saturation

Saturation, *s*. This value must lie in the range $0.01 \le s \le 1.0$

12. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining normal flow across gap surfaces

You can model normal flow across gap surfaces by defining a fluid leak-off coefficient for the pore fluid material. This coefficient defines a pressure-flow relationship between the cohesive element's middle nodes and their adjacent surface nodes. The fluid leak-off coefficients can be interpreted as the permeability of a finite layer of material on the cohesive element surfaces. See the following sections for more information:

- *Defining the Constitutive Response of Fluid within the Cohesive Element Gap*
- *UFLUIDLEAKOFF*

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Fluid Leakoff**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify how you want to define the fluid leak-off coefficients:
   - Select **Coefficients** to enter the coefficients directly in the **Edit Material** dialog box.
   - Select **User** to define the coefficients in user subroutine *UFLUIDLEAKOFF*. If you select this option, skip to Step 6.
3. Toggle on **Use temperature-dependent data** to define the fluid leak-off coefficients as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the fluid leak-off coefficients depend.
5. Enter the following data in the **Data** table:

Top Coefficient

Fluid leak-off coefficient at the top element surface.

Bottom Coefficient

Fluid leak-off coefficient at the bottom element surface.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining tangential flow across gap surfaces

You can use the **Edit Material** dialog box to define tangential flow constitutive parameters for pore pressure cohesive elements. For more information, see *Defining the Constitutive Response of Fluid within the Cohesive Element Gap*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Gap Flow**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify how you want to define the flow parameters:
   - Select **Newtonian** to define the viscosity for a Newtonian fluid.
   - Select **Power law** to define the consistency and exponent for a power law fluid.
   - Select **Bingham plastic** to define the consistency and yield stress for a Bingham plastic fluid.
   - Select **Herschel-Bulkley** to define the consistency, exponent, and yield stress for a Herschel-Bulkley fluid.
3. Toggle on **Use temperature-dependent data** to define the flow parameters as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the flow parameters depend.
5. If you selected **Newtonian** in Step 2, you can toggle on **Maximum permeability** to enter the maximum permeability value that Abaqus can use.
6. If you selected **Newtonian** in Step 2, enter the following data in the **Data** table:

Viscosity

Pore fluid viscosity, μ.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

7. If you selected **Power law** in Step 2, enter the following data in the **Data** table:

Consistency

Fluid consistency, K.

Exponent

Power law exponent, α.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

8. If you selected **Bingham plastic** in Step 2, enter the following data in the **Data** table:

Consistency

Fluid consistency, K.

Exponent

Power law exponent, α.

Yield Stress

Yield stress, τ_0.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

9. If you selected **Herschel-Bulkley** in Step 2, enter the following data in the **Data** table:

Consistency

Fluid consistency, K.

Yield Stress

Yield stress, τ_0.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

10. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining thermal convection across gap surfaces

You can use the **Edit Material** dialog box to define convective heat transfer with gap flow as well as thermal convection between the gap fluid cohesive elements. For more information, see *Defining Thermal Convection for a Cohesive Element*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Pore Fluid->Gap Convection**.
   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify how you want to define the Nusselt number:
   - Select **Flux** to define a constant flux solution for convective heat transfer.
   - Select **Temperature** to define a constant temperature solution for convective heat transfer.
   - Select **Tabular** to specify the Nusselt number as a function of temperature and/or field variables.
3. If you selected **Tabular** in Step 2, toggle on **Use temperature-dependent data** to define the Nusselt number as a function of temperature.
   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the flow parameters depend.
5. If you selected **Tabular** in Step 2, enter the following data in the **Data** table:

Nusselt Number

The Nusselt number.

Temp

Temperature.

Field *n*

Predefined field variables.

For detailed information on how to enter data, see *Entering tabular data*.

6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining gasket behavior

You can use the **Edit Material** dialog box to define gasket behavior.

In this section:

- *Defining gasket behavior in the thickness direction*
- *Specifying a gasket contact area or contact width for average pressure output*
- *Defining the elastic transverse shear behavior of a gasket*
- *Defining the membrane behavior of a gasket*

Defining gasket behavior in the thickness direction

Abaqus/Standard measures the thickness-direction deformation as the closure between the bottom and top faces of the gasket element; therefore, the thickness-direction behavior must always be defined in terms of closure. In all cases you can define the thickness-direction behavior as a function of temperature and/or field variables. For more information, see the following sections:

- *Defining the Thickness-Direction Behavior of the Gasket*
- *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*

1. From the menu bar in the **Edit Material** dialog box, select **Other->Gasket->Gasket Thickness Behavior**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)
2. Click the arrow to the right of the **Type** field, and specify how you want to define gasket thickness-direction behavior:

   - Select **Elastic-Plastic** to define an elastic-plastic model. For more information, see *Defining a Nonlinear Elastic-Plastic Model*.
   - Select **Damage** to define a damage elasticity model. For more information, see *Defining a Nonlinear Elastic Model with Damage*.
3. Click the arrow to the right of the **Units** field, and specify a unit system for defining thickness-direction behavior:

   - Select **Stress** to define thickness-direction behavior in terms of pressure versus closure. This option is available for all gasket element types.
   - Select **Force** to define thickness-direction behavior in terms of force versus closure or force per unit length versus closure, depending on the element type with which this behavior is used. This option is valid only for link elements and three-dimensional line elements.

     If you select this option, you can select **Contact Area** from the **Suboptions** menu to define contact area or contact width versus closure curves to output an average pressure through variable CS11. See *Specifying a gasket contact area or contact width for average pressure output*" for detailed instructions.

   For more information about selecting a unit system, see *Choosing a Unit System Used to Define the Thickness-Direction Behavior*.
4. Display the **Loading** tabbed page.
5. Click the arrow to the right of the **Yield onset method** field, and select a method for defining the onset of yield:

   - Select **Relative slope drop** to define yield onset as the point at which the slope of the loading curve decreases by a certain percentage from the maximum slope recorded up to that point. Enter the relative drop value in the field provided. The default is 0.1 (or 10%).
   - Select **Closure value** to specify a closure value at which yield occurs. Enter the closure value in the field provided.
6. In the **Tensile stiffness factor** field, enter a the fraction of the initial compressive stiffness that defines the stiffness in tension. The default is 0.001. For more information, see *Numerical Stabilization of the Thickness-Direction Behavior*.
7. Toggle on **Use temperature-dependent data** to define the gasket thickness behavior as a function of temperature.

A column labeled **Temp** appears in the **Data** table.

8. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the gasket thickness behavior depends.
9. In the **Data** table, define loading in terms of pressure versus closure, force versus closure, or force per unit length versus closure. Enter values for temperature and field variables if applicable. For detailed information on how to enter data, see *Entering tabular data*.
10. Display the **Unloading** tabbed page, and toggle on **Include user-specified unloading curves** if desired. The user-specified unloading curves are in addition to the default unloading curve, which is the scaled portion of the loading curve before the point of yield onset.

    If you leave this option toggled off, skip to Step 13.
11. Toggle on **Use temperature-dependent data** to define the unloading curve as a function of temperature.

    A column labeled **Temp** appears in the **Data** table.
12. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the unloading curve depends.
13. In the **Data** table, specify the unloading curve:

    - If you selected **Damage** in Step 2, provide data points of pressure (or force or force per unit length) versus elastic closure up to a given maximum closure. Enter values for temperature and field variables if applicable. For more information, see *Defining a Nonlinear Elastic Model with Damage*.
    - If you selected **Elastic-Plastic** in Step 2, provide data points of pressure (or force or force per unit length) versus closure (elastic plus plastic) for each given plastic closure in ascending values of closure. For more information, see *Defining a Nonlinear Elastic-Plastic Model*.

    For detailed information on how to enter data, see *Entering tabular data*.
14. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Specifying a gasket contact area or contact width for average pressure output

When you define the thickness-direction behavior of the gasket in terms of force or force per unit length versus closure, Abaqus/Standard provides the thickness-direction force or force per unit length as output variable S11. In this case you can define either a contact width or contact area versus closure curve that will be used to obtain the average "contact" pressure at each integration point as output variable CS11.

For more information, see *Defining the Contact Area for Average Contact Pressure Output*.

1. Define gasket thickness-direction behavior as described in *Defining gasket behavior in the thickness direction*."
2. From the **Suboptions** menu in the **Edit Material** dialog box, select **Contact Area**.

   A **Suboption Editor** appears.
3. Toggle on **Use temperature-dependent data** to define the data as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.
4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
5. In the **Data** table, define the contact area or contact width versus closure curve. Include temperature and field variable data if applicable. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to return to the **Edit Material** dialog box.

Defining the elastic transverse shear behavior of a gasket

Abaqus/Standard measures the relative displacement between the bottom and top of the gasket element along the local 2- or 3-directions to define the transverse shear in the gasket. You can use the **Edit Material** dialog box to specify the elastic transverse stiffness as stress (or force, or force per unit length) per unit displacement. For more information, see *Defining the Transverse Shear Behavior of the Gasket*.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Gasket->Gasket Transverse Shear Elastic**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Click the arrow to the right of the **Units** field, and select the unit system in which you will define the transverse shear behavior:

   - Select **Stress** to define the transverse shear stiffness in terms of stress per unit displacement.
   - Select **Force** to define the transverse shear stiffness in terms of force per unit displacement or force per unit length per unit displacement, depending on the element type to which this behavior refers.

   The unit system that you select for transverse shear behavior must be consistent with the unit system selected for thickness-direction behavior (see *Defining gasket behavior in the thickness direction*"). For more information, see *Choosing a Unit System to Define the Transverse Shear Behavior*.

3. Toggle on **Use temperature-dependent data** to define the shear stiffness as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.

4. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the shear stiffness depends.
5. In the **Data** table, specify the shear stiffness. Include temperature and field variable data if applicable. For detailed information on how to enter data, see *Entering tabular data*.
6. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Defining the membrane behavior of a gasket

You can define the linear elastic behavior of the gasket by providing Young's modulus and Poisson's ratio. These data can be a function of temperature and/or field variables. If you do not specify the linear elastic behavior of the gasket, the gasket has no membrane stiffness. In this case you must ensure that the nodes of the elements are restrained adequately in the directions orthogonal to the thickness direction of the gasket.

1. From the menu bar in the **Edit Material** dialog box, select **Other->Gasket->Gasket Membrane Elastic**.

   (For information on displaying the **Edit Material** dialog box, see *Creating or editing a material*.)

2. Toggle on **Use temperature-dependent data** to define the data as a function of temperature.

   A column labeled **Temp** appears in the **Data** table.

3. Click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables on which the data depend.
4. In the **Data** table, enter Young's modulus and Poisson's ratio data. Include temperature and field variable data if applicable. For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to close the **Edit Material** dialog box. Alternatively, you can select another material behavior to define from the menus in the **Edit Material** dialog box (see *Browsing and modifying material behaviors*, for more information).

Creating and editing sections

This section describes how to create and edit sections using the section editor.

In this section:

- *Creating homogeneous solid sections*
- *Creating generalized plane strain sections*
- *Creating Eulerian sections*
- *Creating composite solid sections*
- *Creating electromagnetic solid sections*
- *Creating homogeneous shell sections*
- *Creating composite shell sections*
- *Creating membrane sections*
- *Creating surface sections*
- *Creating general shell stiffness sections*
- *Creating beam sections*
- *Creating truss sections*
- *Creating gasket sections*
- *Creating cohesive sections*
- *Creating acoustic infinite sections*
- *Creating acoustic interface sections*
- *Defining rebar layers*
- *Creating profiles*

Creating homogeneous solid sections

Homogeneous solid sections are used to define the section properties of two-dimensional, three-dimensional, and axisymmetric solid regions. For more information, see *Solid (Continuum) Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Solid** as the section **Category** and **Homogeneous** as the section **Type**, and click **Continue**.

   The solid section editor appears.

4. Select a material for the solid section. If desired, click  to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the section **Plane stress/strain thickness**. If the section will be used with a two-dimensional region, you must specify the section thickness. Abaqus/CAE ignores the thickness information if it is not needed for the region type.
6. Click **OK** to save your changes and to close the solid section editor.

Creating generalized plane strain sections

Generalized plane strain sections are used to define the section properties of two-dimensional planar regions.

You must create a reference point to indicate the reference node required by generalized plane strain elements. In addition to the material and the section thickness at the reference point, you can specify wedge angles about the global 1- and 2-axes, as defined in *Figure 1*. For more information, see *Generalized Plane Strain Elements*.

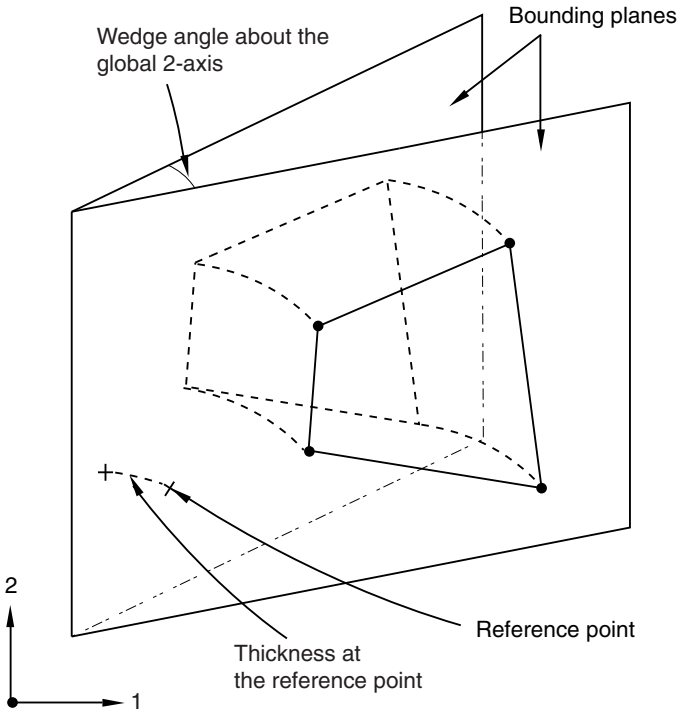


Figure 1: Generalized plane strain section definition.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Solid** as the section **Category** and **Generalized plane strain** as the section **Type**, and click **Continue**.

   The generalized plane strain section editor appears.

4. Select a material for the section. If desired, click to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the material **Thickness at the reference point**.
6. Enter a value for the **Wedge angle about global 1-axis**, the rotation about the global 1-axis at the reference point in radians.
7. Enter a value for the **Wedge angle about global 2-axis**, the rotation about the global 2-axis at the reference point in radians.

8. Click **OK** to save your changes and to close the generalized plane strain section editor.

Additional information

- *The reference point*
- *Creating sections*
- *Choosing the Element's Dimensionality*

Creating Eulerian sections

Eulerian sections are used to specify the materials that can be present in an Eulerian domain. Eulerian sections can be assigned only to Eulerian-type parts, and a single Eulerian section must be assigned to an entire Eulerian part.

An Eulerian section does not create material within an Eulerian part; rather, it provides a list of materials that may exist within the part. The Eulerian part contains only void by default. After creating an Eulerian section, you can add material to the part using a material assignment predefined field (see *Defining a material assignment field*). For more information, see *Eulerian Analysis*. For an overview of modeling Eulerian analyses in Abaqus/CAE, see *Eulerian analyses*."

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Solid** as the section **Category** and **Eulerian** as the section **Type**, and click **Continue**.

   The Eulerian section editor appears.
4. Each material present in the Eulerian section is represented by a row in the **Material Instances** table. To add rows to the table, click mouse button 3 on a row and select **Insert Row Before** or **Insert Row After** from the menu that appears.
5. For each row in the **Material Instances** table, enter the following data:

Base Material

A material that may be present in the Eulerian part. Click in the **Base Material** column, then click the arrow that appears to display the list of available materials, and select the appropriate material.

Instance Name

A name that is used to refer to the base material in assembly-related modules, such as when you are defining the initial composition of an Eulerian domain using the material assignment predefined field (see *Defining a material assignment field*). Separate output is created for each material instance name (see *Viewing output from Eulerian analyses*). In some situations it may be necessary to create multiple, unique material instance names that refer to the same base material; for example, if you want separate output data for different regions of the model that contain the same base material. Abaqus/CAE automatically creates a material instance name when you select a base material; however, you can overwrite this default instance name if desired.

6. Click **OK** to save your changes and to close the Eulerian section editor.

Creating composite solid sections

Composite solid sections are used to define the section properties of three-dimensional regions that are composed of layers of different materials in different orientations. For more information, see *Defining Composite Solid Elements in Abaqus/Standard*.

Composite solid sections can be used only in Abaqus/Standard. Composite solid sections must be assigned only to three-dimensional brick elements with only displacement degrees of freedom. Composite solid elements are primarily intended for modeling convenience. In most cases you should model a composite section as a shell or continuum shell. However, you should use a composite solid section for the following cases:

- When the transverse shear effects are predominant.
- When you cannot ignore the normal stress.
- When you require accurate interlaminate stresses, such as near localized regions of complex loading or geometry.

If the region to which you assign your composite solid section contains multiple elements through its thickness, each element will contain all the material layers defined in the data table, and the analysis results will not be as expected.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Solid** as the section **Category** and **Composite** as the section **Type**, and click **Continue**.

   The composite solid section editor appears.
4. Optionally, enter a layup name. Abaqus/CAE displays this name in a ply stack plot. For more information on naming objects, see *Using basic dialog box components*.
5. If the layers of material in the section are symmetric about a central core, toggle on **Symmetric layers**. Enter the material layers in the data table, starting with the bottom layer in the first row and ending with the central layer. During the analysis Abaqus appends layers to the section definition by repeating the entered layers (including the central layer) in the reverse order to the top of the section. If you name the material layers, each generated layer is labeled in ply stack plots and the output database by adding `Sym_` to the beginning of the repeated layer's original name.
6. Each layer of the composite solid section is represented by a row in the data table.

   Note:

   The stacking direction for the composite layers is determined by the material orientation assigned to the solid region. See *Assigning a material orientation*, for more information.

   To add rows to the table, click mouse button three on a row and select

   Insert Row Before

   or

   Insert Row After

   from the menu that appears. For each layer, enter the following data:

Material

The name of the material forming this layer. Click in the **Material** column, then click the arrow that appears to display the list of available materials, and select the material forming the layer.

Element Relative Thickness

The relative thickness of the layer within each element. Abaqus determines the overall section thickness from the element geometry, which may vary from element to element where the section is defined. Hence, the thickness values that you specify for each layer are relative to the thickness of each element. The actual thickness of a layer is the element thickness times the fraction of the total thickness that is accounted for by each layer. You do not have to use physical units to specify the thickness ratios for the layers, and the sum of the layer relative thicknesses does not have to add to one. For more information, see *Creating a composite layup*.

Orientation Angle

The orientation. The orientation can be specified either as an angle in degrees or as an orientation name. The orientation angle is measured positive counterclockwise around the normal and relative to the section orientation definition.

If you specify an orientation name, Abaqus/CAE assumes a user-defined orientation. You must supply the user subroutine *ORIENT* that contains the definition of the user-defined orientation for the specified orientation name. You cannot define a variable orientation angle using a discrete field; to define ply-by-ply orientation distributions in a composite solid, you must use the composite layup editor (see *Creating and editing composite layups*).

Integration Points

The number of integration points through the thickness. You can specify only odd numbers.

Ply Name

The name of the layer. Abaqus/CAE displays this name when you are viewing the composite plies in the Visualization module and in a ply stack plot.

Naming the layers in a composite solid section is optional. However, if you provide a name for any layer, you must provide names for all of the layers in the section.

7. Click **OK** to save your changes and to close the composite solid section editor.

Creating electromagnetic solid sections

Electromagnetic solid sections are used to define the section properties of two-dimensional and three-dimensional solid regions and must be assigned only to electromagnetic elements in an electromagnetic model. For more information, see *Solid (Continuum) Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Solid** as the section **Category** and **Electromagnetic, Solid** as the section **Type**, and click **Continue**.

   The electromagnetic solid section editor appears.

4. Select a material for the electromagnetic solid section. If desired, click  to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the section **Plane stress/strain thickness**. If the section will be used with a two-dimensional region, you must specify the section thickness. Abaqus/CAE ignores the thickness information if it is not needed for the region type.
6. Click **OK** to save your changes and to close the electromagnetic solid section editor.

Creating homogeneous shell sections

This section describes how to create a homogeneous shell section.

Shell section behavior is defined in terms of the response of the shell section to stretching, bending, shear, and torsion. For more information, see *Shell Section Behavior*.

When you create shell sections, you must choose a section integration method. You can choose to provide the section property data before the analysis (a pre-integrated shell section) or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis.

Shell sections integrated during analysis allow the cross-sectional behavior to be calculated by numerical integration through the shell thickness, thus providing complete generality in material modeling. Any number of material points can be defined through the thickness, and the material response can vary from point to point. This type of shell section is generally used with nonlinear material behavior in the section. It must be used with shells that provide for heat transfer. For more information, see *Using a Shell Section Integrated during the Analysis to Define the Section Behavior*.

Linear moment-bending and force-membrane strain relationships can be defined using pre-integrated shell sections. In this case all calculations are done in terms of section forces and moments. The section properties are specified by an elastic material; optionally, you can also apply an idealization based on assumptions about the expected behavior of the shell. Use a pre-integrated shell section if the response of the shell is linear elastic and its behavior is not dependent on changes in temperature or predefined field variables. For more information, see *Using a General Shell Section to Define the Section Behavior*.

In this section:

- *Creating a homogeneous shell section*
- *Specifying basic properties for homogeneous shell sections*
- *Specifying advanced properties for homogeneous shell section*

Creating a homogeneous shell section

You can create a homogeneous shell section.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Shell** as the section **Category** and **Homogeneous** as the section **Type**, and click **Continue**.

   The shell section editor appears.
4. Choose the **Section integration** method. Choose **During analysis** to specify properties for homogeneous shell sections integrated during the analysis. Choose **Before analysis** to specify properties for pre-integrated homogeneous shell sections.

Specifying basic properties for homogeneous shell sections

On the **Basic** tabbed page:

1. Specify the **Shell thickness**.
   - Choose **Value**, and enter a value for the shell thickness. In continuum shells this value is used to estimate certain section properties, such as hourglass stiffness, which are later computed using the actual thickness computed from the element geometry.
   - Choose **Element distribution**; and select either an analytical field, labeled with an (A), or an element-based discrete field, labeled with a (D), to define a spatially varying element-based shell thickness. Alternatively, you can click  to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
   - Choose **Nodal distribution**; and select either an analytical field, labeled with an (A), or a node-based discrete field, labeled with a (D), to define a spatially varying node-based shell thickness.

     Alternatively, you can click  to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
2. Select a material for the shell section. If desired, click  to create a material; see *Creating or editing a material*, for more information. Linear or nonlinear material behavior can be associated with the section definition. However, if the material response is linear, a more economical approach is to use a generalized shell section.
3. If you are specifying properties for homogeneous shell sections integrated before the analysis, you can specify an **Idealization** to apply to the section based on assumptions about the expected behavior of the shell. For more information, see *Idealizing the Section Response*.
   - Select **No idealization** to account for the complete stiffness of the shell section as determined from the material assignment.
   - Select **Membrane only** if the predominant response of the shell will be in-plane stretching; bending stiffness terms are eliminated from the shell stiffness calculations.
   - Select **Bending only** if the predominant response of the shell will be pure bending; membrane stiffness terms are eliminated from the shell stiffness calculations.
4. If you are specifying properties for homogeneous shell sections integrated during the analysis, select the **Thickness integration rule**.
   - Choose **Simpson** to use Simpson's rule for the shell section integration.
   - Choose **Gauss** to use Gauss quadrature for the shell section integration.

   See *Defining the Shell Section Integration*, for more information.
5. If you are specifying properties for homogeneous shell sections integrated during the analysis, enter the number of thickness integration points. The default number of integration points through the thickness is 5 for Simpson's rule integration and 3 for Gauss quadrature integration. To specify a new value for the number of integration points, you can type the number directly or click the arrows in the **Thickness integration points** text field.
   - If you are using the **Simpson** integration rule, you can specify only odd numbers ranging from 3 to 15.
   - If you are using the **Gauss** integration rule, you can specify numbers ranging from 2 to 15.

Specifying advanced properties for homogeneous shell section

On the **Advanced** tabbed page:

1. Specify the **Section Poisson's ratio** to define the shell thickness behavior.

   - In conventional shell elements that permit finite membrane strains in large-deformation analysis, specifying the section Poisson's ratio causes the shell thickness to change as a function of membrane strains:

     - Toggle on **Use analysis default** to use the default value. In Abaqus/Standard the default value is 0.5, which will enforce incompressible behavior of the element for membrane strains. In Abaqus/Explicit the default is to base the change in thickness on the element material definition.
     - Toggle on **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.

   - In continuum shell elements specifying the section Poisson's ratio defines the thickness behavior for both small- and large-displacement analysis:

     - Toggle on **Use analysis default** to indicate that the change in thickness is based on the element material definition.
     - Toggle on **Specify value**, and enter a value for the Poisson's ratio to cause the shell thickness to change as a function of membrane strains. This value must be between −1.0 and 0.5. A value of 0.5 cannot be used with continuum shells. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.

2. For continuum shell elements, toggle on **Thickness modulus**, and enter a value for the effective thickness modulus. If you do not specify a thickness modulus, Abaqus will try to compute it based on the initial elastic material properties.
3. If you are specifying properties for homogeneous shell sections integrated during the analysis, select a method for defining the **Temperature variation** through the section:

   - Choose **Linear through thickness** to indicate that the temperature at the reference surface and the temperature gradient or gradients through the section are specified. You can use the Load module to specify these temperatures.
   - Choose **Piecewise linear over *n* values** to enter the number of temperature points (values) through the section in the text field provided. You can use the Load module to specify the temperature at each of these points.

4. Toggle on **Density**, and enter a value for the mass per unit surface area of the shell. The mass of the shell includes a contribution from the density in addition to any contribution from the selected material.
5. For most shell sections Abaqus will calculate the transverse shear stiffness values required in the element formulation. If desired, toggle on **Specify values** from the **Transverse Shear Stiffnesses** options to include nondefault transverse shear stiffness effects in the section definition, and enter values for K_{11}, the shear stiffness of the section in the first direction; K_{12}, the coupling term in the shear stiffness of the section; and K_{22}, the shear stiffness of the section in the second direction. If either value K_{11} or K_{22} is omitted or given as zero, the nonzero value will be used for both.

6. If you are specifying properties for homogeneous shell sections integrated during the analysis, click  at the bottom of the shell section editor to define rebar layers in the shell section, as described in *Defining rebar layers*.
7. Click **OK** to save your changes and to close the shell section editor.

Creating composite shell sections

This section describes how to create a composite shell section.

Shell section behavior is defined in terms of the response of the shell section to stretching, bending, shear, and torsion. For more information, see *Shell Section Behavior*. Composite shell sections are composed of layers made of different materials in different orientations.

When you create shell sections, you must choose a section integration method. You can choose to provide the section property data before the analysis (a pre-integrated shell section) or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis.

Shell sections integrated during analysis allow the cross-sectional behavior to be calculated by numerical integration through the shell thickness, thus providing complete generality in material modeling. Any number of material points can be defined through the thickness, and the material response can vary from point to point. This type of shell section is generally used with nonlinear material behavior in the section. It must be used with shells that provide for heat transfer. For more information, see *Using a Shell Section Integrated during the Analysis to Define the Section Behavior*.

Linear moment-bending and force-membrane strain relationships can be defined using pre-integrated shell sections. In this case all calculations are done in terms of section forces and moments. The section properties are specified by elastic material plies; optionally, you can also apply an idealization based on assumptions about the expected behavior or makeup of the shell. Use a pre-integrated shell section if the response of the shell is linear elastic and its behavior is not dependent on changes in temperature or predefined field variables. For more information, see *Using a General Shell Section to Define the Section Behavior*.

In this section:

- *Creating a composite shell section*
- *Specifying basic properties for composite shell sections*
- *Specifying advanced properties for composite shell sections*

Creating a composite shell section

You can create a composite shell section.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Shell** as the section **Category** and **Composite** as the section **Type**, and click **Continue**.

   The shell section editor appears.

4. Choose the **Section integration** method. Choose **During analysis** to specify properties for a composite shell section integrated during the analysis. Choose **Before analysis** to specify properties for a pre-integrated composite shell section.
5. Enter a layup name. Abaqus/CAE displays this name in a ply stack plot. For more information on naming objects, see *Using basic dialog box components*.
6. Click  at the bottom of the shell section editor to define rebar layers in the shell section, as described in *Defining rebar layers*.
7. Click **OK** to save your changes and to close the shell section editor.

Specifying basic properties for composite shell sections

You can specify basic properties for composite shell sections.

On the **Basic** tabbed page:

1. Enter the layup name. For more information on naming objects, see *Using basic dialog box components*.
2. If you are specifying properties for composite shell sections integrated before the analysis, specify the **Idealization** to apply to the section based on assumptions about the expected behavior or makeup of the shell. For more information, see *Idealizing the Section Response*.
   - Select **No idealization** to account for the complete stiffness of the shell section as determined by the material assignments and layer composition.
   - Select **Smeared properties** if you do not know the exact stacking sequence for the material layers in the composite shell. Contributions from each specified layer are smeared across the entire thickness of the shell, resulting in a general response independent of the stacking sequence.
   - Select **Membrane only** if the predominant response of the shell will be in-plane stretching; bending stiffness terms are eliminated from the shell stiffness calculations.
   - Select **Bending only** if the predominant response of the shell will be pure bending; membrane stiffness terms are eliminated from the shell stiffness calculations.
3. If you are specifying properties for composite shell sections integrated during the analysis, select the **Thickness integration rule**.
   - Choose **Simpson** to use Simpson's rule for the shell section integration.
   - Choose **Gauss** to use Gauss quadrature for the shell section integration.

   See *Defining the Shell Section Integration* for more information.
4. If the layers of material in the section are symmetric about a central core, toggle on **Symmetric layers**. Enter the material layers in the data table, starting with the bottom layer in the first row and ending with the central layer. During the analysis Abaqus appends layers to the section definition by repeating the entered layers (including the central layer) in the reverse order to the top of the section. Each generated layer is labeled in ply stack plots and the output database by adding `Sym_` to the beginning of the repeated layer's original name.
5. Each layer of the composite shell section is represented by a row in the data table. To add rows to the table, click mouse button three on a row and select **Insert Row Before** or **Insert Row After** from the menu that appears. For each layer, enter the following data:

Material

The name of the material forming this layer. Click in the **Material** column, then click the arrow that appears to display the list of available materials, and select the material forming the layer.

Thickness

The layer thickness. For continuum shell elements Abaqus determines the thickness from the element geometry, and the thickness may vary through the model for a given section definition. Hence, the thickness values that you specify are only relative thicknesses for each layer. The actual thickness of a layer is the element thickness times the fraction of the total thickness that is accounted for by each layer. You do not have to use physical units to specify the thickness ratios for the layers, and the sum of the layer relative thicknesses does not have to add to one. Abaqus uses the shell thickness to estimate certain section properties, such as hourglass stiffness, which are later computed from the element geometry.

Orientation Angle

The orientation, either as a reference to a section orientation definition or as an orientation angle in degrees. The orientation angle, ϕ, is measured positive counterclockwise around the normal and relative to the section orientation definition.

If either of the two local directions from the section orientation is not in the surface of the shell, ϕ is applied after the section orientation has been projected onto the shell surface. If no section orientation has been defined, ϕ is measured relative to the default shell local directions.

If you specify an orientation name, Abaqus/CAE assumes a user-defined orientation. You must supply the user subroutine *ORIENT* that contains the definition of the user-defined orientation for the specified orientation name. You cannot define a variable orientation angle using a discrete field; to define ply-by-ply orientation distributions in a composite shell, you must use the composite layup editor (see *Creating and editing composite layups*).

Integration Points

The number of integration points through the thickness, if you are specifying properties for composite shell sections integrated during the analysis.

The default number of integration points is 3 for Simpson's rule integration and 2 for Gauss quadrature integration.

- If you are using the **Simpson** integration rule, you can specify only odd numbers.
- If you are using the **Gauss** integration rule, you can specify numbers less than or equal to 7.

Ply Name

The name of the layer. Abaqus/CAE displays this name when you are viewing the composite plies in the Visualization module and in a ply stack plot.

Specifying advanced properties for composite shell sections

You can specify advanced properties for composite shell sections.

On the **Advanced** tabbed page:

1. Specify the **Shell thickness**.
   - Choose **Use section thickness** to use the thickness calculated from the individual layer thicknesses.
   - Choose **Element distribution**; and select either an analytical field, labeled with an (A), or an element-based discrete field, labeled with a (D), to define a spatially varying element-based shell thickness. Alternatively, you can click $f(x)$ to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
   - Choose **Nodal distribution**; and select either an analytical field, labeled with an (A), or a node-based discrete field, labeled with a (D), to define a spatially varying node-based shell thickness. Alternatively, you can click $f(x)$ to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
2. Specify the **Section Poisson's ratio** to define the shell thickness behavior.
   - In conventional shell elements that permit finite membrane strains in large-deformation analysis, specifying the section Poisson's ratio causes the shell thickness to change as a function of membrane strains:
     - Toggle on **Use analysis default** to use the default value. In Abaqus/Standard the default value is 0.5, which will enforce incompressible behavior of the element for membrane strains. In Abaqus/Explicit the default is to base the change in thickness on the element material definition.
     - Toggle on **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.
   - In continuum shell elements specifying the section Poisson's ratio defines the thickness behavior for both small- and large-displacement analysis:
     - Toggle on **Use analysis default** to indicate that the change in thickness is based on the element material definition.
     - Toggle on **Specify value**, and enter a value for the Poisson's ratio to cause the shell thickness to change as a function of membrane strains. This value must be between −1.0 and 0.5. A value of 0.5 cannot be used with continuum shells. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.
3. For continuum shell elements, toggle on **Thickness modulus**, and enter a value for the effective thickness modulus. If you do not specify a thickness modulus, Abaqus will try to compute it based on the initial elastic material properties.
4. If you are specifying properties for composite shell sections integrated during the analysis, select a method for defining the **Temperature variation** through the section:
   - Choose **Linear through thickness** to indicate that the temperature at the reference surface and the temperature gradient or gradients through the section are specified. You can use the Load module to specify these temperatures.

- Choose **Piecewise linear over *n* values** to enter the number of temperature points (values) through the section in the text field provided. You can use the Load module to specify the temperature at each of these points.

5. Toggle on **Density**, and enter a value for the mass per unit surface area of the shell. The mass of the shell includes a contribution from the density in addition to any contribution from the selected material.
6. For most shell sections Abaqus will calculate the transverse shear stiffness values required in the element formulation. If desired, toggle on **Specify values** from the **Transverse Shear Stiffnesses** options to include nondefault transverse shear stiffness effects in the section definition, and enter values for K_{11}, the shear stiffness of the section in the first direction; K_{12}, the coupling term in the shear stiffness of the section; and K_{22}, the shear stiffness of the section in the second direction. If either value K_{11} or K_{22} is omitted or given as zero, the nonzero value will be used for both.

Creating membrane sections

Membrane sections are used to define the section properties of thin surfaces in space that offer strength in the plane of the surface but have no bending stiffness. For more information, see *Membrane Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Shell** as the section **Category** and **Membrane** as the section **Type**, and click **Continue**.

   The membrane section editor appears.
4. Select a material for the membrane section. If desired, click **Create** to create a material; see *Creating or editing a material*, for more information.
5. Specify the **Membrane thickness**.
   - Choose **Value**, and enter a value for the membrane thickness.
   - Choose **Element distribution**; and select either an analytical field, labeled with an (A), or an element-based discrete field, labeled with a (D), to define a spatially varying element-based membrane thickness. Alternatively, you can click  to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
6. Specify the **Section Poisson's ratio** to define how the membrane thickness will change with deformation.
   - Toggle on **Use analysis default** to use the default value. In Abaqus/Standard the default value is 0.5, which will enforce incompressible behavior of the element. In Abaqus/Explicit the default is to base the change in thickness on the element material definition.
   - Toggle on **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant thickness, and a negative value will result in an increase in the thickness in response to tensile membrane strains.
7. Click  at the bottom of the membrane section editor to define rebar layers in the membrane section, as described in *Defining rebar layers*.
8. Click **OK** to save your changes and to close the membrane section editor.

Creating surface sections

Surface sections are used to define the section properties of surfaces in space that have no inherent stiffness and behave like membranes with zero thickness. For more information, see *Surface Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Shell** as the section **Category** and **Surface** as the section **Type**, and click **Continue**.

   The surface section editor appears.
4. To bring additional mass into the model, toggle on **Density** and enter the mass density per unit area.
5. Click to define rebar layers in the surface section, as described in *Defining rebar layers*.
6. Click **OK** to save your changes and to close the surface section editor.

Creating general shell stiffness sections

Shell section behavior is defined in terms of the response of the shell section to stretching, bending, shear, and torsion.

General shell stiffness sections allow you to specify shell section properties directly in terms of a stiffness matrix and thermal expansion response. These data completely define the shell section's mechanical response, so no material reference is needed as part of the section definition. General shell stiffness sections provide an efficient and flexible method for defining shell responses when you know the terms of the applicable stiffness and thermal stress matrices. For more information, see *Specifying the Equivalent Section Properties Directly for Conventional Shells*.

General shell stiffness sections cannot be used with variable thickness shells or continuum shells. Like other pre-integrated shell sections, general shell stiffness sections are not integrated through their thickness and do not provide for heat transfer. In an Abaqus/Standard analysis, stresses and strains are not available for output from general shell stiffness sections.

For more information, see *Shell Section Behavior*.

In this section:

- *Creating a general shell stiffness section*
- *Specifying dependencies for general shell stiffness sections*
- *Specifying advanced properties for general shell stiffness sections*

Creating a general shell stiffness section

You can create a general shell stiffness section.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Shell** as the section **Category** and **General shell stiffness** as the section **Type**, and click **Continue**.

   The general shell stiffness section editor appears.

4. On the **Stiffness** tabbed page, enter the symmetric half of the shell stiffness matrix in the data table.

   The first row contains matrix entries D_{11} through D_{16}, the second row contains matrix entries D_{22} through D_{26}, and so on; the last row contains matrix entry D_{66}. See *Entering tabular data*, for more information.

Specifying dependencies for general shell stiffness sections

The **Dependencies** tabbed page allows you to define thermal stresses on the shell section. You can also apply temperature-dependent and field variable–dependent scaling factors to the shell stiffness matrix and thermal stresses. For detailed information on the equations used to define shell stiffness dependencies, refer to *Specifying the Equivalent Section Properties Directly for Conventional Shells*.

On the **Dependencies** tabbed page:

1. If you are defining a thermal expansion coefficient that is a function of temperature, toggle on **Specify reference temperature** and enter the reference temperature, θ^0, in the field provided.
2. To define thermal stresses on the shell section, toggle on **Apply thermal stress** and enter the stress components in the table provided: ε_{11} (**Sigma11**), ε_{22} (**Sigma22**), γ_{12} (**Gamma12**), κ_{11} (**K11**), κ_{22} (**K22**), and κ_{12} (**K12**).
3. To define scaling factors that depend on temperature, toggle on **Use temperature-dependent data**.

   A column labeled **Temp** appears in the **Scaling factors** data table.
4. To define scaling factors that depend on field variables, click the arrows to the right of the **Number of field variables** to increase or decrease the number of field variables.

   Field variable columns appear in the **Scaling factors** data table.
5. In the **Scaling factors** data table, enter the **Scaling modulus** (*Y*) for the stiffness matrix and the **Thermal Expansion Coefficient** (α) for the thermal stresses.

Specifying advanced properties for general shell stiffness sections

You can specify advanced properties for general shell stiffness sections.

On the **Advanced** tabbed page:

1. Specify the **Section Poisson's ratio** to define the shell thickness behavior. Specifying the section Poisson's ratio causes the shell thickness to change as a function of membrane strains.
   - Toggle on **Use analysis default** to use the default value of 0.5, which will enforce incompressible behavior of the element for membrane strains.
   - Toggle on **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.
2. Toggle on **Density**, and enter a value for the mass per unit surface area of the shell. The mass of the shell is determined from this density.
3. For most shell sections Abaqus will calculate the transverse shear stiffness values required in the element formulation. If desired, toggle on **Specify values** from the **Transverse Shear Stiffnesses** options to include nondefault transverse shear stiffness effects in the section definition, and enter values for K_{11}, the shear stiffness of the section in the first direction; K_{12}, the coupling term in the shear stiffness of the section; and K_{22}, the shear stiffness of the section in the second direction. If either value K_{11} or K_{22} is omitted or given as zero, the nonzero value will be used for both.
4. Click **OK** to save your changes and to close the general shell stiffness section editor.

Creating beam sections

Beam section behavior is defined in terms of the response of the beam section to stretching, bending, shear, and torsion.
For more information, see *Beam Section Behavior*.

When you create beam sections, you must choose a section integration method. You can choose to provide the section property data before the analysis (a general beam section) or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis.

In this section:

- *Specifying properties for beam sections integrated during analysis*
- *Specifying properties for general beam sections*

Specifying properties for beam sections integrated during analysis

Beam sections integrated during analysis allow the cross-sectional behavior to be calculated by numerical integration of the stress over the cross-section to define the beam's response as the analysis proceeds. The material behavior is evaluated independently at each point on the section. This type of beam section should be used when the section nonlinearity is caused only by nonlinear material response. For more information, see *Using a Beam Section Integrated during the Analysis to Define the Section Behavior*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Beam** as the section **Category** and **Beam** as the section **Type**, and click **Continue**.

   The beam section editor appears.
4. Choose **During analysis** as the **Section integration** method.
5. Select a profile for the beam section. If desired, click to create a profile; see *Creating profiles*, for more information.

   The **Profile shape** field is updated to reflect your choice.
6. On the **Basic** tabbed page:
   a. Select a **Material name** to be used with this beam section definition. If desired, click to create a material; see *Creating and editing materials*, for more information.
   b. Enter a value for the **Section Poisson's ratio** to provide uniform strain in the section due to strain of the beam axis (so that the cross-sectional area changes when the beam is stretched). This value must be between −1.0 and 0.5. A value of 0.5 will enforce incompressible behavior.
   c. Select a method for defining the **Temperature variation** through the section:
      - Choose **Linear by gradients** to indicate that the temperature at the cross-section origin and the temperature gradient or gradients through the section are specified. You can use the Load module to specify these temperatures.
      - Choose **Interpolated from temperature points** to indicate that the shape of the beam section profile determines the number and location of the temperature points. (For more information on temperature points, see *Beam Cross-Section Library*.) You can use the Load module to specify the temperature at each of these points.
7. On the **Stiffness** tabbed page, do the following:
   a. Select **Use consistent mass matrix formulation** to let Abaqus calculate the mass formulation for the beam section using the McCalley-Archer consistent mass matrix based on the cubic interpolation of deflections and quadratic interpolation of rotations. If you toggle off this option, Abaqus performs this calculation using a lumped mass formulation.
   b. Toggle on **Specify transverse shear** to include nondefault transverse shear stiffness effects in the section definition and specify the **Slenderness compensation**:
      - Select **Use analysis product default** to let Abaqus calculate the shear stiffness and the slenderness compensation factor from the elastic material definition for the beam section.

- Select **Value** to specify the transverse shear stiffness effects directly.
  - Enter a slenderness compensation factor in the field provided.
  - Enter values for the K_{23} and K_{13} shear stiffnesses of the section in the fields provided.

8. On the **Fluid Inertia** tabbed page, toggle on **Specify fluid inertia effects** to simulate the inertial effects of the beam being submerged in a fluid. For details, see *Additional Inertia due to Immersion in Fluid*.
   a. Specify whether the beam is **Fully submerged** or **Half submerged** in the fluid. If you select **Half submerged**, the added inertia per unit length is reduced by a factor of one-half.
   b. Specify the **Fluid density**.
   c. Enter the effective radius of the wetted cross-section in the **Section radius** field.
   d. Specify C_A, the added mass coefficient for **Lateral motions of the beam**.
   e. Specify C_{A-E}, the added mass coefficient for **Motions along beam axis**.
   f. If the beam cross-section origin is different than the centroid of the wetted cross-section, specify the centroid's **X** and **Y** coordinates relative to the cross-section origin.
9. Click **OK** to save your changes and to close the beam section editor.

Specifying properties for general beam sections

In general beam sections the cross-section properties are calculated only once, during preprocessing. All section computations during the analysis are performed in terms of the precomputed values. No material definition is needed for a general beam section. Use this type of beam section when the response of the beam is linear or when it is nonlinear and the nonlinearity arises from more than just material nonlinearity, such as in cases when section collapse occurs. For more information, see *Using a General Beam Section to Define the Section Behavior*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Beam** as the section **Category** and **Beam** as the section **Type**, and click **Continue**.

   The beam section editor appears.
4. Select **Before analysis** as the **Section integration** method.
5. From the **Beam shape along length** options, do one of the following:

   - To maintain the same cross-sectional profile along the entire length of the beam, select **Constant** and select a beam profile from the **Beam Shape** options. If desired, click  to create a profile; see *Creating profiles*, for more information.
   - To define different cross-sectional profiles at each end of the beam, select **Tapered** and select starting and ending profiles from the **Beam Start** and **Beam End** options, respectively. If desired, click  to create a profile; see *Creating profiles*, for more information. The starting and ending profiles must be the same shape.

     Tapered beams are supported for Abaqus/Standard analyses only. Abaqus/CAE scales beam profiles linearly between the starting and ending profiles.

6. From the **Material definition type** options, do one of the following:

   - To specify the material definition of the section using a predefined material name, select **Material name**.
   - To specify the material definition of the section by creating a new material, select **Table**.

7. If you selected a generalized profile, you can offset the beam section from its node by specifying how far and in which direction along the cross-section axes to move the section centroid and/or shear center. Enter the local $\mathbf{x}_1$- and $\mathbf{x}_2$-coordinates for the **Centroid** and/or the **Shear Center** as desired.
8. On the **Basic** tabbed page:

   - If **Table** is selected as the **Material definition type**, do the following:

   a. To define the section thermal expansion coefficient, toggle on **Use thermal expansion data**.

      A column labeled **Thermal Expansion** appears in the data table.

   b. To define section data that depend on temperature, toggle on **Use temperature-dependent data**.

      A column labeled **Temperature** appears in the data table.

c. To define section data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

Field variable columns appear in the data table.

d. Enter values for the section **Young's Modulus** and torsional **Shear Modulus** in the data table.

e. Enter a value for the **Section Poisson's ratio** to provide uniform strain in the section due to strain of the beam axis (so that the cross-sectional area changes when the beam is stretched). This value must be between −1.0 and 0.5. A value of 0.5 will enforce incompressible behavior. The default value is 0.

f. To define a density for the beam section, toggle on **Specify section material density** and enter a value in the field provided. This value is required in an Abaqus/Explicit analysis. In an Abaqus/Standard analysis it is needed only when the mass is required, such as in dynamic analysis or gravity loading.

g. If the thermal expansion coefficient is temperature dependent, toggle on **Specify reference temperature** and enter a value for the thermal expansion in the field provided.

- If **Material name** is selected as the **Material definition type**, do the following:

a. Select a **Material name** to be used with this beam section definition.

b. Enter a value for the **Section Poisson's ratio** to provide uniform strain in the section due to strain of the beam axis (so that the cross-sectional area changes when the beam is stretched). This value must be between −1.0 and 0.5. A value of 0.5 will enforce incompressible behavior.

9. On the **Damping** tabbed page, specify damping properties to include mass and stiffness proportional damping in the dynamic response of the section:

a. Enter a value in the **Alpha** field for the α_R factor to create mass proportional damping in direct-integration dynamics. This value is ignored in modal dynamics.

b. Enter a value in the **Beta** field for the β_R factor to create stiffness proportional damping in direct-integration dynamics. This value is ignored in modal dynamics.

c. Enter a value in the **Composite** field for the fraction of critical damping to be used in calculating composite damping factors for the modes (for use in modal dynamics). This value is applicable only to Abaqus/Standard analysis and is ignored in direct-integration dynamics.

10. On the **Stiffness** tabbed page, do the following:

a. Select **Use consistent mass matrix formulation** to let Abaqus calculate the mass formulation for the beam section using the McCalley-Archer consistent mass matrix based on the cubic interpolation of deflections and quadratic interpolation of rotations. If you toggle off this option, Abaqus performs this calculation using a lumped mass formulation.

b. Toggle on **Specify transverse shear** to include nondefault transverse shear stiffness effects in the section definition and specify the **Slenderness compensation**:

- Select **Use analysis product default** to let Abaqus calculate the shear stiffness and the slenderness compensation factor from the elastic material definition for the beam section.
- Select **Value** to specify the transverse shear stiffness effects directly.
  - Enter a slenderness compensation factor in the field provided.
  - Enter values for the K_{23} and K_{13} shear stiffnesses of the section in the fields provided.

11. On the **Fluid Inertia** tabbed page, toggle on **Specify fluid inertia effects** to simulate the inertial effects of the beam being submerged in a fluid. For details, see *Additional Inertia due to Immersion in Fluid*.

a. Specify whether the beam is **Fully submerged** or **Half submerged** in the fluid. If you select **Half submerged**, the added inertia per unit length is reduced by a factor of one-half.

b. Specify the **Fluid density**.

c. Enter the effective radius of the wetted cross-section in the **Section radius** field.

d. Specify C_A, the added mass coefficient for **Lateral motions of the beam**.

e. Specify C_{A-E}, the added mass coefficient for **Motions along beam axis**.

f. If the beam cross-section origin is different than the centroid of the wetted cross-section, specify the centroid's **X** and **Y** coordinates relative to the cross-section origin.

12. On the **Output Points** tabbed page, you can locate points in the beam section for which stress and strain output are required. Enter the local x_1- and x_2-positions of as many section points as needed.

13. Click **OK** to save your changes and to close the beam section editor.

Creating truss sections

Truss sections are used to define the section properties of slender, rod-like structures in two and three dimensions that provide axial strength but no bending stiffness. For more information, see *Truss Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Beam** as the section **Category** and **Truss** as the section **Type**, and click **Continue**.

   The truss section editor appears.

4. Select a material for the truss section. If desired, click to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the **Cross-sectional area**.
6. Click **OK** to save your changes and to close the truss section editor.

Creating gasket sections

Gasket sections are used to define the section properties of thin sealing components that are positioned between structural components. For more information, see *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*. *Gaskets*," describes the entire modeling process for an analysis involving gaskets.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Other** as the section **Category** and **Gasket** as the section **Type**, and click **Continue**.

   The gasket section editor appears.

4. Select a material for the gasket section. If desired, click to create a material; see *Creating or editing a material*, for more information.
5. Specify the **Stabilization stiffness** that is used to stabilize gasket elements that are not supported at all nodes, such as those that extend outside neighboring components.
   - Toggle on **Use default** to use the default value of 10^{-9} times the initial compressive stiffness in the thickness direction. This default value is usually appropriate.
   - Toggle on **Specify**, and enter a value for the desired stabilization stiffness in units of stress (FL^{-2}).
6. Specify the **Initial thickness** of the gasket.
   - Toggle on **Use nodal coordinates** to obtain the initial gasket thickness from the nodal coordinates.
   - Toggle on **Specify**, and enter a value for the initial gasket thickness.
7. Enter a value for the **Initial gap**, which is the gasket closure that is needed to generate a pressure.
8. Enter a value for the **Initial void**, which is the inter-layer spaces in a gasket.
9. Enter a value for the **Cross-sectional area, width, or out-of-plane thickness**, depending on the gasket element type. For two- and three-dimensional link elements the cross-sectional area of the element should be given. For axisymmetric link elements and three-dimensional line elements the width of the element should be given. For general two-dimensional elements the out-of-plane thickness is required. For three-dimensional area elements no additional quantities are needed to define the element geometry.
10. Click **OK** to save your changes and to close the gasket section editor.

Creating cohesive sections

Cohesive sections are used to define the section properties of adhesive layers at the interface between two bonded parts. For more information, see *About Cohesive Elements*. *Adhesive joints and bonded interfaces*," describes the entire modeling process for an analysis involving cohesive sections.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Other** as the section **Category** and **Cohesive** as the section **Type**, and click **Continue**.

   The cohesive section editor appears.

4. Select a material for the cohesive section. If desired, click  to create a material; see *Creating or editing a material*, for more information.
5. Select a **Response** for defining the constitutive behavior of the cohesive section:

   - Select **Traction Separation** if the response is defined directly in terms of traction and separation. Use this option to model a negligibly thin layer of adhesive (debonding).
   - Select **Continuum** to model a strain state involving one direct (opening strain) and two transverse shear components. Use this option to model a finite thickness adhesive layer.
   - Select **Gasket** to specify that the stress state is uniaxial.

6. Specify the **Initial thickness** of the cohesive section.

   - Toggle on **Use analysis default** to use the analysis default value for a **Traction Separation** or **Continuum** response. No default value is available for the **Gasket** response.
   - Toggle on **Use nodal coordinates** to obtain the initial thickness from the nodal coordinates.
   - Toggle on **Specify**, and enter a value for the initial thickness.

7. Enter a value for the section **Out-of-plane thickness**. If the section will be used with a two-dimensional region, you must specify the section thickness. Abaqus/CAE ignores the thickness information if it is not needed for the region type.
8. Click **OK** to save your changes and to close the cohesive section editor.

Creating acoustic infinite sections

Acoustic infinite sections are used to define the section properties of two-dimensional, three-dimensional, and axisymmetric regions that model an acoustic medium undergoing small pressure changes. You use acoustic infinite sections when you want to improve the accuracy of analyses involving exterior domains. You must create a reference point to indicate the reference node required by acoustic infinite elements. For more information, see *Infinite Elements*.

Specify acoustic infinite section properties

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Other** as the section **Category** and **Acoustic infinite** as the section **Type**, and click **Continue**.

   The section editor appears.

4. Select an acoustic medium material for the acoustic infinite section. If desired, click  to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the section **Plane stress/strain thickness**. If the section will be used with a two-dimensional region, you must specify the section thickness. Abaqus/CAE ignores the thickness information if it is not needed for the region type.
6. To define the number of ninth-order polynomials that will be used to resolve the variation of the acoustic field in the infinite direction, click the arrow to the right of the **Order** field to decrease the number of polynomials that will be used (applies only to Abaqus/Explicit analyses). The default value is 10, which is the value always used in Abaqus/Standard.
7. Click **OK** to save your changes and to close the section editor.

Create a reference point for use with acoustic infinite elements

You must create a reference point to be used to determine the “radius” and “node ray” at each node of acoustic infinite elements. You create a reference point associated with a part in the Part module or in the Property module using the procedure described in *Creating a reference point*. For more information, see *Defining the Reference Point and Thickness for Acoustic Infinite Elements*.

Creating acoustic interface sections

Acoustic interface sections are used to define the section properties of two-dimensional, three-dimensional, and axisymmetric regions that model an acoustic medium undergoing small pressure changes. You use acoustic interface sections when you want to couple the acoustic medium to a structural model. For more information, see *Acoustic Interface Elements*.

1. From the main menu bar, select **Section->Create**.

   A **Create Section** dialog box appears.

   Tip: You can also click **Create** in the **Section Manager** or select the create section tool  in the Property module toolbox.

2. Enter a section name. For more information on naming objects, see *Using basic dialog box components*.
3. Select **Other** as the section **Category** and **Acoustic interface** as the section **Type**, and click **Continue**.

   The section editor appears.

4. Select an acoustic medium material for the acoustic interface section. If desired, click  to create a material; see *Creating or editing a material*, for more information.
5. Enter a value for the section **Plane stress/strain thickness**. If the section will be used with a two-dimensional region, you must specify the section thickness. Abaqus/CAE ignores the thickness information if it is not needed for the region type.
6. Click **OK** to save your changes and to close the section editor.

Defining rebar layers

When you create homogeneous shell sections, composite shell sections, membrane sections, or surface sections, you can define one or more layers of reinforcement (rebar) by using the **Rebar Layers** option. You access the **Rebar Layers** dialog box from the section editor. For more information, see *Defining Reinforcement*.

1. From the **Options** field of the shell, membrane, or surface section editor, click .
   The **Rebar Layers** dialog box appears.
2. Specify the type of rebar geometry.
   - Choose **Constant** for a constant rebar spacing.
   - Choose **Angular** if the rebar spacing varies as a function of radial position in a cylindrical coordinate system.
   - Choose **Lift equation–based** if the rebar spacing and orientation are determined by the tire lift equation.
3. In the table, enter a row of data for each rebar layer.
   Each row should contain the following information:
   - The name of the rebar layer.
   - The name of the material forming the rebar layer. Click in the **Material** column, then click the arrow that appears to display the list of available materials, and select the material forming the rebar layer.
   - The cross-sectional area per bar.
   - The rebar spacing in the plane of the section. For angular rebar spacing, specify the spacing angle in degrees. For lift equation–based spacing, specify the spacing in the uncured geometry.
   - The angular orientation of the rebar (in degrees) relative to the 1-direction of the rebar reference orientation. For lift equation–based spacing, specify the angle in the uncured geometry.
     Alternatively, you can specify an orientation name, in which case Abaqus/CAE assumes a user-defined orientation. You must supply the user subroutine *ORIENT* that contains the definition of the user-defined orientation for the specified orientation name.
   - The position of the rebar in the shell thickness direction measured from the middle surface of the shell, positive in the direction of the positive normal to the shell (not applicable for membrane sections or surface sections).
   - The cord extension ratio, e, (applies only for lift equation–based spacing).
   - The radius, r_0, that describes the position of the rebar in the uncured geometry, measured with respect to the axis of rotation in a cylindrical coordinate system (applies only for lift equation–based spacing).
4. Click **OK** to return to the section editor.

Additional information

- *Creating sections*
- *Defining Reinforcement*

Creating profiles

To create a profile you must choose a profile type and enter all of the data necessary to define the profile in the **Edit Profile** dialog box.

In this section:

- *Choosing a profile type*
- *Defining a box profile*
- *Defining a pipe profile*
- *Defining a circular profile*
- *Defining a rectangular profile*
- *Defining a hexagonal profile*
- *Defining a trapezoidal profile*
- *Defining an I-shaped profile*
- *Defining an L-shaped profile*
- *Defining a T-shaped profile*
- *Defining an arbitrary profile*
- *Defining a generalized profile*
- *Defining a channel profile*
- *Defining a hat profile*

Choosing a profile type

The **Create Profile** dialog box allows you to specify which type of profile you want to define. You can define a shape-based profile by providing the geometric data from which Abaqus can calculate the engineering properties of the section. Alternatively, you can define a generalized profile by providing the engineering properties of the section directly.
For more information, see *Beam Section Behavior*.

After you have assigned the beam section and beam orientation to the part, you can use the part display options to view an idealized representation of the shape-based or generalized beam profile. Displaying beam profiles is useful for checking that the correct profile is assigned to a particular region and that the assigned beam orientation results in the expected orientation of the profile. For more information, see *Controlling beam profile display*.

1. From the main menu bar, select **Profile->Create**.

   The **Create Profile** dialog box appears.

   Tip: You can also click **Create** in the **Profile Manager** or select the create profile tool  in the Property module toolbox.

2. Enter a profile name. For more information on naming objects, see *Using basic dialog box components*.
3. Select a profile shape, and click **Continue**.

   The **Edit Profile** dialog box for the profile shape you have chosen appears.
4. In the **Edit Profile** dialog box, enter the required profile data. See the following sections for details:

   - *Defining a box profile*
   - *Defining a pipe profile*
   - *Defining a circular profile*
   - *Defining a rectangular profile*
   - *Defining a hexagonal profile*
   - *Defining a trapezoidal profile*
   - *Defining an I-shaped profile*
   - *Defining an L-shaped profile*
   - *Defining a T-shaped profile*
   - *Defining an arbitrary profile*
   - *Defining a generalized profile*
   - *Defining a channel profile*
   - *Defining a hat profile*

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*
- *Using a General Beam Section to Define the Section Behavior*

Defining a box profile

Define a box profile by providing geometric data for a rectangular, hollow box.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **Width (a)** field, enter the length of the segments parallel to the local 1-axis.
3. In the **Height (b)** field, enter the length of the segments parallel to the local 2-axis.
4. Click the arrow to the right of the **Thickness** field, and indicate how you want to define the thickness of each segment:
   - Select **Uniform** if each of the four segments of the box have the same thickness. If you select this option, enter the segment thickness in the **Thickness** field below.
   - Select **Individual** to enter a value for each segment individually. If you select this option, enter the following:
     - In the **t1** field, enter the segment thickness labeled t_1 in the diagram.
     - In the **t2** field, enter the segment thickness labeled t_2 in the diagram.
     - In the **t3** field, enter the segment thickness labeled t_3 in the diagram.
     - In the **t4** field, enter the segment thickness labeled t_4 in the diagram.
5. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a pipe profile

Define a pipe profile by providing geometric data for a hollow circle and by selecting an integration scheme for a thin-walled pipe or a thick-walled pipe.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. From the **Integration scheme** options, specify a thin-walled pipe or a thick-walled pipe.
3. In the **Radius** field, enter the radius of the circle from the center of the pipe to the outside edge of the pipe wall.
4. In the **Thickness** field, enter the thickness of the pipe wall.
5. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a circular profile

Define a circular profile by providing geometric data for a solid circle.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **r** field, enter the radius of the circle.
3. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a rectangular profile

Define a rectangular profile by providing geometric data for a solid rectangle.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **a** field, enter the length of the rectangle edges parallel to the local 1-axis.
3. In the **b** field, enter the length of the rectangle edges parallel to the local 2-axis.
4. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a hexagonal profile

Define a hexagonal profile by providing geometric data for a hollow hexagon.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **r** field, enter the circumscribing radius.
3. In the **t** field, enter the wall thickness.
4. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a trapezoidal profile

Define a trapezoidal profile by providing geometric data for a solid trapezoid.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **a** field, enter the length of the lower edge of the trapezoid parallel to the local 1-axis.
3. In the **b** field, enter the height of the trapezoid parallel to the local 2-axis.
4. In the **c** field, enter the width of the top edge of the trapezoid parallel to the local 1-axis.
5. In the **d** field, enter the distance between the lower edge of the trapezoid and the local cross-section axis.
6. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining an I-shaped profile

Define an I-shaped profile by providing geometric data for an I-beam section.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **l** field, enter the distance between the lower edge of the bottom flange and the local cross-section axis.
3. In the **h** field, enter the height of the I-shape, from the bottom edge to the top edge.
4. In the **b1** field, enter the width of the lower flange parallel to the 1-axis.
5. In the **b2** field, enter the width of the upper flange parallel to the 1-axis.
6. In the **t1** field, enter the thickness of the lower flange.
7. In the **t2** field, enter the thickness of the upper flange.
8. In the **t3** field, enter the thickness of the segment joining the upper and lower flanges.
9. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining an L-shaped profile

Define an L-shaped profile by providing geometric data for an L-beam section.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **a** field, enter the length of the flange parallel to the local 1-axis.
3. In the **b** field, enter the length of the flange parallel to the local 2-axis.
4. In the **t1** field, enter the thickness of the flange parallel to the local 1-axis.
5. In the **t2** field, enter the thickness of the flange parallel to the local 2-axis.
6. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a T-shaped profile

Define a T-shaped profile by providing geometric data for a T-beam section.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **b** field, enter the length of the segment parallel to the 1-axis.
3. In the **h** field, enter the height of the T-shape, from the bottom edge to the top edge.
4. In the **l** field, enter the distance between the lower edge and the local cross-section axis.
5. In the **tf** field, enter the thickness of the segment parallel to the local 1-axis.
6. In the **tw** field, enter the thickness of the segment parallel to the local 2-axis.
7. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining an arbitrary profile

You can create an arbitrary profile to model the shape of a simple, thin-walled, open or closed section. You define the profile by entering a series of points which are linked by straight line segments.
For more information, see *Arbitrary, Thin-Walled, Open and Closed Sections*.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the data table, enter the local coordinates for **Point 1**.

   For detailed information on how to enter data, see *Entering tabular data*.
3. Enter the local coordinates for **Point 2**. The line joining **Point 1** and **Point 2** is labeled **Segment 1-2** in the data table.
4. Enter a **Thickness** for **Segment 1-2**.
5. Enter the local coordinates for **Point 3**. The line between **Point 2** and **Point 3** is labeled **Segment 2-3** in the data table.
6. Enter a **Thickness** for **Segment 2-3**.

   Note:

   For each individual segment of an arbitrary section there is no bending stiffness about the line joining the end points of the segment. Thus, an arbitrary section must include at least two segments.
7. If necessary, enter local coordinates for additional points, and provide thicknesses for the resulting segments.
8. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a generalized profile

When you create a generalized profile, you provide the engineering properties of the section directly in the **Edit Profile** dialog box.
For more information, see *Beam Section Behavior*.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **Area** field, enter the area of the profile shape.
3. In the **I11** field, enter the moment of inertia for bending about the 1-axis of the section.
4. In the **I12** field, enter the moment of inertia for cross-bending.
5. In the **I22** field, enter the moment of inertia for bending about the 2-axis of the section.
6. In the **J** field, enter the torsional constant.
7. If the profile describes an open-section beam, enter the following:
   - In the **Gamma O** field, enter the sectoral moment, Γ_0.
   - In the **Gamma W** field, enter the warping constant, Γ_W.
8. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Using a General Beam Section to Define the Section Behavior*

Defining a channel profile

Define a channel profile by providing geometric data for a channel section.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **l** field, enter the distance between the lower edge of the bottom flange and the local cross-section 1-axis.
3. In the **h** field, enter the height of the channel section, from the bottom edge to the top edge.
4. In the **b1** field, enter the width of the lower flange of the channel section.
5. In the **b2** field, enter the width of the upper flange of the channel section.
6. In the **t1** field, enter the thickness of the lower flange of the channel section.
7. In the **t2** field, enter the thickness of the upper flange of the channel section.
8. In the **t3** field, enter the thickness of the segment joining the upper and lower flanges of the channel section.
9. In the **o** field, enter the distance between the left edge of the web section and the local cross-section 2-axis of the channel section.
10. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Defining a hat profile

Define a hat profile by providing geometric data for a hat section.

1. Display the **Edit Profile** dialog box, as described in *Choosing a profile type*.
2. In the **l** field, enter the distance between the lower edge of the bottom flange and the local cross-section 1-axis.
3. In the **h** field, enter the height of the hat section, from the bottom edge to the top edge.
4. In the **b** field, enter the bottom width of the hat section.
5. In the **b1** field, enter the width of the upper flange of the hat section.
6. In the **b2** field, enter the width of the lower flange of the hat section.
7. In the **t1** field, enter the thickness of the upper flange of the hat section.
8. In the **t2** field, enter the thickness of the lower flange of the hat section.
9. In the **t3** field, enter the thickness of the segment joining the upper and lower flanges.
10. Click **OK** to save the profile and to close the **Edit Profile** dialog box.

Additional information

- *Defining profiles*
- *Defining sections*
- *Beam Cross-Section Library*

Creating and editing composite layups

This section describes how you create a composite layup.

In this section:

- *Using the ply table when defining a composite layup*
- *Creating conventional shell composite layups*
- *Creating continuum shell composite layups*
- *Creating solid composite layups*

Using the ply table when defining a composite layup

When you are creating a composite layup (for a shell, continuum shell, or solid), you use a row in the ply table to enter information about a ply in the layup.
The number of rows determines the number of plies in the layup, and the first row of the ply table corresponds with the ply on the bottom of the layup.

Each row of the ply table asks for the following:

- Whether the ply is activated.
- The name of the ply.
- The region to which the ply is assigned.
- The name of the material used in this ply.
- The ply thickness.
- The coordinate system specifying the reference orientation of the ply.
- A rotation angle relative to the reference orientation.
- The number of integration points in the ply, if you are specifying properties for a composite layup integrated during the analysis.

For general information about working with tables in Abaqus/CAE, see *Entering tabular data*.

As you enter ply data in the table, the ply region is highlighted and the ply orientation is displayed on the model in the viewport. You can modify the display to turn off highlighting, to display the layup orientation, or to select the orientation vectors to display.

In cases where Abaqus/CAE attempts to draw coordinate systems for ply orientations in composite layups at singularity points of the system (i.e., points for which the user-selected coordinate system and the geometric normal from either geometry or elements cannot be resolved into a valid orientation for display purposes in Abaqus/CAE), the coordinate system will be drawn collapsed.

If the layup orientation is specified using a discrete field, no display is available for either the layup or ply orientation. For continuum shell elements, Abaqus/CAE does not project the displayed orientations onto the midplane surface. In both of these cases, you can perform a data check and view the output database in the Visualization module to verify the orientations. For more information, see *Performing a data check on a model*.

A number of tools are available for adding or removing plies from the ply table. Most tools appear as icons above the ply table. Certain tools can be accessed only by clicking mouse button 3 on the ply table and selecting an item from the contextual menu that appears. The following tools are available:

Edit

Edit the contents of the selected table cell. For example, you can select a region from the part in the viewport, choose a material, or enter the thickness. In most cases you can select multiple cells, and Abaqus/CAE applies your change to each of the selected cells. You can select a column header to select the entire column.

The **Edit** tool is available from the contextual menu. Alternatively, you can double-click on a cell in the **Region**, **Material**, or **CSYS** columns to edit that cell. You can also double-click on a column header to apply the same edit to all of the cells in the column.

Move Plies Up or Down

Move the selected plies up or down one row in the ply table. Click  to move the selected plies up one row, or click  to move the selected plies down one row.

Copy Plies Before or After

Create new plies by copying the selected plies before or after the selected plies. Click  to copy the selected plies before the selected plies, or click  to copy the selected plies after the selected plies.

Insert Row Before or After

Insert an empty row before or after each of the selected plies. Click  to insert an empty row before the selected plies, or click  to insert an empty row after the selected plies.

Delete Plies

Delete the selected plies. Click  to delete the selected plies.

Pattern Plies

Copy selected plies using a pattern. Click  to pattern the selected plies.

The location to which the patterned plies are copied can be either the first or last ply in the layup or the first or last ply from the selected plies. You can also specify the copy location by selecting the name of the ply below which to copy. If you select the name of the ply below which to copy, you can choose to insert the copied plies into the ply table or to overwrite any existing plies.

The pattern can be one of the following:

N-Copy

A specified number of copies of the selected plies.

Symmetry

A single copy of the selected plies. The plies that are copied are reordered so that the order of the original plies and the order of the copied plies is symmetric about the specified location.

Invert Plies

Invert the order of two or more selected adjacent plies. Click  to invert the order of the selected plies.

Read From File

Read ply data from an ASCII file. The data can be delimited with spaces, tabs, or commas. You can specify into which row and column of the ply table Abaqus/CAE should start to read the data. The first column of the table is the state of the suppression (true or false).

The **Read From File** tool is available only from the contextual menu.



Note:

The data in the **Region** column of the file refers to a named set in your part. The name of the region is case-sensitive. Therefore, the name of the region that you use in the file must be identical to the name of an existing set in your part. The same is also true for the data in the **CSYS** column—the name of the coordinate system that you use in the file must be identical to the name of an existing datum coordinate system in your part.

Write To File

Write the entire ply table to a specified file. Abaqus/CAE uses commas to delimit the data.

The **Write To File** tool is available only from the contextual menu.

The options that are available are context sensitive; for example, the  icon and the **Move Plies Up** option in the contextual menu do not appear if you are editing the first row of the ply table.

Additional information

- *Specifying the display of selected plies of a conventional shell composite layup*
- *Specifying the display of selected plies of a continuum shell composite layup*
- *Specifying the display of selected plies of a solid composite layup*

Creating conventional shell composite layups

This section describes how to create a conventional shell composite layup.

Conventional shell composite layups are composed of plies made of different materials in different orientations. A layup can contain a different number of plies in different regions. For more information, see *Composite layups*. Abaqus models a shell composite layup using conventional shell elements that discretize only the reference surface of each ply. Shell section behavior is defined in terms of the response of the shell section to stretching, bending, twist, and transverse shear. For more information, see *Shell Section Behavior*.

When you create conventional shell composite layups, you must choose a section integration method. You can choose to provide the section property data before the analysis (a pre-integrated shell section) or to have Abaqus calculate (integrate) the cross-sectional behavior from section integration points during the analysis.

Conventional shell composite layups integrated during analysis allow the cross-sectional behavior to be calculated by numerical integration through the shell thickness, thus providing complete generality in material modeling. Any number of material points can be defined through the thickness, and the material response can vary from point to point. You generally use shell elements integrated during analysis when the composite layup includes nonlinear material behavior. You must use shell elements integrated during analysis to model heat transfer. For more information, see *Using a Shell Section Integrated during the Analysis to Define the Section Behavior*.

Linear moment-bending and force-membrane strain relationships can be defined using pre-integrated composite layups. In this case all calculations are done in terms of section forces and moments. The section properties are specified by an elastic material; optionally, you can also apply an idealization based on assumptions about the expected behavior or makeup of the layup. You should use pre-integrated composite layups if the response of the layup is linear elastic and its behavior is not dependent on changes in temperature or predefined field variables. For more information, see *Using a General Shell Section to Define the Section Behavior*.

After you have created a conventional shell composite layup, you can use a ply stack plot to view a graphical representation of a core sample through a region of the layup. For more information, see *Viewing a ply stack plot*.

In this section:

- *Creating conventional shell composite layups*
- *Specifying the plies of a conventional shell composite layup*
- *Specifying the offset of a conventional shell composite layup*
- *Specifying the shell parameters of a conventional shell composite layup*
- *Specifying the display of selected plies of a conventional shell composite layup*

Creating conventional shell composite layups

You can define a conventional shell composite layup composed of plies made of one or more materials.
For each ply of the layup, you specify the name, material, thickness, orientation, and the number of integration points.
In addition, you select the region to which the ply is assigned. For more information, see *Shell Section Behavior*.

1. From the main menu bar, select **Composite->Create**.

   A **Create Composite Layup** dialog box appears.

   Tip: You can also click **Create** in the **Composite Layup Manager** or select the create composite layup tool in the Property module toolbox.

2. Enter a composite layup name. Abaqus/CAE displays this name in a ply stack plot. For more information on naming objects, see *Using basic dialog box components*.
3. Specify the initial ply count. When the composite layup editor appears, it will contain a row for each ply; however, you can use the editor to subsequently add or delete plies. The first row of the ply table corresponds to the bottom ply of the layup.
4. Select **Conventional Shell** as the **Element Type**, and click **Continue**.

   The composite layup editor appears.
5. Enter a description of the layup. Abaqus/CAE displays this description in the composite layup manager.
6. Do one of the following to specify the layup orientation:

   - Select **Part global** to use the part's coordinate system, and choose the axis that represents the **Normal direction**.
   - Select **Coordinate system** to select an existing coordinate system (or create a new coordinate system and select it), and do the following:

     1. Choose the axis that represents the **Normal direction**.
     2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select an existing scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

   - Select **Discrete** to define a discrete orientation, and do the following:

     1. Click .
     2. In the **Edit Discrete Orientation** dialog box that appears, define the normal axis and primary axis using the procedure described in *Using discrete orientations for material orientations and composite layup orientations*.
     3. Specify an additional rotation. The orientation is rotated through this angle about the selected normal axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

- Select **User-defined** to define the orientation in user subroutine *ORIENT*. This option is valid only for Abaqus/Standard analyses. See the following sections for more information:
  - *Specifying general job settings*
  - *ORIENT*
- Select the name of an orientation discrete field to specify a coordinate system that is varying spatially across the layup. You can also create a new discrete field by clicking  to the right of the **Definition** field. For more information, see *The Discrete Field toolset*. After selecting the discrete field, you must do the following:
  1. Choose the axis that represents the **Normal direction**.
  2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup.

The layup orientation is the reference orientation for any ply that uses the default orientation system (indicated by **<Layup>** in the **CSYS** column of the ply table). This orientation is used for material calculations and stress output in the individual plies, for the section forces output, and for the transverse shear stiffness. You can specify a different orientation for the individual plies of a continuum shell composite layup by specifying a reference orientation and/or a rotation angle. For more information, see *Understanding composite layups and orientations*.

7. Choose one of the following **Section integration** methods:
   - Choose **During analysis** to specify properties for a shell composite layup integrated during the analysis.
   - Choose **Before analysis** to specify properties for a pre-integrated shell composite layup.
8. If you are specifying properties for a composite layup integrated before the analysis, specify the **Idealization** to apply to the shell based on assumptions about the expected behavior or makeup of the layup. For more information, see *Idealizing the Section Response*.
   - Select **No idealization** to account for the complete stiffness of the shell as determined by the material assignments and ply composition.
   - Select **Smear all layers** if you do not know the exact stacking sequence for the plies in the composite layup. Contributions from each specified ply are smeared across the entire thickness of the layup, resulting in a general response independent of the stacking sequence.
   - Select **Membrane only** if the predominant response of the shell will be in-plane stretching; bending stiffness terms are eliminated from the shell stiffness calculations.
   - Select **Bending only** if the predominant response of the shell will be pure bending; membrane stiffness terms are eliminated from the shell stiffness calculations.
9. If you are specifying properties for a composite layup integrated during the analysis, select the **Thickness integration rule**.
   - Choose **Simpson** to use Simpson's rule for the shell section integration.
   - Choose **Gauss** to use Gauss quadrature for the shell section integration.

   See *Defining the Shell Section Integration*, for more information.
10. When you have finished defining the shell composite layup, click **OK** to save your changes and to close the editor.

Specifying the plies of a conventional shell composite layup

A composite layup is composed of a series of plies. You select the region to which a ply is assigned, and you specify the name, material, thickness, orientation, and the number of integration points of each ply.
You must specify ply names that are unique throughout the entire model to ensure the correct display of ply-based results. Use the icons above the ply table or click mouse button 3 on the ply table to see a menu that allows you to edit the contents of the table cell and to manipulate the data in the table; for example, you can add and delete plies, pattern plies, and invert plies. You can also read data into the table from a file or write data from the table into a file. For more information, see *Using the ply table when defining a composite layup*.

1. From the **Composite Layup** editor, click the **Plies** tab.
2. If the plies in the composite layup are symmetric about a central core, toggle on **Make calculated sections symmetric**. Enter the plies in the ply table, starting with the bottom ply in the first row and ending with the central ply. During the analysis Abaqus appends plies to the layup definition by repeating the entered plies (including the central ply) in the reverse order to the top of the layup. Each generated ply is labeled in ply stack plots and the output database by adding `Sym_` to the beginning of the repeated ply's original name.

   This option cannot be used if the thickness or rotation angle for any of the plies in the layup is defined using a discrete field.
3. For each ply, enter the following data in the ply table:

Ply Name

The name of the ply. Abaqus/CAE displays this name when you are viewing the composite layup in the Visualization module and in a ply stack plot.

Region

Select the region to which the ply is assigned. You can choose faces from the viewport, or you can select a set that refers to a face. If you are displaying a meshed dependent part, you can choose shell elements from the viewport or select an element set. To choose elements from the viewport, you must display the native mesh, and you must use the Selection toolbar to enable the selection of **2D Elements**. You can also select shell elements from a mesh part. For more information, see *Displaying a native mesh*, and *Filtering your selection based on the type of object*.

Material

The name of the material for this ply. Click mouse button 3, select **Edit Material** from the menu that appears, and do either of the following:

- Select the desired material from the list of available materials.
- Click  to create a new material.

Thickness

The ply thickness. Enter a uniform thickness directly in the table, or click mouse button 3 and select **Edit Thickness** to do the following:

- Choose **Specify Value** to enter a uniform thickness for the ply.

- Choose **Distribution** and select a scalar discrete field or analytical field to specify a thickness that varies spatially across the ply.

Coordinate system

To define the coordinate system that will be used as the basis for the reference orientation of the ply, do the following:

1. Click mouse button 3, and select **Edit CSYS** from the menu that appears.

   Note:

   If you choose **Edit Orientation** from the menu, you can define both the coordinate system and the rotation angle.

2. Select the base orientation. You can select the base layup orientation, or you can select a coordinate system. If you select a coordinate system, you must select the axis that defines the normal direction.

Rotation Angle

An additional rotation (counterclockwise about the normal direction) for the ply's reference orientation. Enter a uniform rotation directly in the table, or click mouse button 3 and select **Edit Rotation Angle** to do the following:

- Select a predefined angle (0, 45, 90, or –45 degrees) to define a uniform rotation.
- Select **Uniform**, and enter an **Angle** to define a uniform rotation.
- Select a scalar discrete field to define a rotation that varies spatially across the ply. You can also create a new discrete field by clicking .

Integration Points

The number of integration points, if you are specifying properties for a composite layup integrated during the analysis.

Specifying the offset of a conventional shell composite layup

In most cases you can use the midsurface of an element to indicate the reference surface. However, in some cases you need to define the reference surface as offset from the midsurface of the elements.
For example, a model imported from a CAD system might assume that the shell is located at the top or bottom surface of the elements. In addition, you can specify a shell offset that defines a more precise surface geometry for contact problems where shell thickness is important. The offset value is defined as a fraction of the total thickness measured from the midsurface to the reference surface.

Positive values of the offset are in the positive element normal direction. When the offset is set equal to 0.5, the top surface of the element is the reference surface. When the offset is set equal to –0.5, the bottom surface is the reference surface. The default offset is 0, which indicates that the middle surface of the element is the reference surface. *Figure 1* shows an offset to the top surface of the element.

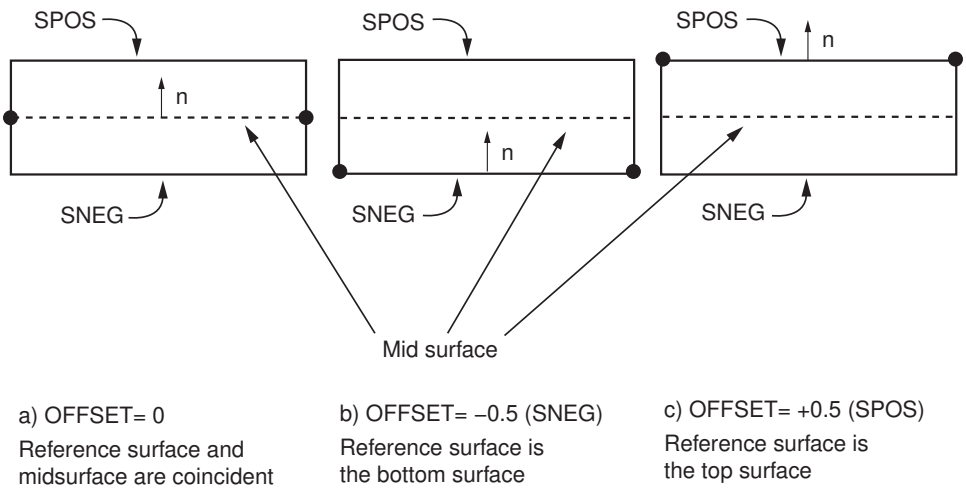


Figure 1: An offset to the top surface of the element.

For more information, see *Defining the Initial Geometry of Conventional Shell Elements*. You can use a discrete field to model elements with continuously varying offsets. For more information, see *The Discrete Field toolset*.

1. From the **Composite Layup** editor, click the **Offset** tab.
2. Do one of the following:
   - Choose the **Middle surface**, **Top surface**, or **Bottom surface** to represent the reference surface.
   - Choose **Specify offset ratio**, and enter a number.
   - Choose **Distribution**, and select an existing scalar discrete field that defines an offset that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an offset, are scalar discrete fields applied to elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.
   - Choose **From Geometry** to use the thickness of the geometric section.

Specifying the shell parameters of a conventional shell composite layup

1. From the **Composite Layup** editor, click the **Shell Parameters** tab.
2. Specify the **Shell thickness**.
   - Choose **Use section thickness** to use the thickness calculated from the individual ply thicknesses.
   - Choose **Element distribution**; and select either an analytical field, labeled with an (A), or an element-based discrete field, labeled with a (D), to define a spatially varying element-based shell thickness. Alternatively, you can click $f(x)$ to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
   - Choose **Nodal distribution**; and select either an analytical field, labeled with an (A), or a node-based discrete field, labeled with a (D), to define a spatially varying node-based shell thickness. Alternatively, you can click $f(x)$ to create a new analytical field or click  to create a new discrete field. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.
   - Choose **Determine overall thickness from geometry** to have Abaqus/CAE calculate the offset from the thickness defined on the geometry. You must choose **From Geometry** on the **Offset** tabbed page.
3. Specify the **Section Poisson's ratio** to define the shell thickness behavior. In conventional shell elements that permit finite membrane strains in large-deformation analysis, specifying the section Poisson's ratio causes the shell thickness to change as a function of membrane strains.
   - Choose **Use analysis default** to use the default value. In Abaqus/Standard the default value is 0.5, which will enforce incompressible behavior of the element for membrane strains. In Abaqus/Explicit the default is to base the change in thickness on the element material definition.
   - Choose **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.
4. If you are specifying properties for a composite layup integrated during the analysis, select a method for defining the **Temperature variation** through the section:
   - Choose **Linear through thickness** to indicate that the temperature at the reference surface and the temperature gradient or gradients through the ply are specified. You can use the Load module to specify these temperatures.
   - Choose **Piecewise linear over *n* values** to enter the number of temperature points (values) through the ply in the text field provided. You can use the Load module to specify the temperature at each of these points.
5. Toggle on **Density**, and enter a value for the density. The mass of the ply includes a contribution from the density in addition to any contribution from the selected material.
6. For most shell sections Abaqus calculates the transverse shear stiffness values required in the element formulation. If desired, toggle on **Specify values** from the **Transverse Shear Stiffnesses** options to include nondefault transverse shear stiffness effects in the section definition, and enter values for K_{11}, the shear stiffness of the section in the first direction; K_{12}, the coupling term in the shear stiffness of the section; and K_{22}, the shear stiffness of the section in the second direction. If either value K_{11} or K_{22} is omitted or given as zero, the nonzero value will be used for both. For more information, see *Defining the Transverse Shear Stiffness*.

Specifying the display of selected plies of a conventional shell composite layup

1. From the **Composite Layup** editor, click the **Display** tab.
2. Choose how Abaqus/CAE highlights the plies that you selected in the ply table:
   - Choose **On** to highlight the selected plies.
   - Choose **Off** to stop highlighting the selected plies. If you have a large number of plies in your model, you may want to turn off highlighting to increase performance.
3. Choose which orientation Abaqus/CAE displays on the selected plies:
   - Choose **Ply** to display the ply orientation on the selected plies.
   - Choose **Layup** to display the layup orientation on the selected plies.
   - Choose **None** to stop displaying an orientation.
4. Specify which orientation vectors you want to display on the selected plies. You can toggle the display of vectors in the 1-direction, the 2-direction, and the 3- (or normal) direction, as well as the reference direction that shows the 1-direction before rotation of the ply's material directions.

Creating continuum shell composite layups

This section describes how to create a continuum shell composite layup.

Abaqus models a continuum shell composite layup using continuum shell elements that fully discretize each ply but have a kinematic behavior that is based on shell theory. Continuum shell composite layups are composed of plies made of different materials in different orientations. A layup can contain a different number of plies in different regions. For more information, see *Composite layups*.

Continuum shell composite layups are expected to have a single element through their thickness, and that single element contains multiple plies that are defined in the ply table. If the region to which you assign your continuum shell composite layup contains multiple elements, each element will contain the plies defined in the ply table, and the analysis results will not be as expected.

You choose the stacking direction of the continuum shell elements in the layup, which allows Abaqus to model the through-thickness response more accurately. In addition, continuum shell composite layups take into account double-sided contact and thickness changes, which provides more accurate contact modeling than conventional shell composite layups. For more information, see *Shell Section Behavior*.

When you create continuum shell composite layups, you must choose a section integration method. You can choose to provide the section property data before the analysis (a pre-integrated continuum shell composite layup) or to have Abaqus calculate (integrate) the cross-sectional behavior from integration points during the analysis.

Continuum shell composite layups integrated during the analysis allow the cross-sectional behavior to be calculated by numerical integration through the continuum shell thickness, thus providing complete generality in material modeling. Any number of material points can be defined through the thickness, and the material response can vary from point to point. You generally use continuum shell elements integrated during analysis when the composite layup includes nonlinear material behavior. You must use continuum shell elements integrated during analysis to model heat transfer. For more information, see *Using a Shell Section Integrated during the Analysis to Define the Section Behavior*.

Linear moment-bending and force-membrane strain relationships can be defined using pre-integrated continuum shell composite layups. In this case all calculations are done in terms of section forces and moments. The section properties are specified by an elastic material; optionally, you can also apply an idealization based on assumptions about the expected behavior or makeup of the layup. You should use pre-integrated continuum shell composite layups if the response of the layup is linear elastic, and its behavior is not dependent on changes in temperature or predefined field variables. For more information, see *Using a General Shell Section to Define the Section Behavior*.

After you have created a continuum shell composite layup, you can use a ply stack plot to view a graphical representation of a core sample through a region of the layup. For more information, see *Viewing a ply stack plot*.

In this section:

- *Creating continuum shell composite layups*
- *Specifying the plies of a continuum shell composite layup*
- *Specifying the shell parameters of a continuum shell composite layup*
- *Specifying the display of selected plies of a continuum shell composite layup*

Creating continuum shell composite layups

You can define a continuum shell composite layup with a specified stacking direction that is composed of plies made of one or more materials. For each ply of the layup, you specify the name, material, relative thickness, orientation, and the number of integration points. In addition, you select the region to which the ply is assigned. For more information, see *Defining a Composite Shell Section*.

1. From the main menu bar, select **Composite->Create**.

   A **Create Composite Layup** dialog box appears.

   Tip: You can also click **Create** in the **Composite Layup Manager** or select the create composite layup tool in the Property module toolbox.

2. Enter a composite layup name. Abaqus/CAE displays this name in a ply stack plot. For more information on naming objects, see *Using basic dialog box components*.
3. Specify the initial ply count. When the composite layup editor appears, it will contain a row for each ply; however, you can use the editor to subsequently add or delete plies.
4. Select **Continuum Shell** as the **Element Type**, and click **Continue**.

   The composite layup editor appears.
5. Enter a description of the layup. Abaqus/CAE displays this description in the composite layup manager.
6. Do one of the following to specify the layup orientation:

   - Select **Part global** to use the part's coordinate system, and choose the axis that represents the **Normal direction**.
   - Select **Coordinate system** to select an existing coordinate system (or create a new coordinate system and select it), and do the following:

     1. Choose the axis that represents the **Normal direction**.
     2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select an existing scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

   - Select **Discrete** to define a discrete orientation, and do the following:

     1. Click .
     2. In the **Edit Discrete Orientation** dialog box that appears, define the normal axis and primary axis using the procedure described in *Using discrete orientations for material orientations and composite layup orientations*.
     3. Choose the axis that represents the **Normal direction**.
     4. Specify an additional rotation. The orientation is rotated through this angle about the selected normal axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to

elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

- Select **User-defined** to define the orientation in user subroutine *ORIENT*. This option is valid only for Abaqus/Standard analyses. See the following sections for more information:
  - *Specifying general job settings*
  - *ORIENT*
- Select the name of an orientation discrete field to specify a coordinate system that is varying spatially across the layup. You can also create a new discrete field by clicking  to the right of the **Definition** field. For more information, see *The Discrete Field toolset*. After selecting the discrete field, you must do the following:
  1. Choose the axis that represents the **Normal direction**.
  2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup.

The layup orientation is the reference orientation for any ply that uses the default orientation system (indicated by **<Layup>** in the **CSYS** column of the ply table). This orientation will be used for material calculations and stress output in the individual plies, for the section forces output, and for the transverse shear stiffness. You can specify a different orientation for the individual plies of a continuum shell composite layup by specifying a reference orientation and/or a rotation angle. For more information, see *Understanding composite layups and orientations*.

7. Choose one of the following to specify the stacking direction of the continuum shell elements:
   - **Element direction 1**
   - **Element direction 2**
   - **Element direction 3**
   - **Layup orientation**. The stacking direction is the normal to the layup orientation.

   You can use the Query toolset to determine the mesh stack orientation. However, the displayed orientations account for only the sweep path; they do not account for changes to the stacking direction as described above. For more information on the Query toolset, see *Using the Query toolset to query the model*. For more information on mesh stack directions, see *Defining the Stacking and Thickness Direction*.
8. Choose one of the following **Section integration** methods:
   - Choose **During analysis** to specify properties for a continuum shell composite layup integrated during the analysis.
   - Choose **Before analysis** to specify properties for a pre-integrated continuum shell composite layup.
9. If you are specifying properties for a composite layup integrated before the analysis, specify the **Idealization** to apply to the shell based on assumptions about the expected behavior or makeup of the layup. For more information, see *Idealizing the Section Response*.
   - Select **No idealization** to account for the complete stiffness of the shell as determined by the material assignments and ply composition.

- Select **Smeared properties** if you do not know the exact stacking sequence for the plies in the composite layup. Contributions from each specified ply are smeared across the entire thickness of the layup, resulting in a general response independent of the stacking sequence.
- Select **Membrane only** if the predominant response of the shell will be in-plane stretching; bending stiffness terms are eliminated from the shell stiffness calculations.
- Select **Bending only** if the predominant response of the shell will be pure bending; membrane stiffness terms are eliminated from the shell stiffness calculations.

10. If you are specifying properties for a composite layup integrated during the analysis, select the **Thickness integration rule**.

- Choose **Simpson** to use Simpson's rule for the shell section integration.
- Choose **Gauss** to use Gauss quadrature for the shell section integration.

See *Defining the Shell Section Integration*, for more information.

11. When you have finished defining the continuum shell composite layup, click **OK** to save your changes and to close the editor.

Specifying the plies of a continuum shell composite layup

A composite layup is composed of a series of plies. You select the region to which a ply is assigned, and you specify the name, material, relative thickness, orientation, and the number of integration points of each ply. You must specify ply names that are unique throughout the entire model to ensure the correct display of ply-based results. Use the icons above the ply table or click mouse button 3 on the ply table to see a menu that allows you to edit the contents of the table cell and to manipulate the data in the table; for example, you can add and delete plies, pattern plies, and invert plies. You can also read data into the table from a file or write data from the table into a file. For more information, see *Using the ply table when defining a composite layup*.

1. From the **Composite Layup** editor, click the **Plies** tab.
2. If the plies in the composite layup are symmetric about a central core, toggle on **Make calculated sections symmetric**. Enter the plies in the ply table, starting with the bottom ply in the first row and ending with the central ply. During the analysis Abaqus appends plies to the layup definition by repeating the entered plies (including the central ply) in the reverse order to the top of the layup. Each generated ply is labeled in ply stack plots and the output database by adding `Sym_` to the beginning of the repeated ply's original name.

   This option cannot be used if the rotation angle for any of the plies in the layup is defined using a discrete field.
3. For each ply, enter the following data in the ply table:

Ply Name

The name of the ply. Abaqus/CAE displays this name when you are viewing the composite layup in the Visualization module and in a ply stack plot.

Region

Select the region to which the ply is assigned. You can choose cells from the viewport, or you can select a set that refers to a cell. If you are displaying a meshed dependent part, you can choose solid elements from the viewport or select an element set. To choose elements from the viewport, you must display the native mesh, and you must use the Selection toolbar to enable the selection of **3D Elements** from the viewport. You can also select solid elements from a mesh part. For more information, see *Displaying a native mesh*, and *Filtering your selection based on the type of object*.

Material

The name of the material for this ply. Click mouse button 3, select **Edit Material** from the menu that appears, and do either of the following:

- Select the desired material from the list of available materials.
- Click  to create a new material.

Element Relative Thickness

The relative ply thickness within each element.

For continuum shell elements Abaqus determines the overall section thickness from the element geometry, which may vary from element to element where the section is defined. Hence, the thickness values that you specify for each ply are relative to the thickness of each element. The

actual thickness of a ply is the element thickness times the fraction of the total thickness that is accounted for by each ply. You do not have to use physical units to specify the thickness ratios for the plies, and the sum of the ply relative thicknesses does not have to add to one. For more information, see *Creating a composite layup*.

Coordinate system

To define the coordinate system that will be used as the basis for the reference orientation of the ply, do the following:

1. Click mouse button 3, and select **Edit CSYS** from the menu that appears.

   Note:

   If you choose **Edit Orientation** from the menu, you can define both the coordinate system and the rotation angle.

2. Select the base orientation. You can select the base layup orientation, or you can select a coordinate system. If you select a coordinate system, you must select the axis that defines the normal direction.

Rotation Angle

An additional rotation (counterclockwise about the normal direction) for the ply's reference orientation. Enter a uniform rotation directly in the table, or click mouse button 3 and select **Edit Rotation Angle** to do the following:

- Select a predefined angle (0, 45, 90, or –45 degrees) to define a uniform rotation.
- Select **Uniform**, and enter an **Angle** to define a uniform rotation.
- Select a scalar discrete field to define a rotation that varies spatially across the ply. You can also create a new discrete field by clicking .

Integration Points

The number of integration points, if you are specifying properties for a composite layup integrated during the analysis.

Specifying the shell parameters of a continuum shell composite layup

You can specify the shell parameters of a continuum shell composite layup.

1. From the **Composite Layup** editor, click the **Shell Parameters** tab.
2. Specify the **Section Poisson's ratio** to define the shell thickness behavior:
   - Toggle on **Use analysis default** to use the default value. In Abaqus/Standard the default value is 0.5, which will enforce incompressible behavior of the element for membrane strains. In Abaqus/Explicit the default is to base the change in thickness on the element material definition.
   - Toggle on **Specify value**, and enter a value for the Poisson's ratio. This value must be between −1.0 and 0.5. A value of 0.0 will enforce constant shell thickness, and a negative value will result in an increase in the shell thickness in response to tensile membrane strains.
3. Toggle on **Thickness modulus**, and enter a value for the thickness modulus. If you do not enter a value, Abaqus assumes the effective thickness modulus is twice the initial in-plane shear modulus based on the material definition.
4. If you are specifying properties for a composite layup integrated during the analysis, select a method for defining the **Temperature variation** through the section:
   - Choose **Linear through thickness** to indicate that the temperature at the reference surface and the temperature gradient or gradients through the ply are specified. You can use the Load module to specify these temperatures.
   - Choose **Piecewise linear over *n* values** to enter the number of temperature points (values) through the ply in the text field provided. You can use the Load module to specify the temperature at each of these points.
5. Toggle on **Density**, and enter a value for the density. The mass of the ply includes a contribution from the density in addition to any contribution from the selected material.
6. For most continuum shell composite layups Abaqus calculates the transverse shear stiffness values required in the element formulation. If desired, toggle on **Specify values** from the **Transverse Shear Stiffnesses** options to include nondefault transverse shear stiffness effects in the composite layup definition, and enter values for K_{11}, the shear stiffness in the first direction; K_{12}, the coupling term in the shear stiffness; and K_{22}, the shear stiffness in the second direction. If either value K_{11} or K_{22} is omitted or given as zero, the nonzero value will be used for both. For more information, see *Defining the Transverse Shear Stiffness*.

Specifying the display of selected plies of a continuum shell composite layup

You can specify the display of selected plies of a continuum shell composite layup.

1. From the **Composite Layup** editor, click the **Display** tab.
2. Choose how Abaqus/CAE highlights the plies that you selected in the ply table:
   - Choose **On** to highlight the selected plies.
   - Choose **Off** to stop highlighting the selected plies. If you have a large number of plies in your model, you may want to turn off highlighting to increase performance.
3. Choose which orientation Abaqus/CAE displays on the selected plies:
   - Choose **Ply** to display the ply orientation on the selected plies.
   - Choose **Layup** to display the layup orientation on the selected plies.
   - Choose **None** to stop displaying an orientation.
4. Specify which orientation vectors you want to display on the selected plies. You can toggle the display of vectors in the 1-direction, the 2-direction, and the 3- (or normal) direction, as well as the reference direction that shows the 1-direction before rotation of the ply's material directions.

Creating solid composite layups

This section describes how to create a solid composite layup.

In most cases, you should model a composite solid as a shell or continuum shell composite layup. However, you should use a solid composite layup for the following cases:

- When the transverse shear effects are predominant.
- When you cannot ignore the normal stress.
- When you require accurate interlaminar stresses, such as near localized regions of complex loading or geometry.

Solid composite layups are expected to have a single element through their thickness, and that single element contains multiple plies that are defined in the ply table. If the region to which you assign your solid composite layup contains multiple elements, each element will contain the plies defined in the ply table, and the analysis results will not be as expected.

In this section:

- *Creating solid composite layups*
- *Specifying the plies of a solid composite layup*
- *Specifying the display of selected plies of a solid composite layup*

Creating solid composite layups

In Abaqus/Standard solid elements can include several layers of different materials for the analysis of laminated composite solids; however, in Abaqus/Explicit solid elements can be composed only of a single homogeneous material. The use of composite solids is limited to three-dimensional brick elements that have only displacement degrees of freedom. Composite solid elements are primarily intended for modeling convenience. They usually do not provide a more accurate solution than composite shell elements. For more information, see *Defining Composite Solid Elements in Abaqus/Standard* and *Modeling Thick Composites with Solid Elements in Abaqus/Standard*.

1. From the main menu bar, select **Composite->Create**.

   A **Create Composite Layup** dialog box appears.

   Tip: You can also click **Create** in the **Composite Layup Manager** or select the create composite layup tool  in the Property module toolbox.

2. Enter a composite layup name. For more information on naming objects, see *Using basic dialog box components*.
3. Specify the initial ply count. When the composite layup editor appears, it will contain a row for each ply; however, you can use the editor to subsequently add or delete plies.
4. Select **Solid** as the **Element Type**, and click **Continue**.

   The composite layup editor appears.
5. Enter a description of the layup. Abaqus/CAE displays this description in the composite layup manager.
6. Do one of the following to specify the layup orientation:

   - Select **Coordinate system** to select an existing coordinate system (or create a new coordinate system and select it), and do the following:

     1. Choose the axis that represents the **Rotation axis**.
     2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select an existing scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

   - Select **Discrete** to define a discrete orientation, and do the following:

     1. Click .
     2. In the **Edit Discrete Orientation** dialog box that appears, define the normal axis and primary axis using the procedure described in *Using discrete orientations for material orientations and composite layup orientations*.
     3. Choose the axis that represents the **Rotation axis**.
     4. Specify an additional rotation. The orientation is rotated through this angle about the selected normal axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup. Abaqus/CAE allows you to select only valid discrete fields, which, for an additional rotation, are scalar discrete fields applied to

elements. You can also create a new discrete field by clicking . For more information, see *The Discrete Field toolset*.

- Select **User-defined** to define the orientation in user subroutine *ORIENT*. This option is valid only for Abaqus/Standard analyses. See the following sections for more information:
  - *Specifying general job settings*
  - *ORIENT*
- Select the name of an orientation discrete field to specify a coordinate system that is varying spatially across the layup. You can also create a new discrete field by clicking  to the right of the **Definition** field. For more information, see *The Discrete Field toolset*. After selecting the discrete field, you must do the following:
  1. Choose the axis that represents the **Rotation axis**.
  2. Specify an additional rotation. The selected coordinate system is rotated through this angle about the selected axis. You can specify an angle, or you can select or create a scalar discrete field that defines an angle that is varying spatially across the layup.

The layup orientation is the reference orientation for any ply that uses the default orientation system (indicated by **<Layup>** in the **CSYS** column of the ply table). This orientation will be used for material calculations and stress output in the individual plies, for the section forces output, and for the transverse shear stiffness. You can specify a different orientation for the individual plies of a solid composite layup by specifying a reference orientation and/or a rotation angle. For more information, see *Understanding composite layups and orientations*.

7. Choose one of the following to specify the stacking direction of the solid elements with respect to a pair of element faces:
   - **Element direction 1**
   - **Element direction 2**
   - **Element direction 3**

   You can use the Query toolset to determine the mesh stack orientation. However, the displayed orientations account for only the sweep path; they do not account for changes to the stacking direction as described above. For more information on the Query toolset, see *Using the Query toolset to query the model*. For more information on mesh stack directions, see *Defining the Stacking and Thickness Direction*.

Specifying the plies of a solid composite layup

A composite layup is composed of a series of plies. You select the region to which a ply is assigned, and you specify the name, material, thickness, orientation, and the number of integration points of each ply. You must specify ply names that are unique throughout the entire model to ensure the correct display of ply-based results. Use the icons above the ply table or click mouse button 3 on the ply table to see a menu that allows you to edit the contents of the table cell and to manipulate the data in the table; for example, you can add and delete plies, pattern plies, and invert plies. You can also read data into the table from a file or write data from the table into a file. For more information, see *Using the ply table when defining a composite layup*.

1. From the **Composite Layup** editor, click the **Plies** tab.
2. If the plies in the composite layup are symmetric about a central core, toggle on **Make calculated sections symmetric**. Enter the plies in the ply table, starting with the bottom ply in the first row and ending with the central ply. During the analysis Abaqus appends plies to the layup definition by repeating the entered plies (including the central ply) in the reverse order to the top of the layup. Each generated ply is labeled in ply stack plots and the output database by adding `Sym_` to the beginning of the repeated ply's original name.

   This option cannot be used if the rotation angle for any of the plies in the layup is defined using a discrete field.
3. For each ply, enter the following data in the ply table:

Ply Name

The name of the ply. Abaqus/CAE displays this name when you are viewing the composite layup in the Visualization module.

Region

Select the region to which the ply is assigned. You can choose cells from the viewport, or you can select a set that refers to a cell. If you are displaying a meshed dependent part, you can choose solid elements from the viewport or select an element set. To choose elements from the viewport, you must display the native mesh, and you must use the Selection toolbar to enable the selection of **3D Elements** from the viewport. You can also select solid elements from a mesh part. For more information, see *Displaying a native mesh*, and *Filtering your selection based on the type of object*.

Material

The name of the material for this ply. Click mouse button 3, select **Edit Material** from the menu that appears, and do either of the following:

- Select the desired material from the list of available materials.
- Click  to create a new material.

Element Relative Thickness

The relative ply thickness.

For solid elements Abaqus determines the thickness from the part geometry, and the thickness may vary through the model for a given layup. Hence, the thickness values that you specify are only relative thicknesses for each ply. The actual thickness of a ply is the element thickness

times the fraction of the total thickness that is accounted for by each ply. You do not have to use physical units to specify the thickness ratios for the plies, and the sum of the ply relative thicknesses does not have to add to one.

Coordinate system

To define the coordinate system that will be used as the basis for the reference orientation of the ply, do the following:

1. Click mouse button 3, and select **Edit CSYS** from the menu that appears.

   Note:

   If you choose **Edit Orientation** from the menu, you can define both the coordinate system and the rotation angle.

2. Select the base orientation. You can select the base layup orientation, or you can select a coordinate system. If you select a coordinate system, you must select the axis that defines the normal direction.

Rotation Angle

An additional rotation (counterclockwise about the normal direction) for the ply's reference orientation. Enter a uniform rotation directly in the table, or click mouse button 3 and select **Edit Rotation Angle** to do the following:

- Select a predefined angle (0, 45, 90, or −45 degrees) to define a uniform rotation.
- Select **Uniform**, and enter an **Angle** to define a uniform rotation.
- Select a scalar discrete field to define a rotation that varies spatially across the ply. You can also create a new discrete field by clicking .

Integration Points

The number of integration points to be used through the ply. This number must be an odd number.

Specifying the display of selected plies of a solid composite layup

You can specify the display of selected plies of a solid composite layup.

1. From the **Composite Layup** editor, click the **Display** tab.
2. Choose how Abaqus/CAE highlights the plies that you selected in the ply table:
   - Choose **On** to highlight the selected plies.
   - Choose **Off** to stop highlighting the selected plies. If you have a large number of plies in your model, you may want to turn off highlighting to increase performance.
3. Choose which orientation Abaqus/CAE displays on the selected plies:
   - Choose **Ply** to display the ply orientation on the selected plies.
   - Choose **Layup** to display the layup orientation on the selected plies.
   - Choose **None** to stop displaying an orientation.
4. Specify which orientation vectors you want to display on the selected plies. You can toggle the display of vectors in the 1-direction, the 2-direction, and the 3- (or normal) direction, as well as the reference direction that shows the 1-direction before rotation of the ply's material directions.

Assigning sections, orientations, normals, and tangents to a part

This section describes how to use the **Assign** menu to assign properties to a part and how to manage section assignments.

In this section:

- *Assigning a section*
- *Managing section assignments*
- *Assigning a beam orientation*
- *Assigning a material orientation or rebar reference orientation*
- *Assigning shell/membrane normal directions*
- *Assigning beam/truss tangent directions*

Assigning a section

You can assign section properties to a part by first creating a section and then selecting **Assign->Section** to assign the section to a part; a region of a part, including its skins or stringers; or a set of elements.
Section properties that you assign to a part are assigned automatically to all instances of that part in the assembly. Abaqus/CAE colors a region green in the Property module to indicate that the region has a section assignment or colors the region yellow if there are overlapping section assignments. If you delete a section used in a section assignment, Abaqus/CAE colors the region red to indicate that the section is not found.



Note: The order of assignments may be relevant; when section assignments overlap, the last assignment will take precedence.

You can also use the **Section Assignment Manager** to view, create, edit, suppress, resume, and delete section assignments. For more information, see *Managing section assignments*. You can use the Query toolset to verify that you have assigned the correct section to a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

You can create overlapping section assignments anywhere in your model except for a region that contains Eulerian parts. Eulerian parts can be assigned only a single section assignment, and you cannot apply a second section assignment to an Eulerian part, even when its original section assignment is suppressed.

When you import a mesh from an input file, some section properties associated with that mesh may also be imported; in these cases it may be unnecessary to assign section properties to the part. For more information, see *Importing a model from an Abaqus input file*.

1. If the part to which you want to assign a section is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The part that you select appears in the current viewport.

2. From the main menu bar, select **Assign->Section**.

   Tip: You can also click **Create** in the **Section Assignment Manager** or select the  tool in the Property module toolbox.

3. Select the regions of the part from the viewport, and click mouse button 2 to indicate you have finished selecting. (For more information, see *Selecting objects within the viewport*.) If an Eulerian part is in the current viewport, Abaqus/CAE automatically selects the entire part.

   Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options* for more information.

   If you select **Skins** as the object type, you can assign the section to the entire skin or to one or more of its faces. From the prompt area, select **(pick entire skin)** or **(pick partial skin)**, then make your selection in the viewport. If the selected part has multiple skins, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you select **Stringers** as the object type, you can assign the section to the entire stringer or to one or more of its edges. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

Abaqus/CAE displays the **Region Selection** dialog box containing a list of available part sets and element sets.

b. Select the desired set, and click **Continue**.



Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click the button—**Select in Viewport** or **Sets**—on the right side of the prompt area.

An **Edit Section Assignment** dialog box appears. This dialog box contains a list of existing sections that can be assigned to the selected region or set. For example, if you selected a solid region, any existing solid sections appear in the **Edit Section Assignment** dialog box. In addition, the dialog box contains a **Create Section** button and the section type, material, and region of the displayed section.

4. In the **Edit Section Assignment** dialog box, select the section of interest and click **OK**.

Abaqus/CAE assigns the selected section to the part or set and colors the selected region green to indicate that the region has a section assignment. If there are overlapping section assignments on the region, Abaqus/CAE colors the region yellow.



Note: The order of assignments may be relevant; when section assignments overlap, the last assignment will take precedence.

5. Assign a section thickness:

From section

Use the thickness defined in the section definition.

From geometry

Use the thickness of the geometric section.

6. If you assign a homogeneous or a composite section to a shell, you can define the method used for the **Shell Offset**. Click the arrow to the right of the **Definition** field, and select the option of your choice from the list that appears:

- Select **Middle surface**, **Top surface**, or **Bottom surface** to represent the reference surface.
- Select **Specify value**, and enter a positive or negative distance (as a fraction of the shell thickness) from the midsurface to the reference surface of the shell in the **Offset ratio** field.

  A positive offset will generate nodes and elements closer to the top surface of the shell, and a negative offset will generate them closer to the bottom surface. Offsets greater than `±0.5` will cause the nodes to be beyond the surface of the shell.
- Select an existing scalar discrete field that defines an offset that is varying spatially across the section. Abaqus/CAE allows you to select only valid discrete fields, which, for an offset, are scalar discrete fields applied to elements. This option is valid only for an Abaqus/Standard analysis.

  Alternatively, you can click  to create a new discrete field. (See *The Discrete Field toolset* for more information.)
- Select **From geometry** to have Abaqus/CAE calculate the offset from the thickness defined on the geometry. This option is the default if **From geometry** is selected in the **Thickness** field. You cannot use this option to calculate the offset in an Abaqus/Explicit analysis.

7. If you want to assign sections to additional regions, repeat Steps 3 and 4. When you have finished assigning sections, use one of the following methods to exit the section assignment mode:
   - If you are selecting regions of the part from the viewport, click mouse button 2 or click **Done** in the prompt area.
   - If you are selecting preexisting sets from the **Region Selection** dialog box, click **Cancel** to close the dialog box, then click mouse button 2 or click **Done** in the prompt area.

Additional information

- *Selecting objects within the viewport*
- *Creating and editing sections*
- *Managing section assignments*
- *Assigning thicknesses and offsets*

Managing section assignments

The **Section Assignment Manager** allows you to view, create, edit, suppress, resume, and delete section assignments. The manager lists section assignments in the order of creation and displays the section name, section type, material name, and region associated with each section assignment. In the Property module, Abaqus/CAE colors a region green to indicate that the region has a section assignment or, if there are overlapping section assignments, colors the region yellow.



Note:

The order of assignments may be relevant; when section assignments overlap, the last assignment will take precedence.

1. From the main menu bar, select **Section->Assignment Manager**.
2. From the bottom of the manager, choose from the following:
   - If you want to create a new section assignment, click **Create** and follow the procedure described in *Assigning a section*. The new section assignment appears in the manager.
   - If you want to edit a section assignment, select the section assignment in the manager and click **Edit**. The **Edit Section Assignment** dialog box appears, and the region to which the section is currently assigned is highlighted in the viewport.
     1. If desired, select a different section for the section assignment.
     2. Click  to select a new region for the section assignment.
     3. Select the region from the viewport or from a list of existing sets, and click mouse button 2 to indicate that you have finished selecting.
     4. Click **OK** to close the **Edit Section Assignment** dialog box.
   - If you want to delete a section assignment, select the section assignment in the manager and click **Delete**.
3. In the left column of the manager, click a green check mark to suppress a section assignment or click a red “X” to resume a suppressed section assignment.

   Note:

   Suppressing and resuming a connector section assignment will suppress or resume the corresponding connector orientation.

   Abaqus/CAE updates the color of the selected part or region to reflect the revised section assignment status.
4. Click **Dismiss** to close the **Section Assignment Manager** dialog box.

Additional information

- *Defining sections*
- *Creating and editing sections*
- *Assigning a section*

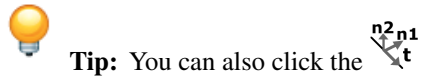
Assigning a beam orientation

After you have assigned beam sections to the wire regions of a part or to its stringers, you must assign an orientation to the beam sections by defining the approximate local 1-direction of the cross-section.
The beam orientations that you assign to a part are assigned automatically to all instances of that part in the assembly. You can use the Query toolset to verify that you have assigned the correct beam orientation to a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

1. If the part to which you want to assign an orientation is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The selected part appears in the current viewport.

2. From the main menu bar, select **Assign->Beam Section Orientation**.

   Tip: You can also click the  tool in the Property module toolbox.

3. Select the wire region of the part from the viewport, and click mouse button 2 to indicate that you have finished selecting. (For more information, see *Selecting objects within the viewport*.)

   Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options* for more information.

   If you select **Stringers** as the object type, you can assign the beam orientation to the entire stringer or to one or more of its edges. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available part sets. Part sets containing only wires appear in the list.

   b. Select the part set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click the button—**Select in Viewport** or **Sets**—on the right side of the prompt area.

4. Abaqus/CAE prompts you to enter the vector representing an approximate $\mathbf{n}_1$-direction (the local 1-direction of the cross-section). For more information on defining the $\mathbf{n}_1$-direction, see *Beam Element Cross-Section Orientation*.

   Note:

   If the part was created in two-dimensional modeling space, the $\mathbf{n}_1$-direction is always normal to the *X*–*Y* plane (0.0, 0.0, −1.0).

   Abaqus/CAE displays the resulting ($\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{t}$) axis system on the selected wire regions.

5. If the displayed ($\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{t}$) axis system is correct, click **OK** in the prompt area to confirm your choice.

   If you wish to change the beam orientation, click the **Previous** button () and enter a new $\mathbf{n}_1$-direction.
6. If you want to assign beam orientations to additional wire regions, repeat Steps 3 through 5.
7. Click mouse button 2 to indicate you have finished assigning beam orientations.

Additional information

- *Creating and editing sections*
- *Selecting objects within the viewport*
- *Plotting material orientations*
- *Beam Element Cross-Section Orientation*

Assigning a material orientation or rebar reference orientation

You can assign a material orientation or a rebar reference orientation.

In this section:

- *Assigning a material orientation*
- *Assigning a rebar reference orientation*

Assigning a material orientation

The global coordinate system determines the default material orientations. However, you can assign specific material orientations to a shell, solid part, region, skin, or stringer by selecting a datum coordinate system or discrete field or by defining a discrete orientation. For an Abaqus/Standard analysis you can define the local coordinate system in user subroutine *ORIENT*. The datum coordinate systems can be rectangular, cylindrical, or spherical; and you can apply an additional rotation to the coordinate system if required. For more information, see *Orientations*.

In cases where Abaqus/CAE attempts to draw coordinate systems for material orientations at singularity points of the system (i.e., points for which the user-selected coordinate system and the geometric normal from either geometry or elements cannot be resolved into a valid orientation for display purposes in Abaqus/CAE), the coordinate system will be drawn collapsed.

For continuum shell elements, Abaqus/CAE does not project the displayed material orientations onto the midplane surface. You can perform a data check and view the output database in the Visualization module to verify the material orientations. For more information, see *Performing a data check on a model*.

The material orientations that you assign to a part or region are assigned automatically to all instances of that part in the assembly. You can use the Query toolset to verify that you have assigned the correct material orientation to a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

You can edit, suppress, and resume previously defined material orientations by clicking mouse button 3 on the material orientation in the Model Tree and selecting the suppress or resume option from the list that appears. For more information, see *Suppressing and resuming objects*.

1. If the part to which you want to assign a material orientation is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The selected part appears in the current viewport.

2. From the main menu bar, select **Assign->Material Orientation**.

   Tip: You can also click the  tool in the Property module toolbox.

3. Select the shell or solid regions from the viewport, and click mouse button 2 to indicate you have finished selecting. (For more information, see *Selecting objects within the viewport*.)

   Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options* for more information.

   If you select **Skins** as the object type, you can assign the material orientation to the entire skin or to one or more of its faces. From the prompt area, select **(pick entire skin)** or **(pick partial skin)**, then make your selection in the viewport. If the selected part has multiple skins, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you select **Stringers** as the object type, you can assign the material orientation to the entire stringer or to one or more of its edges. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available part sets. Part sets containing only shells or solids appear in the list.

b. Select the part set of interest, and click **Continue**.



Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. If you want to specify a material orientation using an existing datum coordinate system, do the following:

a. Select the datum coordinate system.

- Select a coordinate system in the viewport.
- Select a datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.

Abaqus/CAE displays the resulting material orientation specified for the region, and the **Edit Material Orientation** dialog box appears.

b. Choose **Axis 1**, **Axis 2**, or **Axis 3** to specify the coordinate system axis.

- For a shell region, specify the coordinate system axis that approximates the normal to the shell in the **Normal Direction** portion of the dialog box.
- For a solid region, specify the coordinate system axis about which an optional additional rotation occurs in the **Additional Rotation Direction** portion of the dialog box.

c. If desired, you can provide an additional rotation of the material orientation about the shell normal (for a shell region) or about a local axis of the datum coordinate system (for a solid region) using one of the following methods:

- Choose **Angle**, and enter a value in degrees for the additional rotation.
- Choose **Distribution**, and select an existing scalar discrete field that defines an angle that is spatially varying across the region. Alternatively, you can click  to create a new discrete field. (See *The Discrete Field toolset* for more information.)

Abaqus/CAE applies the additional rotation, if any, and displays the resulting material orientation applied to the region.

5. If you want to use the global coordinate system (default) or specify a material orientation in a discrete orientation, user subroutine, or a discrete field, click **Use Default Orientation or Other Method** from the prompt area. You can also select an existing datum coordinate system or create a new one using this procedure.

a. In the **Edit Material Orientation** dialog box that appears, click the arrow to the right of the **Definition** field, and select one of the following:

- Select **Coordinate system**, and click  to select a coordinate system from the viewport or by name. Alternatively, you can click  to create a new datum coordinate system.
- Select **Global** to use the global coordinate system or to reset a previously assigned material orientation back to the global coordinate system.
- Select **Discrete**, and click  to define a discrete orientation. In the **Edit Discrete Orientation** dialog box that appears, define the normal axis and primary axis using the procedure described in *Using discrete orientations for material orientations and composite layup orientations*.

- Select **User-defined** to define the orientation in user subroutine *ORIENT*. This option is valid only for Abaqus/Standard analyses. See the following sections for more information:
  - *Specifying general job settings*
  - *ORIENT*

- Select a discrete field to define a spatially varying orientation. Alternatively, you can click  to the right of the **Definition** field to create a new discrete field. (See *The Discrete Field toolset* for more information.)

b. Choose **Axis 1**, **Axis 2**, or **Axis 3** to specify the coordinate system axis.

- For a shell region, specify the coordinate system axis that approximates the normal to the shell in the **Normal Direction** portion of the dialog box.
- For a solid region, specify the coordinate system axis about which an optional additional rotation occurs in the **Additional Rotation Direction** portion of the dialog box.

c. If desired, you can provide an additional rotation of the material orientation about the shell normal (for a shell region) or about a local axis of the datum coordinate system (for a solid region) using one of the following methods:

- Choose **Angle**, and enter a value in degrees for the additional rotation.
- Choose **Distribution**, and select an existing scalar discrete field that defines an angle that is spatially varying across the region. Alternatively, you can click  to create a new discrete field. (See *The Discrete Field toolset* for more information.)

Abaqus/CAE applies the additional rotation, if any, and displays the resulting material orientation applied to the region.

d. Choose one of the following to specify the stacking direction of continuum shell elements or solid elements:

- **Element isoparametric direction 1**
- **Element isoparametric direction 2**
- **Element isoparametric direction 3 (bottom to top)**
- **Normal direction of material orientation (Continuum Shells only)**. The stacking direction is the 3-direction of the material orientation.

If the selected regions are meshed, Abaqus/CAE displays the material stacking direction for each element.

6. Click **OK** in the dialog box.

Additional information

- *Creating and editing sections*
- *Selecting objects within the viewport*
- *Plotting material orientations*
- *The Discrete Field toolset*
- *Controlling datum display*
- *Solid (Continuum) Elements*
- *About Shell Elements*

Assigning a rebar reference orientation

The angular orientation of a rebar layer is defined relative to the rebar reference orientation. The global coordinate system determines the default rebar reference orientation. However, you can assign a rebar reference orientation to a shell part, region, skin, or stringer by selecting an existing datum coordinate system from the viewport and then selecting an axis on the datum coordinate system that approximates the direction of the shell normal.
The datum coordinate system can be rectangular, cylindrical, or spherical; and you can apply an additional rotation to the coordinate system if required. For more information, see *Orientations*.

The rebar reference orientations that you assign to a part or region are assigned automatically to all instances of that part in the assembly. You can use the Query toolset to verify that you have assigned the correct rebar reference orientation to a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

1. If the part to which you want to assign a rebar reference orientation is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The selected part appears in the current viewport.

2. From the main menu bar, select **Assign->Rebar Reference Orientation**.

   Tip: You can also click the  tool in the Property module toolbox.

3. Select the regions from the viewport, and click mouse button 2 to indicate you have finished selecting. (For more information, see *Selecting objects within the viewport*.)

   Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options* for more information.

   If you select **Skins** as the object type, you can assign the rebar reference orientation to the entire skin or to one or more of its faces. From the prompt area, select **(pick entire skin)** or **(pick partial skin)**, then make your selection in the viewport. If the selected part has multiple skins, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you select **Stringers** as the object type, you can assign the rebar reference orientation to the entire stringer or to one or more of its edges. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the part set of interest, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. Specify a local coordinate system in the selected region using one of the following methods:

   - Select a coordinate system in the viewport.
   - Select a datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.

- Click **Use Default Orientation** to use the global coordinate system.

Abaqus/CAE displays the resulting rebar orientation specified for the region, and the **Edit Rebar Orientation** dialog box appears.

5. Choose **Axis 1**, **Axis 2**, or **Axis 3** to specify the coordinate system axis.
   - For a shell region, specify the coordinate system axis that approximates the normal to the shell in the **Normal Direction** portion of the dialog box.
   - For a solid region, specify the coordinate system axis about which an optional additional rotation occurs in the **Additional Rotation Direction** portion of the dialog box.
6. If desired, you can provide an additional rotation of the rebar reference orientation about the shell normal (for a shell region) or about a local axis of the datum coordinate system (for a solid region). Choose **Angle**, and enter a value in degrees for the additional rotation.

   Abaqus/CAE applies the additional rotation, if any, and displays the resulting rebar reference orientation applied to the region.
7. Click **OK** in the dialog box.

Additional information

- *Creating and editing sections*
- *Selecting objects within the viewport*
- *Controlling datum display*
- *About Shell Elements*

Assigning shell/membrane normal directions

When you create a part with shell regions or an axisymmetric part with wire regions, Abaqus assigns a direction for the normals of the regions. You can reverse the directions of the normals for these regions. In addition, you can reverse the directions for the normals of imported parts or selected orphan elements.

You can also reverse the normals for skin reinforcements; however, if there are multiple skins defined on the surface of a geometric part, you cannot reverse the normal of one of those skins without reversing the normals of every skin defined for that surface. If the surface with multiple skins is a feature of a mesh part, you can reverse the normal of an individual skin without reversing the normals for the other skins defined on that surface.

When you reverse the normal direction of a region, the local normal that defines the material orientation is reversed. You can use the Query toolset to verify that the material orientation is correct for a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

1. If the part to which you want to assign a normal is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The selected part appears in the current viewport.

2. From the main menu bar, select **Assign->Element Normal**.

   Tip: You can also click the 

   tool in the Property module toolbox or select **Mesh->Orientation->Normal** in the Mesh module.

   Abaqus/CAE displays the part using the shaded render style. The side of the shell where the surface normal coincides with the shell normal (top face) is colored brown; the opposite side (bottom face) is colored purple.

3. If the selected part is geometry-based, use one of the following methods to select the regions whose normals you want to reverse:

   - Select the regions from the viewport and click mouse button 2. (For more information, see *Selecting objects within the viewport*.)

     Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options*, for more information.

     If you select **Skins** as the object type, you can assign the section to the entire skin or to one or more of its faces. From the prompt area, select **(pick entire skin)** or **(pick partial skin)**, then make your selection in the viewport. If the selected part has multiple skins, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available part sets.

     2. Select the part set of interest, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   Abaqus/CAE reverses the shell/membrane normal directions of the selected regions.

4. If the selected part is a mesh part, use one of the following methods to select the shell elements whose normals you want to reverse:

Selecting individual elements:

1. Click the arrow next to the **Selection method** field in the prompt area, and select **Individually** from the list that appears.
2. Select an element whose normal you want to flip.
3. **[Shift] + Click** on additional elements to add them to your selection.
4. If necessary, **[Ctrl] + Click** on selected elements to unselect them.
5. When you have finished selecting elements, click mouse button 2.

Specifying an existing element set

1. Click **Sets** on the right side of the prompt area.

   Abaqus/CAE displays the **Region Selection** dialog box containing a list of element sets that you have created.
2. Select the set of elements that you want to edit, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

Selecting elements using the angle method:

1. From the field in the prompt area, select **by angle**.
2. Enter an angle (from 0° to 90°), and select an element face. Abaqus/CAE selects every adjacent shell element from the selected face until the angle between the element faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects*, for more information.)
3. After you use the **by angle** method, you can **[Shift] + Click** on additional elements to add them to your selection or **[Ctrl] + Click** on selected elements to remove them from your selection. (See *Combining selection techniques*, for more information.)
4. When you have finished selecting elements, click mouse button 2.



Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. For more information, see *Selecting objects within the viewport*."

5. From the prompt area, select a method for reversing the shell/membrane normal:
   - Click **Flip All** to reverse the normal of all selected elements.
   - Click **Select Normal** to change the normals of the selected elements so that they point in the same direction as the normal of a reference element that you specify.
6. If you chose **Select Normal** in the previous step, do the following:

a. In the viewport, select the reference element.

b. In the prompt area, click **OK**.

Abaqus/CAE changes the normals of the selected elements so that they point in the same direction as the reference element normal.

7. Repeat the previous steps as often as necessary to reverse additional shell/membrane normals. When you have finished assigning normal directions, use one of the following methods to exit the assignment mode:

- If you are selecting regions of the part or elements from the viewport, click mouse button 2 or click **Done** in the prompt area.
- If you are selecting preexisting sets from the **Region Selection** dialog box, click **Cancel** to close the dialog box, then click mouse button 2 or click **Done** in the prompt area.

Abaqus/CAE reverts to the render style that was selected prior to starting the procedure to reverse the shell/membrane normals.

Additional information

• *Creating and editing sections*

• *Selecting objects within the viewport*

Assigning beam/truss tangent directions

When you create a part with wire regions, Abaqus assigns the default tangent directions to the regions. Beam section orientations depend on the tangent directions. You can reverse the direction of a beam or truss tangent to obtain the desired beam section orientation.
You can use the Query toolset to verify that the beam section orientations are correct for a selected region. For more information, see *Using the Query toolset to obtain assignment information*.

You can also reverse the tangent directions for stringer reinforcements; however, if there are multiple stringers defined on the edge of a geometric part, you cannot reverse the tangent of one of those stringers without reversing the tangents of every stringer defined for that edge. If the edge with multiple stringers is a feature of a mesh part, you can reverse the tangent of an individual stringer without reversing the tangents for the other stringers defined for that edge.

1. If the part to which you want to assign a tangent is not visible in the current viewport, click the name of the desired part in the **Part** list located in the context bar.

   The selected part appears in the current viewport.

2. From the main menu bar, select **Assign->Element Tangent**.

   Tip: You can also click the tool in the Property module toolbox or select **Mesh->Orientation->Tangent** in the Mesh module.

   Abaqus/CAE displays an arrow in the direction of the tangent at the center of each region.

3. Use one of the following methods to select the regions whose tangents you want to reverse:

   - Select the regions from the viewport, and click mouse button 2. (For more information, see *Selecting objects within the viewport*.)

     Tip: You can limit the types of objects that you can select in the viewport by using the tools in the **Selection** toolbar. See *Using the selection options* for more information.

     If you select **Stringers** as the object type, you can assign the beam/truss tangent direction to the entire stringer or to one or more of its edges. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE will display the ambiguous picking options in the prompt area while you make your selection.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available part sets.

     2. Select the part set of interest, and click **Continue**.

   Abaqus/CAE reverses the direction of the tangents for the selected regions.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. Repeat the previous step as often as necessary to reverse additional tangent directions. When you have finished assigning tangent directions, use one of the following methods to exit the assignment mode:

- If you are selecting regions of the part from the viewport, click mouse button 2 or click **Done** in the prompt area.
- If you are selecting preexisting sets from the **Region Selection** dialog box, click **Cancel** to close the dialog box, then click mouse button 2 or click **Done** in the prompt area.

Additional information

- *Creating and editing sections*
- *Selecting objects within the viewport*

Using discrete orientations for material orientations and composite layup orientations

A discrete orientation defines a spatially varying orientation at the centroid of each mesh element. The orientation can be based on the topology of the part, allowing you to define a continually varying orientation. You define the normal axis and a primary axis, and Abaqus/CAE uses these axes to construct a right-handed Cartesian coordinate system. For two-dimensional and axisymmetric parts, the normal axis is always the out-of-plane direction and represents the positive 3-axis of the coordinate system. Discrete orientations can be used for material orientations and composite layup orientations.

A variety of selection methods are available to define the desired axes. The datum axis and vector value methods define a constant axis direction, whereas the region selection methods allow for varying orientations. For example, you may want to select a curved surface and allow the normal to follow the shape of the curve.

Abaqus/CAE uses the normal axis and primary axis that you specify and the following algorithm to calculate the discrete orientation:

1. Abaqus/CAE preserves the normal axis direction and uses the specified primary axis direction as a construction direction. If the specified primary axis direction and the normal axis direction form a 90° angle, the final primary axis direction will be the same as the one specified. Otherwise, the primary axis direction will be adjusted to form a Cartesian system.
2. If you define the normal axis by selecting a surface or face, Abaqus/CAE finds the closest point on the surface or face and uses the surface normal at that point as the normal axis direction ($\widehat{N}$).
3. If you define the primary axis by selecting an edge, Abaqus/CAE finds the closest point on the edge and uses the edge tangent at that point as the primary axis construction direction ($\widehat{P_1}$).
4. Abaqus/CAE computes the secondary axis direction ($\widehat{S}$) by taking the cross product of the primary axis direction and the normal axis direction ($\widehat{S} = \widehat{P_1} \times \widehat{N}$).
5. Abaqus/CAE computes the final primary axis direction ($\widehat{P}$) by taking the cross product of the secondary axis direction and the normal axis direction ($\widehat{P} = \widehat{S} \times \widehat{N}$).

    1. To display the **Edit Discrete Orientation** dialog box, do the following:
        a. Assign a material orientation or create a composite layup.
            - Display the **Edit Material Orientation** dialog box using the procedure described in *Assigning a material orientation or rebar reference orientation*.
            - Display the composite layup editor using one of the procedures described in *Creating and editing composite layups*.

        b. Click the arrow to the right of the **Definition** field, and select **Discrete** from the list that appears.
        c. Click .
    2. For three-dimensional parts, choose the coordinate system axis that you want the **Normal axis direction** to represent in the resultant orientation.
    3. For three-dimensional parts, define the normal axis of the orientation using one of the following methods:
        - Select **Surface/Faces** to select a region, and do the following:
            1. Click .

2. From the prompt area, select **individually** or select **by angle** and enter an angle. For more information, see *Using the angle and feature edge method to select multiple objects*.
3. Identify the regions that define the normal axis of the material orientation by selecting the regions in the viewport or by clicking **Surfaces** or **Sets** in the prompt area and selecting an existing surface or set.

- Select **Datum axis** to select datum geometry, and do the following:

  1. To create a datum axis, click  and select two points in the viewport that define the axis.
  2. Click .
  3. Select the datum axis that defines the normal axis of the material orientation.

- Select **Vector (i,j,k)**, and enter the values for the vector components.

Abaqus/CAE displays arrows in the viewport that indicate your selections and axis direction; for example, arrows labeled `N-3`. If necessary, you can click **Flip Direction** to obtain the desired direction.

4. Choose the coordinate system axis that you want the **Primary axis direction** to represent in the resultant orientation.
5. Define the primary axis of the orientation using one of the following methods:

- Select **Edges** to select a region, and do the following:

  1. Click .
  2. From the prompt area, select **individually** or select **by edge angle** and enter an angle.
  3. Identify the regions that define the primary axis of the material orientation by selecting the regions in the viewport or by clicking **Sets** in the prompt area and selecting an existing set.

- Select **Datum axis** to select datum geometry, and do the following:

  1. To create a datum axis, click  and select two points in the viewport that define the axis.
  2. Click .
  3. Select the datum axis that defines the primary axis of the material orientation.

- Select **Vector (i,j,k)**, and enter the values for the vector components.

Abaqus/CAE displays arrows in the viewport that indicate your selections and axis direction; for example, arrows labeled `P-1`. If necessary, you can click  to obtain the desired direction.

6. Click **Continue** to save the discrete orientation definition and to close the **Edit Discrete Orientation** dialog box.

Additional information

- *Creating conventional shell composite layups*
- *Creating continuum shell composite layups*
- *Creating solid composite layups*
- *Assigning a material orientation*

Creating material calibrations

This section describes how to import material data into Abaqus, filter and process the data, and use the data to specify Abaqus material behaviors.

In this section:

- *What is material calibration?*
- *Creating and editing data sets for calibration*
- *Processing calibration data*
- *Defining calibration behaviors*

What is material calibration?

Material calibration is the process of deriving Abaqus material behaviors like elasticity and plasticity from sets of material test data. You can create material calibrations in three general steps:

- Create data sets or import data sets into Abaqus/CAE.
- Process these data sets using filters and tools that enable you to scale, smooth, and truncate the data and to convert the data between nominal and true forms.
- Derive material properties such as Young's modulus, Poisson's ratio, or Mullins effect from your data sets.

To begin a new material calibration, double-click **Calibrations** in the Model Tree. In the **Create Calibration** dialog box that appears, enter a calibration name and click **OK**. Abaqus/CAE adds the new calibration to the Model Tree.

Creating and editing data sets for calibration

You can import material data from text (`.txt`) files and customize these data in a data table. You can also specify quantity types to describe the data, such as stress and strain or force and displacement.

1. From the Model Tree, expand **Calibrations** and double-click **Data Sets**.

   The **Create Data Set** dialog box appears.

2. To import a data set from a file, do the following:

   a. Click **Import Data Set**.

      The **Read Data From Text File** dialog box appears.

   b. Navigate to the file you want to import. For more information about file selection, see *Using file selection dialog boxes*.

   c. If the data in your imported file are delimited by a character other than spaces, tabs, or commas, toggle on **other** and enter the delimiting character in the **Delimiter** field.

   d. From the **Data Set Type** options, select the pairing of output variables that describes your imported data. The default type is **Stress/Strain**.

   e. If desired, specify the numbers of the fields from which Abaqus/CAE should import data. By default, Abaqus/CAE reads data from the first and second fields in the data set.

   f. If desired, specify a **Data Set Form** of **True** to import the material data in true form. By default, material data from text files are imported in nominal (or engineering) form.

   Abaqus/CAE displays the imported data and the selected quantity types in the table in the dialog box.

3. If desired, customize the data displayed in the table by adding, deleting, or modifying rows. For more information about tabular data entry, see *Entering tabular data*.

4. Click **OK** to create the new data set and to close the **Create Data Set** dialog box.

   Abaqus/CAE displays the new data set in the **Data Sets** container in the Model Tree and plots the data in the viewport.

Processing calibration data

The calibration data processing options enable you to clean up your material data before using it to define material behaviors.

In this section:

- *Converting calibration data between nominal and true forms*
- *Scaling calibration data*
- *Smoothing calibration data*
- *Truncating and shifting calibration data*

Converting calibration data between nominal and true forms

The **Convert** option enables you to convert the data in a calibration data set from nominal form to true form or vice versa. Abaqus/CAE performs the conversion using the algorithms described in *Stress and Strain Measures*.

1. In the Model Tree, click mouse button 3 on the data set that you want to convert and select **Process** from the menu that appears.

   The **Data Set Processing** dialog box appears.

2. Select **Convert**, and click **Continue**.

   The **Change Data Set Form** dialog box appears.

3. Select the form to which you want to convert the selected data set. Options are **Nominal (Engineering) Form** and **True Form**.
4. If desired, you can retain the original data set and create a new data set when you perform the data conversion. You can also customize the name of the newly created data set.
5. Click **OK**.

   Abaqus/CAE performs the selected form conversion for the data.

Scaling calibration data

The **Scale** option enables you to apply different scaling factors to either column in a calibration data set.

1. In the Model Tree, click mouse button 3 on the data set that you want to convert and select **Process** from the menu that appears.

   The **Data Set Processing** dialog box appears.

2. Select **Scale**, and click **Continue**.

   The **Scale Data Set** dialog box appears.

3. Specify a scaling factor for either column in the data set by editing the values in the **Col 1** or **Col 2** fields.
4. If desired, click **Preview** to review the new data values created by the scaling factor values.

   Abaqus/CAE reflects the new scale factor for each column in the data set.

5. If desired, you can retain the original data set and create a new data set when you perform the data conversion. You can also customize the name of the newly created data set.
6. Click **OK**.

   Abaqus/CAE performs the selected scaling for the data.

Smoothing calibration data

The **Smooth** option enables you to operate on a calibration data set to produce data that has a smoother curve. The resulting data set has the same *X*-coordinate values as the original data set. Abaqus/CAE uses an exponential moving average algorithm to smooth the data; this algorithm is also used by the **smooth2***X–Y* operator. For more information, see *Smoothing an X–Y data object*.

1. In the Model Tree, click mouse button 3 on the data set that you want to convert and select **Process** from the menu that appears.

   The **Data Set Processing** dialog box appears.

2. Select **Smooth**, and click **Continue**.

   The **Smooth Data Set** dialog box appears.

3. In the **Weight** field, specify a value for the smoothing parameter. You can specify a smoothing factor between 0 and 1 for the curve; a smaller value produces a smoother curve. The default smoothing factor is 0.75.
4. If desired, you can retain the original data set and create a new data set when you perform the data conversion. You can also customize the name of the newly created data set.
5. Click **OK**.

   Abaqus/CAE smooths the data set.

Truncating and shifting calibration data

The **Truncate and Shift** option enables you to exclude data points from a calibration data set whose *X*-coordinate values are less than or greater than a user-specified value or to include data points that lie between two specified *X*-coordinate values.

1. In the Model Tree, click mouse button 3 on the data set that you want to convert, and select **Process** from the menu that appears.

   The **Data Set Processing** dialog box appears.

2. Select **Truncate and Shift**, and click **Continue**.

   The **Truncate Data Set** dialog box appears.

3. Specify a truncation point on the left side or the right side of the data set, or specify points on both sides of the data set. You can set a truncation point by dragging a slider in the viewport or by entering a value in the appropriate **X-axis** field.
4. Shift the selected subset of data points using either of the following techniques:

   - Toggle on **Offset first point to (0,0)** to shift the selected subset of data points so that its leftmost point resides at (0,0).
   - Shift the selected subset of data points along either axis by specifying values in the **Offset X-axis** or **Offset Y-axis** fields.

5. If desired, you can retain the original data set and create a new data set when you perform the data conversion. You can also customize the name of the newly created data set.
6. Click **OK**.

   Abaqus/CAE performs the specified truncation or shift.

Defining calibration behaviors

This section describes how to extract material behavior constants from calibration data sets.

You can also add support for custom calibration behaviors, which appear as new options in the **Calibration Behavior** dialog box. For more information, see "Creating custom material calibration plug-ins in Abaqus/CAE" in the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Abaqus/CAE assumes that the material is fully incompressible when it calculates material properties and plots material response curves from test data other than the volumetric test.

In this section:

- *Calibrating data for isotropic elastic material behavior*
- *Calibrating data for isotropic elastic-plastic material behavior*
- *Calibrating data for hyperelasticity with permanent set*

Calibrating data for isotropic elastic material behavior

The isotropic elastic calibration behavior enables you to derive isotropic elastic data (Young's modulus and Poisson's ratio) from calibration data sets and to apply these material constants to the elastic material properties of a material definition in your model.

1. From the Model Tree, expand the **Calibrations** container and double-click **Behaviors**.

   The **Create Calibration Behavior** dialog box appears.

2. Enter a name for the material calibration behavior, select **Elastic Isotropic**, and click **Continue**.

   The **Edit Behavior** dialog box appears.

3. From the **Parameter Set 1** options, do the following to calculate a value for Young's modulus:

   a. From the **Data set** list, select the data from which you want to calculate Young's modulus.

   b. Click .

      Abaqus/CAE computes Young's modulus and displays its value to the right of the **Young's modulus** label.

4. From the **Parameter Set 2** options, do the following to calculate a value for Poisson's ratio:

   a. From the **Data set** list, select the data from which you want to calculate Poisson's ratio.

   b. Click .

      Abaqus/CAE computes Poisson's ratio and displays its value to the right of the **Poisson's ratio** label.

5. From the **Material** list, select the material definition to which you want to apply this calibration behavior; or click  to create a new material definition for this calibration behavior. For more information about defining a new material model, see *Creating or editing a material*.

6. Click **OK** to save the isotropic elastic behaviors to the selected material model.

   Abaqus/CAE adds the new behavior to the model tree and adds the specified Young's modulus and Poisson's ratio to the **Elastic** material properties for the specified material.

Additional information

• *Creating a linear elastic material model*

Calibrating data for isotropic elastic-plastic material behavior

The isotropic elastic-plastic calibration behavior enables you to derive isotropic elastic and plastic material behaviors.

1. From the Model Tree, expand the **Calibrations** container and double-click **Behaviors**.

   The **Create Calibration Behavior** dialog box appears.

2. Enter a name for the material calibration behavior, select **Elastic Plastic Isotropic**, and click **Continue**.

   The **Edit Behavior** dialog box appears.

3. From the **Elastic-Plastic Data** options, do the following:

   a. Expand the **Data set** list and select the data from which you want to calculate the first set of calibration values.

   b. From the **Ultimate point** options, either click  to calculate the ultimate point automatically or click  and select the ultimate point from the viewport.

   Abaqus/CAE plots the ultimate point in the viewport and displays its coordinates in the dialog box.

   c. From the **Yield point** options, click  and pick the yield point from the viewport.

   Abaqus/CAE plots a line between the origin and the yield point in the viewport, displays the coordinates for the yield point in the dialog box, and calculates the Young's modulus and displays its value to the right of the **Young's modulus** label.

   d. Select the plastic points for this material calibration by doing either of the following:

   - Drag the **Plastic points** slider to the right to calculate a greater number of plastic points or drag the slider to the left to calculate fewer plastic points.
   - Click  to pick plastic points from the viewport.

   Abaqus/CAE adds plastic data points to the table in the dialog box. You can edit any of these data if you want to customize further the plastic data.

4. From the **Poisson's Ratio Data** options, do the following:

   a. From the **Data set** list, select the data from which you want to calculate Poisson's ratio.

   b. Click .

   Abaqus/CAE computes the Poisson's ratio, displays its value in the **Poisson's ratio** field, and plots it in the viewport. If desired, you can adjust the calculated value of Poisson's ratio by changing the value in the field.

5. From the **Material** list, select the material definition to which you want to apply this calibration behavior; or click  to create a new material definition for this calibration behavior. For more information about defining a new material model, see *Creating or editing a material*.

6. Click **OK**.

   Abaqus/CAE updates the new calibration behavior. If you specified a material definition, Abaqus/CAE maps the isotropic elastic-plastic calibration behavior parameters to the **Elastic** and **Plastic** material behaviors of that material definition.



Note:

Any elastic or plastic material behaviors in the selected material are overwritten when you map data from a calibration behavior to the material definition.

Additional information

- *Creating a linear elastic material model*
- *Defining classical metal plasticity*

Calibrating data for hyperelasticity with permanent set

The hyperelasticity with permanent set calibration behavior enables you to extract plastic and hyperelastic material behaviors and Mullins effect from uniaxial and biaxial data sets of the loading, unloading, and reloading of elastomers and thermoplastics. You can extract data from a uniaxial test, a biaxial test, or from both types of tests. The calibration process includes the following steps:

1. Upload uniaxial and/or biaxial test data files into Abaqus/CAE as new data sets.
2. Extract the loading, unloading, and reloading cycles and the permanent set data from the data files you provide and create separate data sets for the load, unload, and reload phase of every cycle.
3. If desired, select any data cycles that you want to exclude from calculations of material behavior.
4. Select the yield point from the viewport and, if desired, edit individual points on the primary loading data set to create a smoother curve. The permanent set curves are based upon the current yield point, so these curves also change when you select a new yield point.
5. Once you determine the test data sets that you want to use for deriving material behaviors and you specify primary curve options, you can derive material behaviors from the selected data. Abaqus/CAE maps plastic, hyperelastic, and Mullins effect material behaviors to the material you select.

    1. From the Model Tree, expand the **Calibrations** container and double-click **Behaviors**.

       The **Create Calibration Behavior** dialog box appears.
    2. Enter a name for the material calibration behavior, select **Hyperelasticity with permanent set**, and click **Continue**.

       The **Edit Behavior** dialog box appears.
    3. Perform the following steps from the **Uniaxial** or **Biaxial** tabbed page:
        a. Select the cycle from which to extract data to calibrate the Mullins effect. By default, Abaqus/CAE extracts the last unloading and reloading curves.

           - Select **Last cycle found** to extract the last unloading and reloading curves from each strain level in the supplied test data.
           - Select **First cycle found** to extract the first unloading and reloading curves from each strain level in the supplied test data.

        b. Expand the **Data set** list, and select the data from which you want to calculate the calibration values for uniaxial or biaxial data tests.

        c. Click .

           Abaqus/CAE extracts the primary loading curve, the specified unloading and reloading curves for each cyclic strain level, and the permanent set curves, then creates new calibration data sets for each component of each cyclic strain level. Each new data set is available in the **Uniaxial Test Data Sets** or **Biaxial Test Data Sets** options and is plotted in the viewport.

        d. Toggle on the individual loading, unloading, or reloading data sets that you want to include in the material calibration calculations. As you toggle on a data set, Abaqus/CAE displays its corresponding *X–Y* curve in the viewport. You can select any of the following:

           - Select **All** to include all the raw data found in the selected test data file.
           - Select **Primary** to include data from the primary loading curve.
           - Select **Unloading** to include data from the unloading curves for each cyclic strain level, or expand this container to select individual unloading curves.

- Select **Reloading** to include data from the reloading curves for each cyclic strain level, or expand this container to select individual reloading curves.
- Select **Permanent Set** to include data from both permanent set curves, or expand this container to select either stress- or strain-related components of permanent set.

e. From the **Yield Point** options, do either of the following:

- Click  , and select the yield point on the primary curve from the viewport.
- Click  , and enter either the **Strain** or **Stress** value; Abaqus/CAE calculates the other value from the primary curve and populates the remaining field.

4. If desired, extract a second data set from the **Uniaxial** or **Biaxial** tabbed page.

5. If you extracted both uniaxial and biaxial test data, Abaqus/CAE applies the data equally by default in the calculations of material behaviors. Perform the following steps from the **Options** tabbed page if you want one data set to have greater weight in these calculations:

a. In the **Material Properties** options, drag the **Weight** slider toward the type of data (uniaxial or biaxial) that you want to assign greater weight in the material behavior calculations.

b. Specify whether the selection of relative weight is based on a linear interpolation or a logarithmic interpolation.

6. From the **Material** list, select the material definition to which you want to apply this calibration behavior; or click  to create a new material definition for this calibration behavior. For more information about defining a new material model, see *Creating or editing a material*.

7. Click **OK**.

Abaqus/CAE updates the new calibration behavior and maps the hyperelasticity with permanent set calibration behavior parameters to the **Hyperelastic**, **Plastic**, and **Mullins Effect** material behaviors of that material definition.

 **Note:**

Any hyperelastic, plastic, or Mullins effect material behaviors in the selected material are overwritten when you map data from a calibration behavior to the material definition.

Additional information

- *Defining classical metal plasticity*
- *Hyperelastic Behavior of Rubberlike Materials*
- *Stress Softening in Rubberlike Materials*

Using the Special menu in the Property module

You can use the **Special** menu in the Property module to define the following engineering features:

- **Skin**. A skin reinforcement defines a skin that is bonded to the surface of an existing part and specifies its engineering properties. For more information, see *Skin and stringer reinforcements*.
- **Inertia**. You can define lumped mass, rotary inertia, and heat capacitance at a point on a part. You can also define mass and inertia proportional damping. In an Abaqus/Standard analysis, you can define composite damping. For more information, see *Inertia*.
- **Springs/Dashpots**. You can define springs and dashpots that exhibit the same linear behavior independent of field variables. You can also define both spring and dashpot behavior on the same set of points. In an Abaqus/Explicit or an Abaqus/Standard analysis, you can model springs and dashpots that connect two points, following the line of action between the two points. In an Abaqus/Standard analysis, you can also model springs and dashpots that connect two points, acting in a fixed direction, or that connect points to ground. For more information, see *Springs and dashpots*.

Additional information

- *Creating and editing skin reinforcements*
- *Defining point mass and rotary inertia*
- *Defining heat capacitance*
- *Creating springs and dashpots connecting two points*
- *Creating springs and dashpots connecting points to ground*

Using the Query toolset to obtain assignment information

You can use the Query toolset to display the following:

- A list of all of the regions to which you have assigned sections.
- The name of a section assigned to a selected region.
- Information regarding the regions that require section assignments.
- The beam orientations assigned to selected wire regions.
- The material orientations assigned to selected shell and solid regions.
- A graphic representation of the plies in a composite layup or a composite section.
- The rebar reference orientations assigned to selected shell regions.
- The direction of shell/membrane normals assigned to all shell and axisymmetric wire regions.
- The direction of beam/truss tangents assigned to all wire regions.
- The composite layups and plies that contain disjoint regions.

1. From the main menu bar, select **Tools->Query**.

   Tip: You can also query the model by clicking the tool in the Query toolset.

   Abaqus/CAE displays the **Query** dialog box.

   You can request either a general query or a module-specific query. The **Shell element normals** and **Beam element tangents** queries are general queries. For a discussion of the information displayed by general queries, see *Obtaining general information about the model*. The **Section assignments**, **Regions missing sections**, **Beam orientations**, **Material orientations**, **Rebar orientations**, **Ply stack plot**, and **Disjoint ply regions** queries are specific to the Property module.

2. From the **Property Module Queries** list, select the property of interest.
3. Select the region that you want to query in the viewport.

   Tip: You can limit the types of objects that you can select in the viewport by clicking the selection filter tool in the prompt area and then clicking the selection filter of your choice in the dialog box that appears. See *Using the selection options*, for more information.

4. Once you have selected the region that you want to query, the following information appears:

Section assignment queries

Abaqus/CAE displays the name of the section or sections assigned to the selected region in the message area. If you query a region with a section assignment that has been suppressed, Abaqus/CAE reports no section assignments for that region.

Regions missing sections

If any regions of your part require a section assignment and do not have one, Abaqus/CAE highlights these regions in the viewport and prompts you to save the regions as a set. If you want to save these regions as a named set, toggle on **Save regions in a set** from the dialog box that appears; and, if desired, customize the default set name.

Beam section orientation queries

Abaqus/CAE displays the name of the beam orientation assigned to the selected beam region in the message area. In addition, the $\mathbf{n}_1$-direction for each beam region in the part appears in the message area.

Material orientation queries

Abaqus/CAE displays the type of the material orientation assigned to the selected region in the message area. For the `GLOBAL`, `SYSTEM`, and `DISCRETE` types, Abaqus/CAE displays the material orientation triads in the viewport. In addition, information concerning the material orientation of each region in the part appears in the message area.

Rebar orientation queries

Abaqus/CAE displays the name of the rebar orientation assigned to the selected region in the message area. In addition, information concerning the rebar reference orientation of each region in the part appears in the message area.

Ply stack plot

Abaqus/CAE creates a new viewport and displays a graphical representation of a core sample through a composite layup or a composite section. The image shows the plies in the layup along with details of each ply, such as its fiber orientation, thickness, reference plane, and integration points. For more information, see *Viewing a ply stack plot*."

Disjoint ply regions

Abaqus/CAE displays in the message area the names of the composite layups and the plies within them that contain disjoint regions.

5. To exit the querying procedure, click the cancel button  in the prompt area.

Additional information

- *Creating and editing sections*
- *Understanding the role of the Query toolset*

The Assembly module

You use the Assembly module to create and modify the assembly.
A model contains one main assembly, which is composed of instances of parts from the model as well as instances of other models. The tutorial in *Using Additional Techniques to Create and Analyze a Model in Abaqus/CAE* contains examples of how you use the Assembly module to create part instances and position them relative to each other in a global coordinate system.

In this section:

- *Understanding the role of the Assembly module*
- *Entering and exiting the Assembly module*
- *Working with part instances*
- *Working with model instances*
- *Creating the assembly*
- *Creating patterns of instances*
- *Performing Boolean operations on part instances*
- *Understanding toolsets in the Assembly module*
- *Using the Assembly module toolbox*
- *Creating and manipulating part and model instances*
- *Applying constraints to part and model instances*
- *Using the Query toolset to query the assembly*

Understanding the role of the Assembly module

When you create a part, it exists in its own coordinate system, independent of other parts in the model. In contrast, you use the Assembly module to create instances of your parts and to position the instances relative to each other in a global coordinate system, thus creating the assembly. You position part instances by sequentially applying position constraints that align selected faces, edges, or vertices or by applying simple translations and rotations.

You can also create instances of other models in your main model, allowing you to add complete subassemblies in addition to individual parts. Model instances are created in the exact same way as part instances and can be positioned and manipulated in a similar fashion.

An instance maintains its association with the original part or model. If the geometry of a part or model changes, Abaqus/CAE automatically updates all instances of the part or model to reflect these changes. You cannot edit the geometry of an instance directly.

Your main model can contain many parts and model subassemblies, and a part or model can be instanced many times in the main model assembly; however, a model contains only one top-level assembly. Loads, boundary conditions, predefined fields, and meshes are all applied to the complete assembly. Even if your model consists of only a single part, you must still create an assembly that consists of just a single instance of that part.

A part instance can be thought of as a representation of the original part. You can create either independent or dependent part instances. An independent instance is effectively a copy of the part. A dependent instance is only a pointer to the part, partition, or virtual topology; and as a result, you cannot mesh a dependent instance. However, you can mesh the original part from which the instance was derived, in which case Abaqus/CAE applies the same mesh to each dependent instance of the part.

A model instance is always dependent, not independent.

Entering and exiting the Assembly module

You can enter the Assembly module at any time during an Abaqus/CAE session by clicking **Assembly** in the **Module** list located in the context bar. The **Instance**, **Constraint**, **Feature**, and **Tools** menus appear on the main menu bar.

To exit the Assembly module, select any other module from the **Module** list. You need not save your assembly before exiting the module; it will be saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

Working with part instances

This section describes part instances, how they relate to the original part, how you link and exclude part instances, and how you use them to create the assembly.

In this section:

- *Understanding the relationship between models, parts, instances, and assemblies*
- *What is the difference between a dependent and an independent part instance?*
- *How do I decide whether to create a dependent or an independent part instance?*
- *Changing from a dependent to an independent part instance or vice versa*
- *Linking part instances between models*
- *Excluding part instances from an analysis*
- *Sets and part instances*

Understanding the relationship between models, parts, instances, and assemblies

A model can contain many parts; however, it can contain only one top-level assembly. The assembly is composed of instances of the parts positioned relative to each other in a global coordinate system, as described in *What is a part instance?*. The top-level assembly can also contain model instances that effectively create subassemblies from other models.

The concept of parts, part instances, and the assembly is carried throughout the Abaqus/CAE modeling process:

1. You create a part in the Part module; each part is a distinct entity that can be modified and manipulated independently of other parts. Parts exist in their own coordinate system and have no knowledge of other parts.
2. You define section properties in the Property module and also associate a material with a section. You use the Property module to assign these section properties to a part or to a selected region of a part.
3. You create instances of your parts in the Assembly module, and you position those instances relative to each other in a global coordinate system to form the assembly. You can also add instances of other models in the assembly.

   Abaqus/CAE allows you to create either independent or dependent part instances, as described in *What is the difference between a dependent and an independent part instance?*. Both independent and dependent part instances maintain their association with the original part. When you modify the original part in the Part module, Abaqus/CAE updates any instances of that part when you return to the Assembly module. You can instance a part many times and assemble multiple instances of the same part. Each instance of the part is associated with the section properties assigned to the part in the Property module.
4. You use the Interaction and Load modules to complete the definition of the model by, for example, defining contact and applying items such as loads and boundary conditions. The Interaction and Load modules operate on the assembly.
5. You use the Mesh module to mesh the assembly. You can do either of the following:

   - Individually mesh each independent instance of a part in the assembly.
   - Mesh the original part. Abaqus/CAE then associates the mesh with each dependent instance of the part in the assembly.

   The two meshing approaches are described in *What is the difference between a dependent and an independent part instance?*.

Creating a part or model instance, contains detailed instructions on creating part instances.

What is the difference between a dependent and an independent part instance?

When you create a part instance, you can choose to create either a dependent part instance or an independent part instance. You can also edit a part instance and change it from dependent to independent or vice versa. When you create a model instance, it is always dependent.

Dependent part instances

By default, Abaqus/CAE creates a dependent instance of a part. A dependent instance is only a pointer to the original part. In effect, a dependent instance shares the geometry and the mesh of the original part. As a result, you can mesh the original part, but you cannot mesh a dependent instance. When you mesh the original part, Abaqus/CAE applies the same mesh to all dependent instances of the part. Most modifications are not allowed on a dependent part instance; for example, you cannot add partitions or create virtual topology. However, operations that do not modify the geometry of a dependent part instance are still allowed; for example, you can create sets, apply loads and boundary conditions, and define connector section assignments. If you have already meshed a part or added virtual topology to the part, you can create only a dependent instance of the part.

If you apply an adaptive remeshing rule to a dependent part instance in the Mesh module, Abaqus/CAE remeshes the original part and applies the new mesh to each dependent instance of the part.

You cannot change the mesh attributes of an individual dependent part instance; for example, the mesh seeds, mesh controls, and element types. However, you can change the mesh attributes of the original part, and Abaqus/CAE propagates the changes to all dependent instances of the part. Although you have already meshed the original part and applied the same mesh to its dependent instances, the mesh is visible only in the Mesh module. You continue to work with the native Abaqus/CAE geometry in the Assembly, Interaction, and Load modules. In general, you cannot use the Edit Mesh toolset to edit the mesh of a dependent part instance; however, you can use the Edit Mesh toolset to edit and project the nodes of a dependent part instance. Abaqus/CAE moves the nodes of the original meshed part, and your modifications appear on all dependent instances of the part.

The advantages of dependent part instances are that they consume fewer memory resources and you need mesh the part only once. In addition, Abaqus/CAE instances a dependent part instance in the input file by writing a single set of nodal coordinates and element connectivity to define the part along with a transform to define each part instance.

Independent part instances

In contrast, an independent part instance is a copy of the geometry of the original part. You cannot mesh a part from which you created an independent part instance; however, you can mesh the independent instance. In addition to meshing, you can perform most other operations on an independent instance; for example, you can add partitions and create virtual topology. The disadvantages of independent instances are that they consume more memory resources, and you must mesh each independent instance individually. In addition, Abaqus/CAE does not take advantage of instantiation in the input file with independent part instances—sets of nodal coordinates and element connectivity are written to the input file for each independent part instance.

You cannot create both a dependent and an independent instance of the same part. As a result, if you create a dependent instance of a part, all subsequent instances must be dependent. The same argument applies to independent instances. Instances of mesh parts are always dependent.

You can use the Model Tree to determine if an instance is dependent or independent. When you mesh an independent part instance, the mesh appears in the Model Tree under the part instance container, as shown in *Figure 1*. In addition, *Figure 1* also illustrates that as you move the cursor over an instance, the information displayed by the Model Tree indicates whether the instance is dependent or independent.

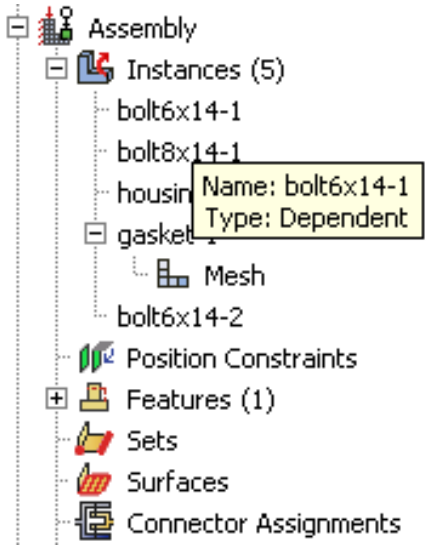


Figure 1: The Model Tree indicates whether a part instance is dependent or independent.

How do I decide whether to create a dependent or an independent part instance?

If your assembly contains a few part instances that are unrelated, dependent instances have little advantage over independent instances. Each part is different, and you must create an instance of each part. In contrast, if your assembly contains identical part instances, you can save time by assembling dependent instances of the part. When you subsequently mesh the original part, Abaqus/CAE applies that mesh to each dependent instance of the part in the assembly. In addition, dependent instances consume fewer memory resources and result in a smaller input file.

For example, *Figure 1* illustrates an assembly of independent and dependent part instances. The pump housing is an independent part instance, and the eight bolts are dependent part instances. The figure on the left shows the assembly in the Assembly module. The figure on the right shows the assembly in the Mesh module. The user has meshed the part representing the bolt, and Abaqus/CAE associated the mesh with each dependent instance of the bolt.

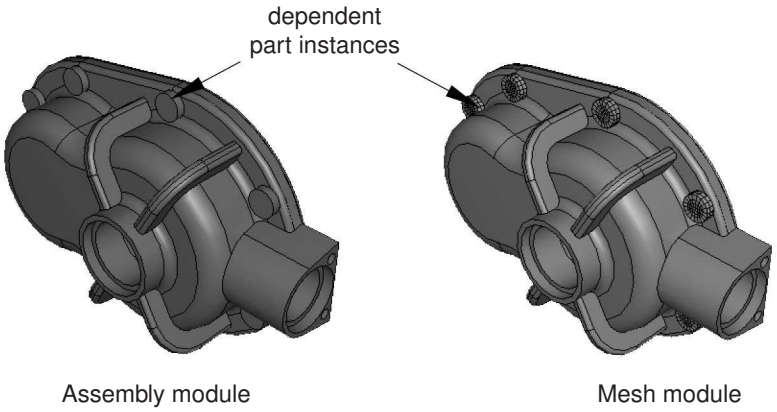


Figure 1: Dependent part instances in the Assembly and Mesh modules.

You will find it more convenient to use dependent part instances when you use the linear or radial pattern tool to create a pattern of identical instances. When you mesh the original part, Abaqus/CAE applies the same mesh to each dependent instance in the pattern. In contrast, if you create a pattern of independent instances, you must mesh each instance individually.

Abaqus/CAE creates dependent instances by default. Unless your assembly contains only a few parts, it is recommended that you work with dependent instances because of the memory savings and the resulting performance gain.

Changing from a dependent to an independent part instance or vice versa

The restrictions on dependent part instances may limit your ability to partition or mesh the assembly, or you may find that you wish to apply virtual topology to an instance. To switch between making a part instance dependent or independent, you can click mouse button 3 on the instance in the Model Tree and select **Make Dependent** or **Make Independent** from the menu that appears.

If you mesh a part and create a dependent instance of the part, Abaqus/CAE associates the mesh with the instance. If you subsequently change the instance from dependent to independent, Abaqus/CAE continues to associate the mesh with the independent instance. However, the reverse is not true. If you create an independent instance, mesh the instance, and subsequently convert the instance to dependent, Abaqus/CAE deletes the mesh from the dependent instance. The same applies to partitions and virtual topology. Abaqus/CAE deletes any partitions or virtual topology applied to an independent part instance when you change it to dependent.

In some cases, you can work around the restrictions on a dependent part instance by creating a copy of the original part and by creating an independent instance of the copy. You can then partition or mesh the new instance or apply virtual topology to it. Similarly, although you cannot create both a dependent and independent instance of the same part, you can create a copy of the part and create either type of instance from the copy.

Linking part instances between models

You can link part instances between models. Linking part instances allows instances and parts to be updated automatically when you modify the instance or part in the original model.

In the Model Tree, select the part instances that you want to link (child instances) to part instances in another model. Click mouse button 3, select **Link Instances**, and specify the parent model and part instances to which you want to link each child instance. Similarly, you can unlink part instances that were previously linked. For detailed instructions, see *Using the Model Tree to manipulate part instances*.

If you select all instances of a part to be linked, the part is also linked automatically. The part and its features, sets, and surfaces are updated using the parent part. Assembly-level features and sets and surfaces are not copied. Instances are updated using the parent instances and retain sets and surfaces defined on them.

If you select only some of the instances of a part to be linked, a new part is created (with `–LinkedCopy` appended to the part name) before linking the instance and the new part to the parent model.

Linked part instances and parts are not editable. The position of the linked child instance is solely determined from the position of the parent instance and cannot be updated.

By default, linked part instances and parts are colored gray in the viewport. Icons are displayed in the Model Tree to indicate the linked status of parts and part instances and to indicate the linked and excluded status of part instances if the part instances are also excluded from the analysis, as shown in *Figure 1*. `Beam-1` is a linked part instance, and `Beam-2` is a linked and excluded part instance. For more information, see *Excluding part instances from an analysis*.

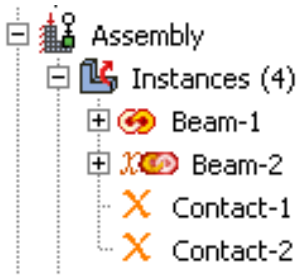


Figure 1: Model Tree icons indicating linked and excluded status of part instances.

Additional information

- *Using the Model Tree to manipulate part instances*

Excluding part instances from an analysis

You can exclude part instances from the analysis so that they are not written to the input file when the analysis job is submitted. An excluded part instance participates in all operations other than the analysis.

In the Model Tree, select the part instances that you want to exclude from the analysis. Click mouse button 3, and select **Exclude from Simulation**. Similarly, you can include part instances that were previously excluded by selecting **Include in Simulation**. Constraints on the part instances are retained if you exclude the instances from the analysis and subsequently include them.

By default, part instances that are excluded from the analysis are colored dark gray in the viewport. Icons are displayed in the Model Tree to indicate the excluded status of part instances and to indicate the linked and excluded status of part instances if the part instances are also linked between models, as shown in *Figure 1*. `Contact-1` and `Contact-2` are part instances that are excluded from the analysis, and `Beam-2` is a linked and excluded part instance. For more information, see *Linking part instances between models*.

Additional information

• *Using the Model Tree to manipulate part instances*

Sets and part instances

Part sets are transferred when you create a part instance from a part. For example, you might create a set from a region of a part and use the Property module to assign a section to that set. When you instance the part in the Assembly module, Abaqus creates part instance sets that refer to any part sets that you previously created. Abaqus provides read-only access to these part instance sets in assembly-related modules. You cannot access a part instance set from the **Set Manager**; however, you can select an eligible part instance set during a procedure by clicking the **Set** button and selecting the set from the **Region Selection** dialog box that appears. For more information, see *Understanding sets and surfaces*.

Working with model instances

You can create instances of other models in your main model, allowing you to add complete subassemblies in addition to individual parts.
Model instances are created in the exact same way as part instances and can be positioned and manipulated in a similar fashion.

When you create a new model instance, the main assembly of the referenced model is instantiated in the assembly of the current working model. The instance produces a subassembly from the contents of the other model. Since the referenced model assembly might in turn contain other model instances as children, multiple levels of complex subassemblies are possible.

You must include the external model to be instantiated in the current model database (`.cae`) file to be available. If the model you want to instantiate is contained in a different model database, use **File->Import->Model** to import it into the current model database. A model database file can always contain multiple models.

Renaming model instances is not supported in Abaqus/CAE. You should take care when renaming models because Abaqus/CAE does not update the model instance name in the instantiated models.

Characteristics of model instances

Model instances have the following characteristics:

- A particular model can be instantiated multiple times, and you can instantiate as many different models as desired.
- Model instances are always dependent, not independent.
- You can freely mix model instances with part instances.
- Model instance subassemblies can contain either geometric parts or orphan mesh parts.
- Model instances can be positioned and oriented in the main assembly by using transformations (**Translate**, **Translate To**, **Rotate**) and positioning constraints. The transformations and constraints must be applied to a complete model instance subassembly, not to any of its children. If you select a child instance within a model instance, the transformation or constraint will be applied to the entire parent model instance.
- Linear and radial patterns are not supported and cannot be used with model instances.
- Part instance commands such as **Suppress/Resume**, **Hide/Show**, **Delete**, **Show Parents/Children**, and **Switch Context** can also be used on model instances.

  You cannot use the **Suppress** and **Delete** commands on the child instances of a model instance; you can use them only on the model instance itself. If you suppress a child instance (part or model) in the original (referenced) model assembly, it is also suppressed in the main model. To see that the suppressed instance is correctly suppressed in the main model, you must use the **Model** list in the context bar to switch from the original (referenced) model to the main model. Moving to the main model in the Model Tree will not regenerate the model instance children consistently. (For information about the context bar, see *Components of the main window*.) The child instance must then be resumed in the original model.
- **Replace**, **Exclude from Simulation**, **Merge/Cut**, and **Link Instances** are not supported and cannot be used on model instances.
- The Partition toolset is not supported and cannot be used with model instances.
- The Query toolset is supported and can be used to determine the position and attributes of model instances.
- All sets and surfaces defined in the referenced model are brought into the model instance, maintaining the Model Tree hierarchy of features. These sets and surfaces will be available in the main model.
- Surface-to-surface contact and self-contact interactions defined in the initial step (along with their contact interaction properties) are the only history-level features defined in the referenced model that are brought into the model instance (although they are not visible in the model tree); other history-level features (such as steps, loads, boundary conditions, other interactions, and amplitudes) are not brought into the model instance. Some model-level features (fasteners and other engineering features) defined in the referenced model are not brought into the model instance.

Interactions in the main model depend on the interaction option used. Interaction copies to the main model are restricted based on the following patterns:

- For Abaqus/Explicit model instances, you cannot copy to any main model type.
- For Abaqus/Standard or unknown model type instances, you can copy only to an Abaqus/Standard or an unknown usage main model type.

- Model instances are supported and selectable in **Display Groups** and in the **Instance** tab of the **Assembly Display Options**.
- The Virtual Topology toolset is not supported for model instances.

Any part-level attributes that are needed in your subassembly (referenced) model must be created and assigned in that original model and cannot be created in the main model assembly. For example, materials, sections, orientations, and skin/stringer assignments must be created in the original model. Meshing can be performed on the original independent part instances, and the meshes will appear in the model instance.

When you create a model instance, all the part instances of the referenced model assembly are added to the main model assembly as child part instances. Any suppressed part instances or instances that are excluded from the simulation will retain the same status in the subassembly.

If you modify or delete an existing part instance or model instance subassembly in the main model assembly, Abaqus/CAE automatically regenerates the child instances from all parent instances (parts and models) whenever you switch out of and back into the Assembly module of the main model.

If you try to create a new model instance from another model that in turn contains child model instances, any problems with model referencing circularity will be prevented by Abaqus/CAE.

Abaqus/CAE ensures consistency of the modeling space for model instances; if all instances in the main model are three-dimensional, any other models to be instantiated must also be three-dimensional.

Model instance data saved in input files

When Abaqus/CAE generates the input (`.inp`) file for a model assembly that contains model instances, a single flattened assembly is generated. All model instance subassemblies are written as a flat list of instances under the single assembly block.

Most features from the original model of a model instance are saved in the input file, with some exceptions:

- Surface-to-surface contact and self-contact interactions defined in the initial step are the only history-level features from a model instance subassembly that are saved in the input file. The contact interaction property name and surface names are prepended with the model instance name in the main assembly; for example:

  model-instance-name`#`*contact-property-name*
  model-instance-name`#Surf-1, `*model-instance-name*`#Surf-2`

- Model-level features from a model instance are saved in the input file; for example, materials, section assignments, connector section assignments, skins, stringers, and orientations. Materials and element controls defined in a model instance are prepended with the model name in the main assembly; for example,

  model-name`#`*material-name*

  Connector sections assignments are prepended with the model instance name in the main assembly; for example:

  model-instance-name`#Wire-3-Set-1`

  Other model-level data such as initial conditions and amplitude definitions from a model instance are not saved in the input file.
- Engineering features such as mass and inertia elements, springs, and dashpots defined at the part level in the model instance are saved to the input file, but engineering features defined at the assembly level in the original model are not.

- Sets and surfaces from a model instance are saved in the input file. These set and surface names are also prepended with the model instance name in the main assembly; for example:

  model-instance-name`#Set-1`

- Constraints, reference points, attachment points, attachment lines, and wires from a model instance are saved in the input file.
- For constraints the model instance name will be prepended to the constraint name; for example:

  model-instance-name`#`*constraint-name*

- Attachment points, attachment lines, and wires will be available through sets created in the subassembly.

The following limitations exist:

- Restart analysis is not supported for a model containing model instances.
- Instances of models containing assembled fasteners are not supported.

Creating the assembly

After you create a part instance or a model instance, you apply a succession of position constraints and positioning operations to position it relative to other instances in the global coordinate system. This section describes the tools that Abaqus/CAE provides to position and constrain part and model instances. This section also describes how you can replace a part instance.

In this section:

- *The position tools in the Assembly module*
- *How the position constraint methods differ*
- *How conflicts can arise between position constraints, translations, and rotations*
- *Positioning a part or model instance using the Translate To tool*
- *Replacing an instance*

The position tools in the Assembly module

Each part exists in its own coordinate system in the Part module, and model instances are created in their own coordinate system. You use the Assembly module to position and orient instances of these parts and models relative to each other in a global coordinate system. Abaqus/CAE provides the following tools for positioning part and model instances:

Auto-offset

When you create the first part or model instance in the Assembly module, Abaqus/CAE displays a triad indicating the origin and the orientation of the global coordinate system. Abaqus/CAE positions the first instance so that the origin of the part or model aligns with the origin of the global coordinate system and the axes are aligned. If you create additional instances, Abaqus/CAE continues to position the new instances such that their coordinate system aligns with the global coordinate system. Since this usually results in new instances overlapping existing ones, Abaqus/CAE allows you to apply an offset before it creates the instance. The offset is applied along the *X*-axis for three-dimensional and two-dimensional instances and along the *Y*-axis for axisymmetric instances.

Basic positioning tools

Abaqus/CAE provides the following basic methods for positioning part and model instances:

- You can translate selected instances along a vector by specifying the coordinates of the start point and end point of the translation vector. You can use the following methods to determine the distance moved by the selected instances:
  - The selected instances move along the translation vector from the start point to the end point.
  - The selected instances move along the translation vector from the start point toward the end point and continue to move until a selected face or edge is a specified distance from a face or edge selected from the fixed instances. For more information, see *Positioning a part or model instance using the Translate To tool*.
- You can rotate selected instances about an axis. You specify the *X*-, *Y*-, and *Z*-coordinates of the start point and end point of the axis of rotation and the angle of rotation.

Position constraint tools

A position constraint defines a relationship between two instances. Unlike a simple translation or rotation, you do not specify the position directly. Position constraints define a set of rules that must always be met by the part or model instances in the assembly; for example, a face that must be parallel to another face.

Position constraints defined in the Assembly module create constraints only on the initial positions of instances, whereas constraints defined in the Interaction module define constraints on the analysis degrees of freedom. In the Assembly module constraints are stored as features of the assembly. If you modify a part or move a part or model instance, Abaqus/CAE attempts to apply all existing position constraints when it regenerates the assembly. Each of the position constraints is described in *How the position constraint methods differ*.

Creating the final assembly is an iterative process of creating instances, applying position constraints, and applying translations and rotations. After each repositioning, Abaqus/CAE displays a temporary image indicating the result of the operation. You can accept the new position, cancel the operation, or step back through the repositioning procedure by clicking the **Previous** button  in the prompt area.

You can use the Query toolset to obtain the coordinates of a vertex and to measure the distance between selected vertices. This may help you determine the vector along which you need to translate instances or the angle through

which you need to rotate them. *Using the Query toolset to query the assembly*, contains detailed instructions on how to obtain information about the assembly.

How the position constraint methods differ

A position constraint defines a relationship between two part or model instances—one that will move (the movable instance) and one that will remain stationary (the fixed instance). When you apply a position constraint, Abaqus/CAE computes a position for the movable instance that satisfies this relationship; you do not specify the position directly. You can apply the following position constraints to instances in the Assembly module:

- Parallel face (three-dimensional instances only)
- Face to face (three-dimensional instances only)
- Parallel edge
- Edge to edge
- Coaxial (three-dimensional instances only)
- Coincident point
- Parallel coordinate systems

In general, applying a single position constraint is not sufficient to define the precise location of a movable instance. You must apply several position constraints—usually three for a three-dimensional assembly and two for a two-dimensional assembly—to position an instance in the desired location. Part and model instances can overlap as a result of applying position constraints; Abaqus/CAE does not prevent overclosure between edges, faces, or cells. Similarly, Abaqus/CAE does not prevent you from overconstraining instances or duplicating a constraint.

The definition of a constraint feature includes all the faces and edges that you originally selected. If you subsequently modify a part or move a part or model instance, Abaqus/CAE automatically recalculates the constraint based on your original selection of faces and edges. As a result, one or more instances might move after the assembly is regenerated. For example, different edges might become parallel. For more information on features, see *Manipulating features in the Assembly module* and *The Feature Manipulation toolset*.

The following position constraints are provided by the Assembly module:

Parallel Face

A parallel face position constraint causes a selected face of the movable instance to become parallel with a selected face of the fixed instance. However, the position constraint does not specify the precise location of the movable instance, and the distance between the parallel faces is arbitrary. To apply a parallel face position constraint between two instances, you do the following:

- Select the faces to be constrained to be parallel from the movable instance and the fixed instance, as shown in *Figure 1*.

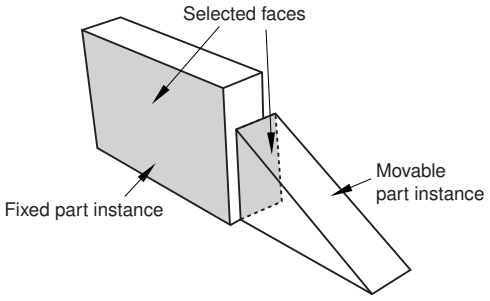


Figure 1: Select the faces to become parallel.

- Abaqus/CAE displays arrows normal to the selected faces. You prescribe the orientation of the movable instance by selecting the direction of the arrow normal to its selected face. *Figure 2* illustrates the result of applying the position constraint and the effect on the movable instance of reversing the direction of the arrow.

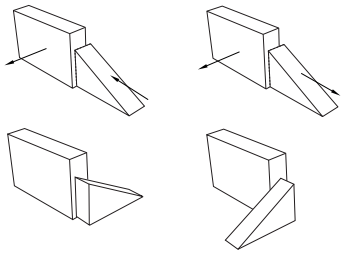


Figure 2: The result of applying a parallel face position constraint and the effect of changing the direction of the arrow normal to the selected face of the movable instance.

Abaqus/CAE rotates the movable instance until the two selected faces are parallel and the arrows are pointing in the same direction.

The faces you select from the movable and fixed instances must be planar. The parallel face position constraint can be applied only to three-dimensional instances.

Face to Face

A face-to-face position constraint is similar to a parallel face position constraint except that you define the clearance between the parallel faces. The clearance is measured between the two selected faces, positive along the normal to the fixed instance. Other than this clearance, the precise location of the movable instance is not constrained. Assuming that you selected the same two faces shown in *Figure 1*, the effect of applying a face-to-face constraint is shown in *Figure 3*. *Figure 3* also illustrates the effect on the movable instance of reversing the direction of the arrow normal to its selected face.

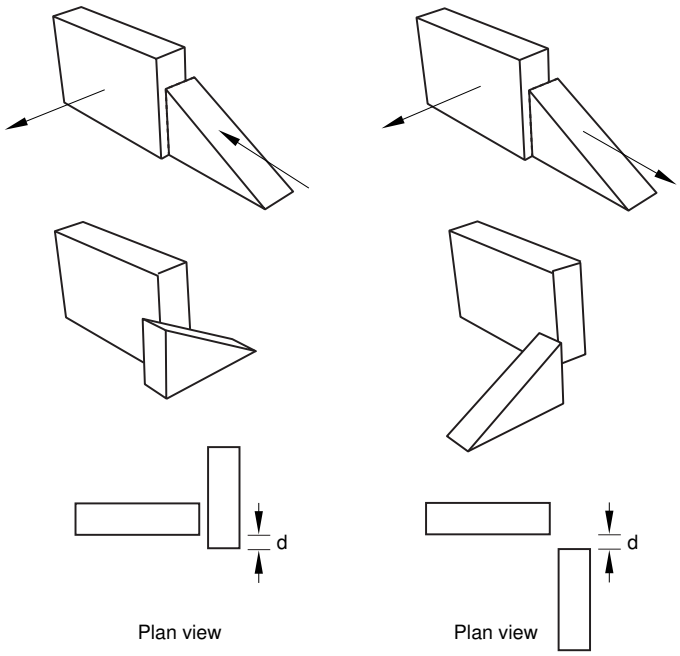


Figure 3: The result of applying a face-to-face constraint and the effect of changing the direction of the arrow normal to the selected face of the movable instance.

Abaqus/CAE rotates the movable instance until the two selected faces are parallel and the arrows point in the same direction. In addition, the movable instance is translated to satisfy the clearance specified. The faces you select from the movable and fixed instances must be planar. The face-to-face position constraint can be applied only to three-dimensional instances.

Parallel Edge

A parallel edge position constraint causes a selected edge of the movable instance to become parallel with a selected edge of the fixed instance. However, the position constraint does not specify the precise location of the movable instance, and the distance between the parallel edges is arbitrary. To apply a parallel edge position constraint between two instances, you do the following:

- Select the edges to be constrained to be parallel from the movable and fixed instance, as shown in *Figure 4*.

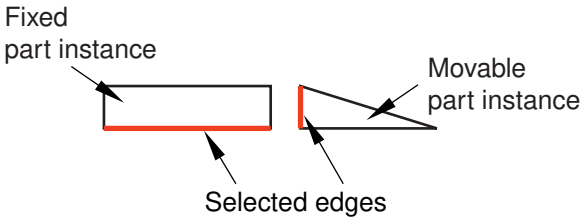


Figure 4: Select the edges to become parallel.

- Abaqus/CAE displays arrows along the selected edges. You prescribe the orientation of the movable instance by selecting the direction of the arrow along its selected edge. *Figure 5* illustrates the result of applying the position constraint and the effect on the movable instance of reversing the direction of the arrow.

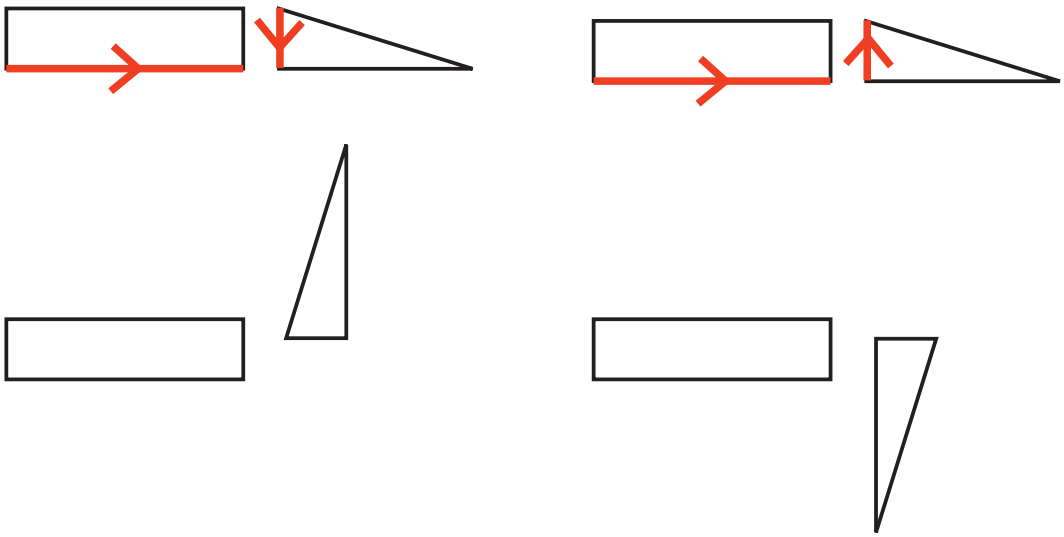


Figure 5: The result of applying a parallel edge constraint and the effect of changing the direction of the arrow along the selected edge of the movable instance.

Abaqus/CAE rotates the movable instance until the two selected edges are parallel and the arrows point in the same direction.

The edges you select from the movable and fixed instances must be straight. You can select an edge from an instance, or you can select a datum axis or one of the axes of a datum coordinate system. The parallel edge position constraint can be applied only to two-dimensional and three-dimensional instances. It has no effect on axisymmetric instances.

Edge to Edge

An edge-to-edge position constraint is similar to a parallel edge position constraint except that the clearance between the parallel edges is defined by the constraint. Assuming that you selected the same two edges shown in *Figure 4*, the effect of applying an edge-to edge position constraint to a two-dimensional assembly is shown in *Figure 6*. *Figure 6* also illustrates the effect on the movable instance of reversing the direction of the arrow along its selected edge.

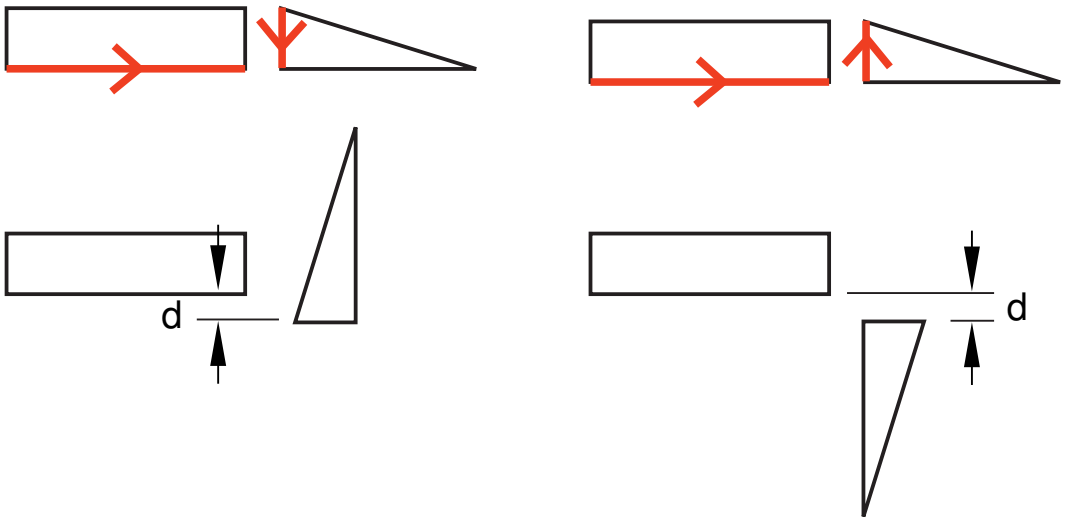


Figure 6: The result of applying an edge-to-edge constraint and the effect of changing the direction of the arrow along the selected edge of the movable instance.

The modeling space of the assembly determines the behavior of Abaqus/CAE after you apply an edge-to-edge position constraint.

- If the assembly is three-dimensional, Abaqus/CAE positions the movable instance so that the edges are coincident.
- If the assembly is two-dimensional, you can specify the clearance between the selected edges. The clearance is measured between the two selected edges, positive along the normal to the fixed instance.

Other than this behavior, the precise location of the movable instance is not constrained. The edge-to-edge position constraint can be applied to two-dimensional, three-dimensional, and axisymmetric instances.

Coaxial

A coaxial position constraint causes a selected cylindrical or conical face of the movable instance to become coaxial with a selected cylindrical or conical face of the fixed instance. However, the coaxial position constraint does not constrain the precise location of the movable instance. To apply a coaxial position constraint between two instances, you do the following:

- Select the cylindrical or conical faces to be constrained to be coaxial from the movable and fixed instance, as shown in *Figure 7*.

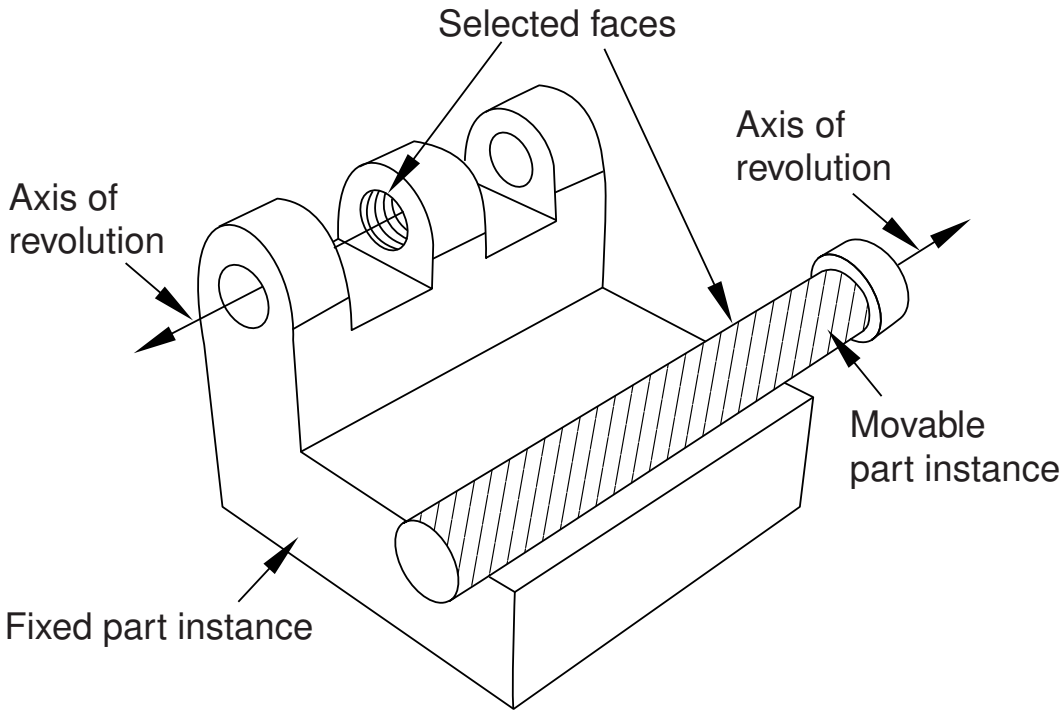


Figure 7: Select the faces to become coaxial.

- Abaqus/CAE displays arrows along the axis of revolution of the selected instances. You prescribe the orientation of the movable instance by selecting the direction of the arrow along its axis of revolution. *Figure 8* illustrates the result of applying the coaxial position constraint.

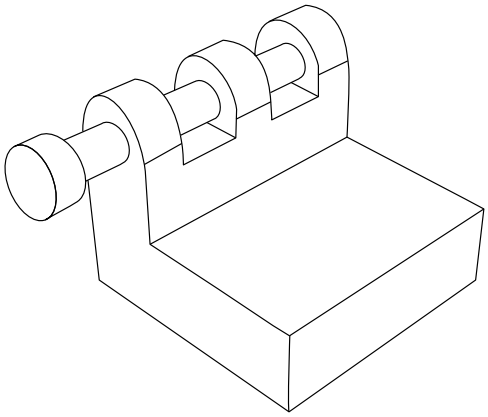


Figure 8: The effect of applying a coaxial constraint.

Abaqus/CAE rotates and translates the movable instance until the two selected faces are coaxial and the arrows are pointing in the same direction. The coaxial position constraint can be applied only to three-dimensional instances.

Coincident Point

A coincident point constraint causes a selected point on the movable instance to coincide with a selected point on the fixed instance. However, the coincident point constraint does not constrain the orientation of the movable

instance. The orientation of the movable instance does not change after the constraint is applied, as shown in *Figure 9*. For detailed instructions, see *Constraining two instances with coincident points*.

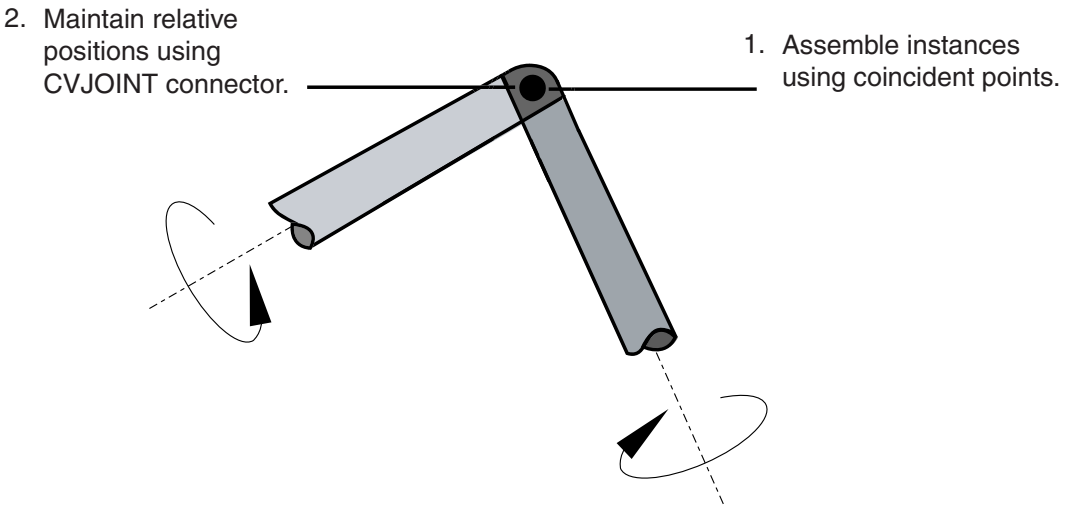


Figure 9: The effect of applying a coincident point constraint.

Parallel CSYS

A parallel coordinate systems constraint causes the axes of a datum coordinate system on the movable instance to become parallel with the axes of a datum coordinate system on the fixed instance. However, the parallel coordinate systems constraint does not specify the precise location of the movable instance. *Figure 10* illustrates the effect of applying a parallel coordinate systems constraint and a coincident point constraint to two instances.

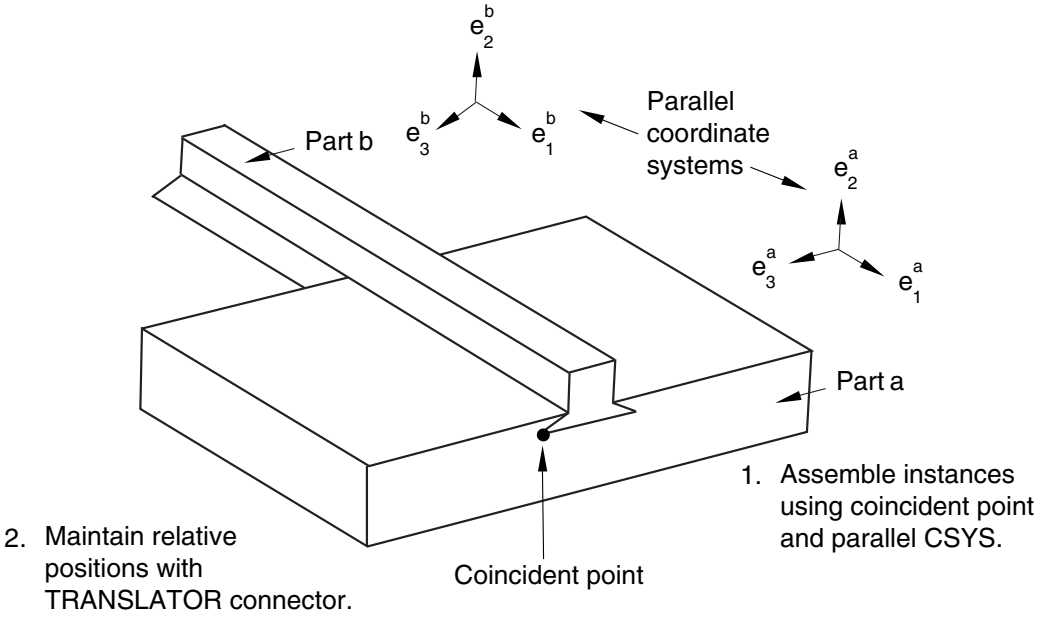


Figure 10: The effect of applying parallel coordinate systems and coincident point constraints.

The coordinate systems can be either rectangular (*X*-, *Y*-, and *Z*-axes), cylindrical (*R*-, θ-, and *Z*-axes), or spherical (*R*-, θ-, and ϕ-axes). For detailed instructions, see *Constraining two instances with parallel coordinate systems*.

You can use datums to position part and model instances. When you are prompted to select a face, you can also select a datum plane. When you are prompted to select an edge, you can also select a datum axis or one of the axes of a datum coordinate system. You can select a datum that you created in the Part module because the datum is associated with an instance of the part and moves with the part instance. However, if the position constraint uses a datum that you created in the Assembly module by selecting from a part instance (such as a face of a part instance), Abaqus/CAE changes its regeneration behavior and regenerates features in the order that you created them. For more information, see *How are position constraints regenerated?*. You cannot select a datum as the movable part instance if you created the datum in the Assembly module and it depends on more than one part instance; for example, a datum axis that runs through vertices of two part instances.

How conflicts can arise between position constraints, translations, and rotations

In some situations attempting to apply a position constraint results in a conflict with existing position constraints. If that is the case, Abaqus/CAE displays an error message, and you can either apply a different position constraint or use the Feature Manipulation toolset to modify the existing position constraints.

Similarly, attempting to translate or rotate a part or model instance may result in a conflict with existing position constraints. If a conflict occurs, Abaqus/CAE does the following:

Translation

Abaqus/CAE applies the components of translation only along the unconstrained degrees of freedom. If all of the degrees of freedom are constrained, Abaqus/CAE displays an error message and the translation fails.

Rotation

Abaqus/CAE displays an error message, and the rotation fails.

If you experience conflicts with an existing position constraint, you can remove all the existing position constraints without changing the position of the instances by using **Instance->Convert Constraints**. You can then apply the new position constraint, translation, or rotation. You cannot restore position constraints that were removed. Alternatively, you can delete a position constraint, and Abaqus/CAE will move the instance back to its original position.

Positioning a part or model instance using the Translate To tool

The **Translate To** tool positions two part or model instances by translating one instance along a user-defined vector defining the direction of motion until selected faces or edges of the movable instance are separated by a specified distance from selected faces or edges of the fixed instance.

When you use the **Translate To** tool to position instances in three-dimensional modeling space, you select faces to come into contact; for instances in two-dimensional or axisymmetric modeling space, you select edges to come into contact. In addition, when you use the **Translate To** tool to position axisymmetric instances, the translation vector must be parallel to the axis of revolution.

When you use the **Translate To** positioning tool, you can select more than one face or edge from both the fixed and the movable instances. Selecting multiple faces or edges is useful if you are not sure what part of the model will come in contact when the movable instance moves along the selected vector. However, for faster processing you should select as few faces or edges as possible.

To translate a movable part or model instance to a fixed instance, you do the following:

- Select faces or edges from the instance that will move and from the instance that will remain stationary.
- Prescribe the motion of the movable instance by defining a translation vector. *Figure 1* illustrates the selected edges and translation vector.

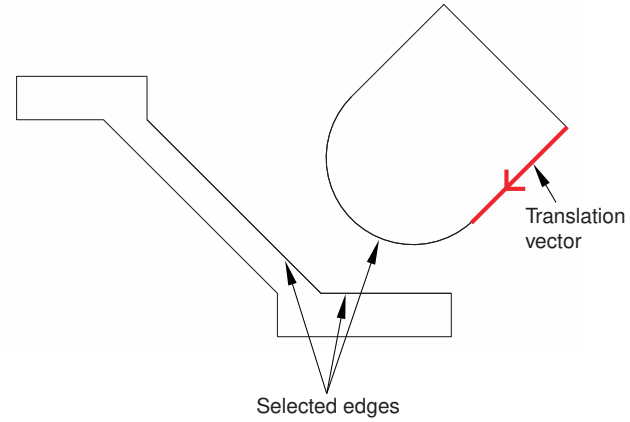


Figure 1: Select the edges to contact, and define the translation vector.

- Define the desired clearance between the selected faces or edges. *Figure 2* shows the effect of the contact constraint after specifying a clearance value of zero and a clearance value of *d*.

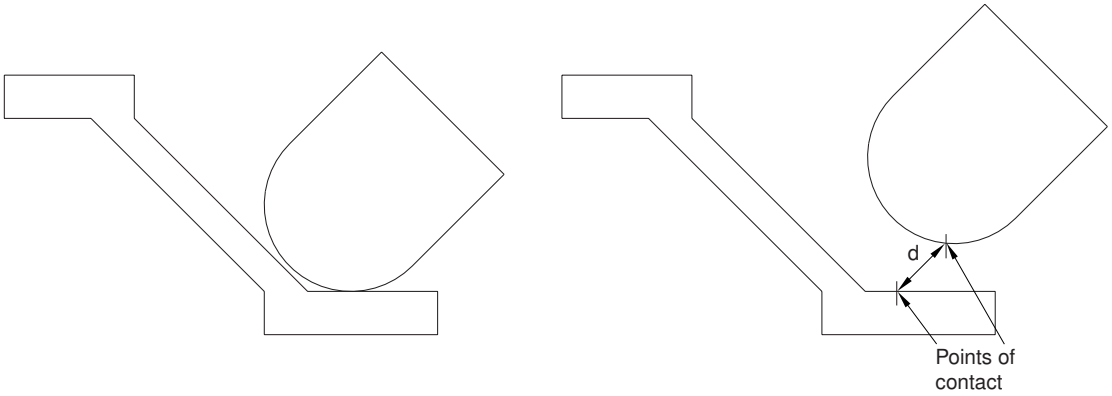


Figure 2: The effect of applying a contact constraint and specifying clearance values of zero and *d*.

To measure the clearance *d*, Abaqus/CAE first moves the instance along the translation vector until any pair of selected faces or edges come into contact. Abaqus/CAE then moves the instance along the translation vector a distance specified by the clearance value. The clearance can be zero or a positive or negative number; a negative value for the clearance results in overclosure between the selected faces or edges. When you use the **Translate To** tool, Abaqus/CAE calculates the position of the movable instance within a tolerance based on its size. If you want to avoid any possibility of overclosure, you should specify a small clearance value, rather than simply specifying zero.

Abaqus/CAE displays an error message and does not move the instance if contact between the selected faces or edges is not possible along the translation vector.

Even though you translate the movable instances until contact occurs with a fixed instance, the physical proximity of the selected surfaces is not enough to indicate any type of interaction between them. You must use the Interaction module to specify mechanical contact between surfaces. The **Translate To** positioning tool is satisfied only within a tolerance based on the size of your model. As a result, contact may not be precise unless it is applied between two planar surfaces.

Abaqus/CAE approximates a curved face with a set of faceted faces. Likewise, Abaqus/CAE approximates a curved edge with a set of faceted edges. The number of facets depends on the degree of curve refinement that you specified when creating the part in the Part module. Use the box zoom tool  to view the faceting applied to curved faces or edges in the assembly. When you are translating curved faces or curved edges, Abaqus/CAE computes the contact position using this faceted representation. You may wish to set the curve refinement to a finer setting based on the curvature of faces or edges that you know will be coming into contact. For more information, see *Controlling curve refinement*.

Replacing an instance

You can replace a part/model instance with an instance of a second part/model. To be precise, you are replacing the part/model from which the part/model instance is created. Abaqus/CAE positions the new instance such that its origin is located at the origin of the original instance and their axes align. In addition, you can choose whether the new instance inherits all the constraints from the instance it replaced.

The replace operation does not change the attributes of the instance. For example, if the original instance is dependent, the instance that replaces it will also be dependent. As a result, if an independent instance of a part exists, you cannot use the replace procedure to create a dependent instance of the same part.

Replacing an instance is useful when you are replacing an instance with one that has similar geometry. For example, the new instance might have additional detail that was not present in the original instance. You can also replace a geometry-based part with a mesh representation of the same part. For example, you could replace a part with the mesh representation of the deformed part imported from an output database.

Creating patterns of instances

You can create multiple copies of a selected instance in either a linear or radial pattern. You can specify the number of instances to create and the structure of the pattern, as described below.

Linear pattern

A linear pattern positions the new instances linearly along a direction; for example, the *X*-direction. The origin of the selected instance and the origins of the new instances lie on the line specified by the direction. You can specify the number of instances and the spacing between the instances. In addition, you can change the orientation of the linear pattern by selecting a line from the assembly that represents the new direction.

You can create a matrix of copied instances by creating copies in a second direction; for example, the *Y*-direction. The options are the same as for the first direction; you can control the number of copies, the spacing, and the orientation. By default, the first direction is the *X*-axis and the second direction is the *Y*-axis. For example, *Figure 1* illustrates how a part instance can be patterned in both the *X*- and *Y*-axes.

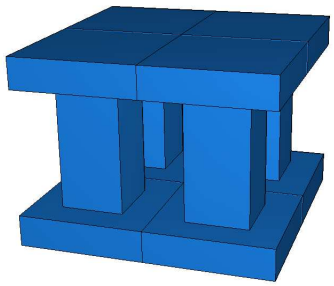


Figure 1: Instances patterned in two linear directions.

Radial pattern

A radial pattern positions the new instances in a circular pattern. You can specify the number of instances, and you can specify the angle between the first and last copy, where a positive angle corresponds to a counterclockwise direction. For example, *Figure 2* illustrates a radial pattern of the same instances that appear in *Figure 1*.

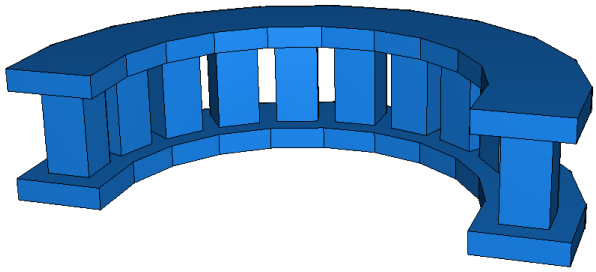


Figure 2: A radial pattern of instances.

By default, Abaqus/CAE creates the radial pattern about the *Z*-axis. Alternatively, you can select a line from the assembly that defines the axis of the circular pattern.

If you create a pattern of instances that are touching and you want to treat the pattern as a single part, you must use the **Merge/Cut** tool to merge all of the part instances in the pattern into a single part instance. For example, the instances

in the radial pattern illustrated in *Figure 2* overlapped each other and have been merged into a single part instance. For more information, see *Performing Boolean operations on part instances*. If you do not merge the part instances, the pattern may include duplicate faces or nodes where the instances touch.

If a part contains part-level sets or surfaces, Abaqus/CAE creates separate assembly-level sets and surfaces for each individual instance in a pattern (see *How do part sets and assembly sets differ?*, for further discussion of part- and assembly-level sets and surfaces). For example, if the top face of the original part in *Figure 1* and *Figure 2* is included in a part-level surface, Abaqus/CAE initially creates individual assembly-level surfaces for the top face of each instance in the patterned assembly. It is often helpful to merge these repeated sets and surfaces into a single set or surface. When you merge patterned part instances, Abaqus/CAE also merges any repeated sets or repeated surfaces into a single set or surface on the merged part and part instance. If you do not merge the patterned part instances, you can still merge sets or surfaces using the **Boolean** option in the Model Tree (see *Performing Boolean operations on sets or surfaces*, for instructions).

You will find it more convenient to use dependent instances when you create a linear or radial pattern of instances. When you mesh the original part, Abaqus/CAE applies the same mesh to each instance in the pattern. In contrast, if you create a pattern of independent instances, you must mesh each instance individually. For more information, see *What is the difference between a dependent and an independent part instance?*.

Performing Boolean operations on part instances

This section describes how you merge and cut part instances.

You can select instances of parts that you created using Abaqus/CAE and merge them into a single instance.In addition, you can cut away the geometric portion of a part instance using the geometric portion of other part instances to make the cut. You can also merge instances of parts containing both geometry and orphan elements.

In this section:

- *Merging and cutting part instances*
- *Merging and cutting independent and dependent part instances*

Merging and cutting part instances

Select **Instance**->**Merge/Cut** from the main menu bar to merge multiple instances of parts. The parts to be merged can contain any combination of geometry and orphan mesh nodes and elements; and there are options for merging the geometry, the mesh (orphan and native), or both. In addition, you can cut the geometric portion of a part instance using the geometric portion of one or more part instances to make the cut. Both merge and cut operations create a new part instance and a new part. When you merge or cut part instances, you can choose to suppress or delete the original part instances. The merge and cut operations are described in more detail below.

Merge

You can select multiple part instances and merge them into a single part instance. For example, *Figure 1* shows two part instances that model a 15-pin connector. The two part instances are positioned along a common face and then merged into a single part instance that can be meshed and analyzed.

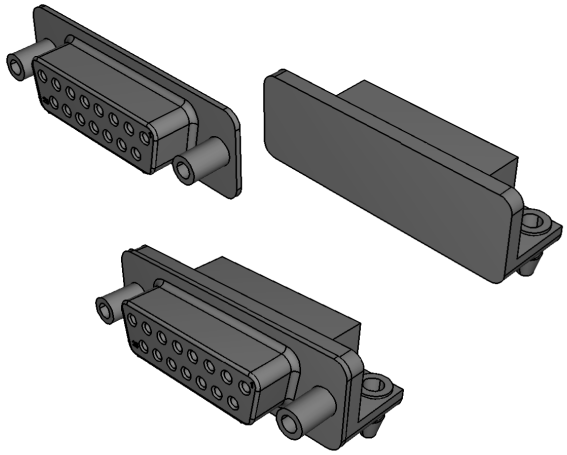


Figure 1: Two part instances merged into a single part instance.

You can merge part instances even if the instances are not touching or overlapping. You can choose whether to remove or retain the intersecting boundaries between the merged part instances, as shown in *Figure 2*. If desired, you can use the **Part Copy** dialog box to create a mirror image of a part about one of the three principal planes. For more information, see *Copying a part*.

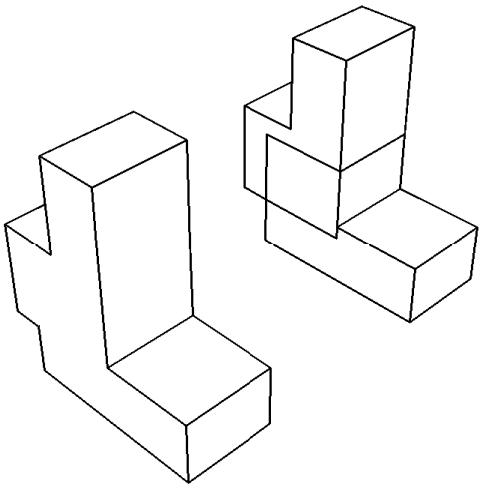


Figure 2: The effect of removing and retaining intersecting boundaries.

If you merge meshes, you can specify the **Node merging tolerance**, which is the maximum distance between nodes that will be merged. Abaqus/CAE creates a compatible mesh by deleting nodes that are closer than the specified distance and replacing them with a single new node. The location of the new node is the average position of the deleted nodes. If the value that you entered for the **Node merging tolerance** is too large, Abaqus/CAE may detect duplicate nodes from the same element. Abaqus/CAE will not merge nodes from the same element. However, the large tolerance can result in a distorted mesh, and Abaqus/CAE asks if you want to continue or end the merging procedure. If no nodes are closer than the specified distance, Abaqus/CAE asks if you want to cancel the procedure or to create a single instance from the selected instances.

When you merge meshed part instances that intersect, you can choose whether to create duplicate elements as well as duplicate nodes. A duplicate element has the same connectivity as another element. By default, Abaqus/CAE deletes duplicate elements, and in most cases you should accept the default behavior. However, you must retain duplicate elements if you want to model a material with a combination of material properties that are not supported by Abaqus, as described in the discussion of stability in *No Compression or No Tension*.

You can choose between the following methods for merging the nodes:

Boundary only

By default, Abaqus/CAE merges the meshed part instances only along their boundaries (defined by free faces for three-dimensional instances and by free edges for two-dimensional instances). Free faces and edges are those faces and edges that belong to only one geometric entity or element. Using this setting, Abaqus/CAE does not check for duplicate nodes in the interior of the parts, which speeds up the merging process. You should retain this default setting if three-dimensional part instances intersect at only a common face or if two-dimensional instances intersect at only a common edge.

All

If the part instances overlap, you may want to merge all the nodes in the selected part instances.

None

Alternatively, you can choose to merge none of the nodes, in which case Abaqus/CAE merges the part instances into a single instance but retains all the original nodes.

In many cases you will be merging part instances that do not intersect but share a common face; for example, the two part instances shown in *Figure 3*.

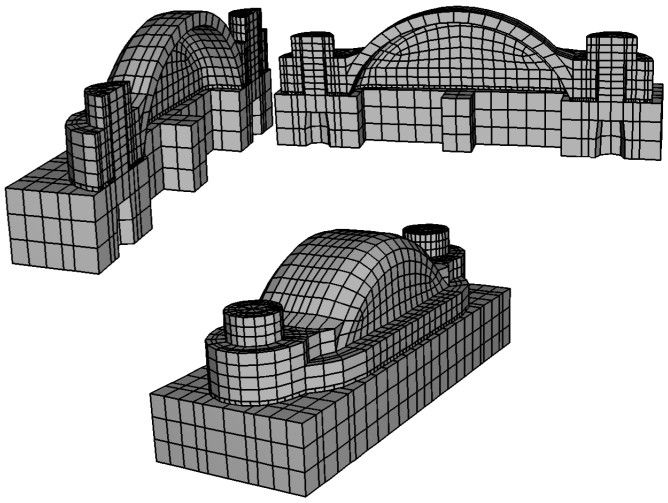


Figure 3: Two meshed part instances merged into a single meshed part instance.

You can also merge selected nodes of a meshed part using the Edit Mesh toolset; for more information, see *Manipulating nodes*.

Although the resulting merged mesh may appear acceptable in the viewport, the mesh may contain small gaps between a node and an element face that are not readily apparent. The mesh may also contain merged faces that have an incompatible mesh pattern. You can use the **Mesh gaps/intersections** tool in the Query toolset to check for small gaps and incompatible faces. For more information, see *Obtaining general information about the model*.

When you merge part instances, any sets or surfaces on the original parts and part instances are mapped to the new part and part instance. If sets or surfaces from different parts have the same name, they are merged into a single set or surface on the merged part and part instance. If you choose to remove intersecting boundaries between the merged part instances, portions of sets or surfaces that lie on those boundary edges and faces are removed from the mapped sets and surfaces.

Section assignments from the original parts are also mapped to the new part. If parts in the original assembly intersect, Abaqus/CAE can map only a single section in the intersecting regions. Similarly, if parts are exactly touching or intersecting and the intersecting boundaries are removed during the merge, Abaqus/CAE maps only a single section to the entire merged part. In these intersecting situations, the section that gets mapped is dependent on a variety of factors and may not match your modeling intent. When merging intersecting regions, you should retain the intersecting boundaries; the boundaries preserve the original section assignments in nonintersecting regions and make it easier to modify mapped section assignments if necessary (for details, see *Managing section assignments*).



Note:

Beam section assignments and rebar reference orientations are not mapped to the merged part. You must recreate them and any associated properties after the merge.

You might want to merge part instances for the following reasons:

- If geometry in separate instances touches or overlaps but you do not merge the instances, Abaqus/CAE creates a separate mesh for each instance and you must apply tie constraints to effectively merge the nodes. In contrast, when you merge part instances, Abaqus/CAE creates a single combined mesh and you do not need to apply computationally expensive tie constraints. In effect, you have created a compatible mesh between

the part instances. If you want to retain the concept of separate part instances, you can create partitions at the common interface of the merged instances.

- Merging part instances allows you to assign material properties to the single part created by the merge operation instead of to each part individually.
- You can apply a display body constraint to a group of merged part instances instead of applying the constraint to each part instance individually.
- When you import a complex assembly, the assembly may appear in Abaqus/CAE as a large number of individual part instances that will be meshed individually. You can merge all the part instances into a single part instance, or you can merge groups of part instances into several separate part instances.

You have the following three options when merging part instances:

Geometry

Merge only the geometry. Any orphan mesh portions of the instances being merged are deleted from the merged part and part instance.

Mesh

Merge all native and orphan mesh components. Any geometry of the instances being merged is deleted from the merged part and part instance. The native mesh portion of the original parts becomes part of the orphan mesh in the new part.

Both

Merge both the geometry and the orphan mesh. Any native meshes are deleted in the process of merging the geometry.

Cut

You can select the geometric portion of a single part instance to be cut, and then you can select the geometry of one or more part instances that are touching or overlapping the part instance to be cut. Abaqus/CAE uses the geometry that will make the cut (the die) to cut away from the geometry of the instance to be cut (the blank). Geometry must touch or overlap to create a cut part and part instance. If the part instance being cut includes orphan mesh elements, they are unaffected by the cut operation.

When you cut a part instance, sets, surfaces, and section assignments from the original part and part instance are mapped to the new part and part instance. Portions of original sets and surfaces that lie within cut portions of the original geometry are removed from the mapped sets and surfaces.

The cut operation is useful if you want to create a mold from a part or vice versa. *Figure 4* shows a bottle and a rectangular blank and how the cut process creates the mold.

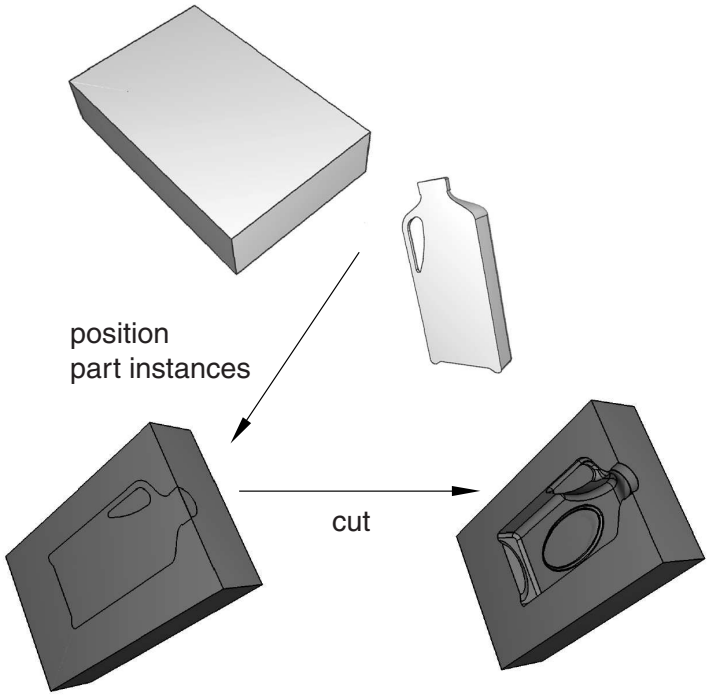


Figure 4: A mold created from a blank and a die using the cut operation.

You cannot make a cut with a shell part instance. Therefore, before the cut operation was performed, the bottle was converted from a shell to a solid part in the Part module. For more information, see *Creating a solid feature from a shell*. In addition, the original part instances (the blank and the die) were suppressed after the cut operation. The cut operation is also useful for modeling a structure and an acoustic medium when you are performing an acoustic or shock analysis.

Note:

You cannot merge or cut part instances that contain virtual topology. When you merge part instances, composite layups and material orientation of the original part are not mapped to the merged part.

For detailed instructions, see *Merging or cutting part instances*.

Merging and cutting independent and dependent part instances

Merging selected part instances results in a new part instance and a new part. If you merge independent part instances, the resulting part instance is also independent. Similarly, if you merge dependent part instances, the resulting part instance is also dependent. Finally, if you merge a combination of independent and dependent part instances, the resulting part instance is dependent.

When you merge the meshes of meshed geometry and/or orphan mesh elements, the resulting part instance is always an orphan mesh part and it is always dependent. When you merge both the meshes and geometry of parts containing geometry and orphan mesh nodes and elements, the resulting part instance is a hybrid containing geometry and orphan nodes and elements, and it is always dependent.

Cutting the geometry of selected part instances also results in a new part instance and a new part. The discussion of merging the geometry of independent and dependent part instances applies to cutting the geometry of independent and dependent part instances; however, orphan mesh elements within a part instance cannot be cut or used to cut the geometry of another instance.

Understanding toolsets in the Assembly module

The Assembly module provides several toolsets that allow you to modify the features that define the assembly. This section describes how these toolsets are used within the Assembly module.

For more detailed information about each toolset, refer to:

- *The Datum toolset*
- *The Feature Manipulation toolset*
- *The Partition toolset*
- *The Query toolset*
- *The Reference Point toolset*
- *The Set and Surface toolsets*

The Display Group toolset is discussed in *Using display groups to display subsets of your model*.

In this section:

- *Using datum geometry in the Assembly module*
- *Manipulating features in the Assembly module*
- *Partitioning the assembly*
- *Querying the assembly*
- *Creating reference points*
- *Using sets and surfaces in the Assembly module*

Using datum geometry in the Assembly module

Within the Assembly module, you use the Datum toolset to provide additional reference geometry (vertices, edges, and surfaces) that is not provided by the assembly. You use the reference geometry to help you define position constraints and to position part or model instances. For example, you can use a datum plane when creating a parallel face or face-to-face constraint if the desired surface does not exist. Similarly, you can use a datum axis when creating a parallel edge or edge-to-edge constraint if the desired edge does not exist. A datum is a parent feature of any constraint in which it was selected. Datums do not modify the geometry of a part or model instance; as a result, you can create datums that refer to both independent and dependent part instances.

Datum geometry that you create in the Part module is transferred along with the rest of the part's geometry when you create a part instance in the Assembly module. In addition, when you translate and rotate a part instance in the Assembly module, a datum created in the Part module is translated and rotated along with the instance. In contrast, a datum created in the Assembly module follows only the reference points that were used to create the datum. As a result if you translate and rotate a part instance, the behavior of the datum may not reflect your design intent. If you know that a datum should be associated with a part, you should create the datum in the Part module.

Figure 1 illustrates a model in which a deformable curved shell will be compressed between two rigid surfaces.

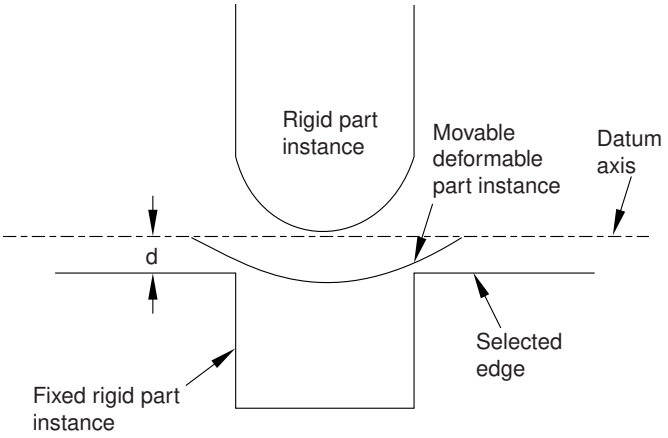


Figure 1: An edge-to-edge constraint applied between a datum axis and a selected edge.

The shell is positioned easily by applying an edge-to-edge position constraint between a selected edge of the lower rigid surface (the fixed part instance) and a datum axis associated with the shell (the movable part instance). The datum axis was created with the deformable part in the Part module and moves along with the movable part instance when the position constraint is applied. In contrast, *Figure 2* illustrates an edge-to-edge position constraint applied between three movable part instances and a fixed datum axis that provides reference geometry. In this example the datum axis was created along the *X*-axis of the assembly and is not associated with any part instance. Applying three edge-to-edge position constraints, one to each of the three part instances shown, would result in alignment of the three instances along the datum axis.

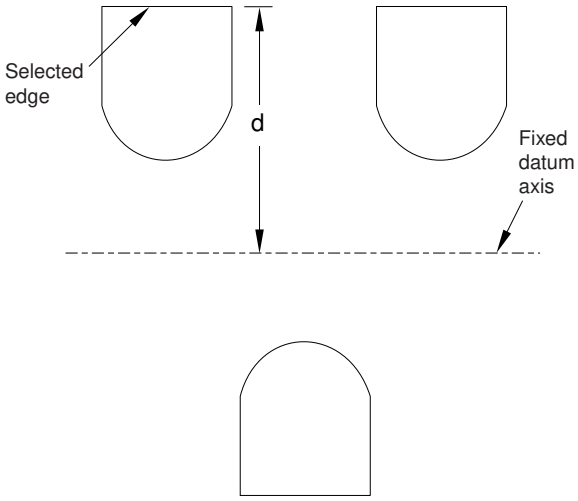


Figure 2: Edge-to-edge constraints applied between multiple parts and a fixed datum axis.

A datum is a feature of the assembly and is regenerated along with the rest of the assembly. You can make datum geometry invisible while still retaining it in the assembly by selecting **View->Assembly Display Options** from the main menu bar. For more information, see *Controlling datum display*.

The triad indicating the origin and the orientation of the global coordinate system is a datum coordinate system created by the Assembly module. You can suppress or delete it, but you cannot modify it.

Additional information

- *Understanding toolsets in the Assembly module*
- *The Datum toolset*

Manipulating features in the Assembly module

Along with datum geometry and partitions, part instances, model instances, and position constraints are considered to be features of the assembly and appear in the list of features in the Model Tree.

Part instances

You can suppress, resume, and delete part instances. You can partition a part instance, but you cannot edit its shape or its features. To modify a part instance, you must edit the original part in the Part module; Abaqus/CAE automatically regenerates instances of a modified part when you return to the Assembly module.

You can make a part instance invisible while still retaining it in the assembly by selecting **View->Assembly Display Options->Instance** from the main menu bar. For more information, see *Controlling instance visibility*. This technique is not the same as suppressing a part instance; a suppressed part instance is removed from the assembly until you resume it. You can also use display groups to make part instances invisible; for more information, see *Using display groups to display subsets of your model*.

You can link part instances, and you can exclude part instances from an analysis; for more information, see *Linking part instances between models* and *Excluding part instances from an analysis*.

Model instances

You can suppress, resume, and delete model instances. To modify a model instance, you must edit the original model's assembly.

You can make a model instance invisible while still retaining it in the assembly by selecting **View->Assembly Display Options->Instance** from the main menu bar. You can also use display groups to make model instances invisible.

Position constraints

You can edit, suppress, resume, and delete position constraints. You can modify the following parameters of a position constraint:

- The direction of the arrow normal to the selected face or along the selected edge of the movable part instance.
- The clearance between the selected face or edge of the movable part instance and the selected face or edge of the fixed part instance. The clearance parameter applies only to face-to-face, edge-to-edge, and contact constraints.

Translations and rotations are not stored as features and cannot be edited, suppressed, resumed, or deleted.

You can use the Feature Manipulation toolset to modify features of the assembly. When you are prompted to select a feature to modify, you can select a visible feature such as a part instance, a datum, or a partition from the viewport. However, to select a position constraint, you must select it from the Model Tree.

The following feature manipulation tools are available from the Feature Manipulation toolset:

Edit

When you edit a feature, Abaqus/CAE displays the **Edit Feature** dialog box and you can modify the feature's parameters or the sketch that defined the feature. You cannot edit part instances; you must return to the Part module to modify the original part.

Regenerate

When you modify features in a complex assembly, it may be convenient to postpone regeneration until you make all your changes, since regeneration can be time consuming. Select **Feature->Regenerate** when you are ready to regenerate the assembly.

Rename

Rename a feature.

Suppress

Suppressing a feature temporarily removes it from the definition of the assembly. A suppressed feature is invisible, cannot be meshed, and is not included in the analysis of the model. Suppressing a parent feature will suppress all of its child features.

Resume

Resuming a feature restores a suppressed feature to the assembly. You can choose to resume all features, the set of features most recently suppressed, or just a selected feature.

Delete

Deleting a feature removes it from the assembly; you cannot restore a deleted feature.

Query

When you query a feature, Abaqus/CAE displays information in the message area and writes the same information to the replay file (*abaqus*`.rpy`) in the form of comments.

Options

The **Feature Options** dialog box allows you to control whether Abaqus/CAE performs self-intersection checks and enables you to prioritize the regeneration of constraint features over other assembly features.

For a more detailed explanation of the Feature Manipulation toolset, see *The Feature Manipulation toolset*.

Additional information

- *Understanding toolsets in the Assembly module*
- *The Feature Manipulation toolset*

Partitioning the assembly

Within the Assembly module, you can use the Partition toolset to partition the assembly into additional regions. You can use vertices, edges, and faces from one part instance to create a partition that divides a second part instance; for example, you might use the **Extend Face** method to partition a cell by extending a face of one part instance into a second part instance. Partitions cannot span part instances.

A partition in the assembly appears in every module that operates on the assembly. Partitions you create in the Part module are transferred along with the rest of the part's geometry when you create a part instance in the Assembly module. Partitions are features of the assembly, and they are regenerated along with the rest of the assembly. You cannot turn off the display of partitions. Partitions modify the geometry of a part instance; as a result, you cannot partition a dependent part instance.

The Partition toolset is not supported and cannot be used with model instances.

Additional information

- *Understanding toolsets in the Assembly module*
- *The Partition toolset*

Querying the assembly

You can use the Query toolset to request either general information or module-specific information. For a discussion of the information displayed by general queries, see *Obtaining general information about the model*.

In addition, you can use the Assembly module-specific queries to determine the following attributes of a part instance or a model instance:

- Name, type, and modeling space
- Origin
- The sum of the translations and rotations applied to the instance

For more information, see *Using the Query toolset to query the assembly*.

Creating reference points

From the main menu bar, select **Tools->Reference Point** to create a reference point on a part instance or a model instance. You can create multiple reference points on the assembly; Abaqus/CAE labels them **RP-1**, **RP-2**, **RP-3**, etc. For more information, see *The Reference Point toolset*.

Abaqus/CAE displays the reference point at the desired location along with its label. You can change the reference point label by clicking mouse button 3 on the feature in the Model Tree and selecting **Rename** from the menu that appears. If desired, you can turn off the display of the reference point symbol and the reference point label; for more information, see *Controlling reference point display*.

Using sets and surfaces in the Assembly module

Sets created by selecting geometry from the assembly are called assembly sets, and you use the Set toolset to create and manage assembly sets. For example, you can select an assembly set to indicate where loads, boundary conditions, and interactions are applied. You can also use assembly sets to define regions of the model from which Abaqus/CAE will generate output during the analysis; for example, selected vertices or faces. Assembly sets can include regions from multiple part instances.

In contrast, part sets are created by selecting geometry from a part in the Part module or the Property module. When you instance a part in the Assembly module, you can refer to any part sets that you previously created; however, Abaqus provides only read-only access to these part sets in assembly-related modules. In addition, you cannot access a part set from the **Set Manager** in assembly-related modules; however, you can select an eligible part set during a procedure by clicking the **Set** button and selecting the set from the **Region Selection** dialog box that appears. For more information, see *Understanding sets and surfaces*.

You create surfaces by selecting faces or edges from the assembly, and you use the Surface toolset to create and manage surfaces. Typically you select a surface when a procedure is expecting a face; for example, when you are applying distributed loads, such as pressure loads, and defining contact interactions. For more information, see *What is a surface?*.

For model instances, any sets or surfaces defined in the original model are brought into the model instances, maintaining the Model Tree hierarchy of features.

Additional information

- *Understanding toolsets in the Assembly module*
- *The Set and Surface toolsets*

Using the Assembly module toolbox

You can access all the Assembly module tools through either the main menu bar or through the Assembly module toolbox. *Figure 1* shows the hidden icons for all the Assembly module tools in the toolbox.

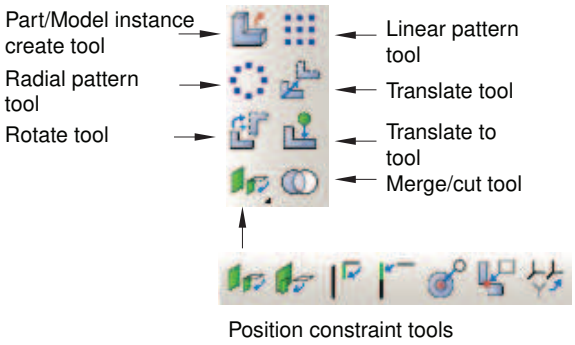


Figure 1: The Assembly module tools.

To see a tooltip containing a brief definition of an Assembly module tool, hold the mouse over the tool for a moment. For information on using toolboxes and selecting hidden icons, see *Using toolboxes and toolbars that contain hidden icons*.

Creating and manipulating part and model instances

This section describes how you use the Assembly module's **Instance** menu to create part and model instances and to position the instances relative to the global coordinate system.
You can also use the **Instance** menu to replace one part instance with another and to convert the constraints applied to a selected part instance to an absolute position. You can access additional functionality using the Model Tree.

In this section:

- *Using the Instance menu*
- *Using the Model Tree to manipulate part instances*
- *Using the Model Tree to switch the context for part or model instances*
- *Creating a part or model instance*
- *Creating a linear pattern of instances*
- *Creating a radial pattern of instances*
- *Translating part or model instances*
- *Translating a part or model instance to another instance*
- *Rotating part or model instances*
- *Replacing an instance*
- *Converting constraints*
- *Merging or cutting part instances*

Using the Instance menu

Use the **Instance** menu to do the following:

- *Create* instances of parts from the current model and add them to the assembly. You can also create instances of other models to add to the current assembly.
- Create a *linear pattern* of part instances.
- Create a *radial pattern* of part instances.
- *Translate* selected part or model instances along a specified vector.
- *Translate* selected part or model instances along a specified vector until they are a specified distance from another instance.
- *Rotate* selected part or model instances through a specified angle about a specified axis.
- *Replace* a part instance with a second part instance.
- *Convert* any position constraints to an absolute position.
- *Merge or cut* part instances.

You may find it more convenient to access the instance tools using the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

Additional information

- *Using the Assembly module toolbox*
- *The Assembly module*

Using the Model Tree to manipulate part instances

You can access additional functionality for manipulating part instances using the Model Tree. After you select a part instance or multiple instances in the Model Tree, you can click mouse button 3 and select from the following options:

- To change a part instance from dependent to independent, select **Make Independent** from the menu that appears. Similarly, you can select **Make Dependent** from the menu to change a part instance from independent to dependent.
- To link part instances, select **Link Instances** from the menu that appears to display the **Link Instances** dialog box and do the following:

  1. For each child part instance, specify the model and part instance to which you want to link. By default, the dialog box displays all models in the model database other than the current model that have part instances with the same names as the selected instances. You can specify other values by clicking in the **Model** or **Instance** column of the table and selecting from the list of names displayed.
  2. If you want the child part instances excluded from the input file that Abaqus/CAE generates when you submit the job for analysis, toggle on **Also exclude child instances from simulation**.
  3. Click **Link**.

     Icons appear in the Model Tree to indicate the linked status of the part instances. By default, linked part instances are colored gray in the viewport.

  Similarly, you can select **Unlink Instances** from the menu to unlink part instances that were previously linked. For more information, see *Linking part instances between models*.
- To exclude part instances from the analysis, select **Exclude from Simulation** from the menu that appears. Icons appear in the Model Tree to indicate the excluded status of the part instance. By default, part instances that are excluded from the analysis are colored dark gray in the viewport. Similarly, you can select **Include in Simulation** from the menu to include a part instance in an analysis for which it was previously excluded. For more information, see *Excluding part instances from an analysis*.
- To ignore the invalid status of a part instance, select **Ignore Invalidity**. For more information, see *Working with invalid parts*.

Additional information

- *What is the difference between a dependent and an independent part instance?*
- *How do I decide whether to create a dependent or an independent part instance?*
- *Changing from a dependent to an independent part instance or vice versa*
- *Linking part instances between models*
- *Excluding part instances from an analysis*

Using the Model Tree to switch the context for part or model instances

You can use the Model Tree in the Assembly module or Mesh module to switch the context for part instances or for model instances and their child part instances.

In the Model Tree, select the instance for which you want to switch the context. If you select multiple instances, the context switches to the part or model for the first instance that you select. Click mouse button 3, and select **Switch to part/model context**. Abaqus/CAE switches the context as follows (depending on your selection):

- In the Assembly module, if you selected a part instance or a child part instance, the context switches to the Part module with the original part from which the instance was created displayed in the viewport.
- In the Mesh module, if you selected a part instance or a child part instance, the context switches to display the original part from which the instance was created in the viewport.
- If you selected a model instance, the context switches to display the original model from which the model instance was created in the viewport.

Additional information

- *Working with part instances*
- *Working with model instances*

Creating a part or model instance

To create a part or model instance, select **Instance->Create** from the main menu bar and select the desired part or model from the **Create Instance** dialog box that appears.
You can select from any of the existing parts in the current model or any of the existing models in the current model database. You can create multiple instances of the same part or model, but you cannot assemble instances of parts or models that were created in different modeling spaces (three-dimensional, two-dimensional, or axisymmetric).

When you create the first instance,Abaqus/CAE displays a graphic symbol indicating the origin and orientation of the assembly's global coordinate system. This symbol is a datum coordinate system. If desired, you can hide it using the assembly display options; for more information, see *Controlling datum display*.

By default, Abaqus/CAE creates dependent part instances. A dependent instance is only a pointer to the geometry of the original part. As a result, many operations are not allowed on a dependent part instance; for example, you cannot add partitions, create virtual topology, or mesh the instance. In contrast, an independent part instance is a copy of the geometry of the original part. You can perform most operations on an independent instance; for example, you can add partitions, create virtual topology, and mesh the instance. You cannot create both an independent and a dependent instance of the same part. You can select a part instance from the Model Tree and change it from independent to dependent or vice versa. For more information, see *What is the difference between a dependent and an independent part instance?*.

When you create an instance, by default Abaqus/CAE positions the instance so that the origin of the original geometry aligns with the origin of the assembly coordinate system. When you create multiple instances, a new instance can be positioned over an existing instance. However, if you toggle on **Auto-offset from other instances** in the **Create Instance** dialog box, Abaqus/CAE translates each new instance along the *X*–axis until it does not overlap any existing instances. If the assembly is axisymmetric, Abaqus/CAE translates the new instance along the axis of revolution instead of along the *X*–axis.

1. From the main menu bar, select **Instance->Create**.

   Abaqus/CAE displays the **Create Instance** dialog box.

   Tip: You can also create an instance using the  tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

2. Choose **Parts** or **Models**.

   Depending on your choice, the **Create Instance** dialog box shows you a list of all the existing parts in the current model or all the other models in the model database.

3. Choose the desired parts or models from the list. You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple items.

   A temporary image of the selected instances appears in the current viewport. Abaqus/CAE positions the temporary images so that their origins coincide with the origin of the global coordinate system.

4. By default, Abaqus/CAE creates **Dependent** part instances. If desired, toggle on **Independent** to create an independent part instance.
5. If desired, toggle on **Auto-offset from other instances** to offset the new instances.
6. **Optional:** Use the **Name** text field at the top of the dialog box to name the instance you are creating.

   - If you do not specify a name, a name is generated automatically.
   - If multiple parts are selected, the name you specify is applied to the first selection. The other instance names are appended with the suffix “-1,” “-2,” “-3,” etc.
   - If you enter an existing name, Abaqus/CAE issues a warning in the CLI and generates a name.

7. If you are satisfied that you have selected the correct instances, click **Apply**.

Abaqus/CAE creates the instances and applies an auto-offset if selected.

8. To create additional instances, repeat this procedure from Step 2. When you have finished creating instances, click **OK** to close the **Create Instance** dialog box.

Additional information

- *Using the Instance menu*
- *Working with part instances*
- *Working with model instances*

Creating a linear pattern of instances

To create multiple copies of a selected instance in a linear pattern, select **Instance->Linear Pattern** from the main menu bar. You can create a pattern that extends in one direction (for example, horizontally or vertically), or you can create a pattern that extends in two directions (for example, both horizontally and vertically). You cannot edit a pattern after you create it. You can specify the following:

- The number of instances to create in each direction, including the selected instance. You can create any number of instances.
- The spacing between each instance along the specified direction.
- A line that defines the direction along which Abaqus/CAE generates the instances.

For more information, see *Creating patterns of instances*.

1. From the main menu bar, select **Instance->Linear Pattern**.

   Tip: You can also create a linear pattern of instances using the tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

2. Select the instances that you want to copy.

   Tip: To select more than one instance, hold down the **[Shift]** key as you click each instance or drag a rectangle around the instances. To unselect an instance, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*.

3. Click mouse button 2 to indicate that you have finished selecting instances.

   Abaqus/CAE displays the **Linear Pattern** dialog box.

4. From the **Linear Pattern** dialog box, configure the pattern in **Direction-1** (by default, **Direction-1** is the *X*-direction):

   a. Click the arrows to the right of **Number** to increase or decrease the number of copies to create, including the selected instances. The number of copies in the assembly updates when you click the arrows and provides a preview of the setting.

      Alternatively, you can type in a number and press **[Enter]** to preview the setting. You can enter any number greater than or equal to 1. If you enter a value of 1, Abaqus/CAE displays only the selected instances and does not create any copies of the selected instances; in effect, you are disabling copies in **Direction-1**.

   b. Enter the **Spacing** between each copy along the specified direction.

   c. By default, Abaqus/CAE creates the copies along the *X*-direction. If you want to change the direction in which Abaqus/CAE creates the copies, click and select a line from the assembly to define the new direction. You must pick a straight edge or a datum axis.

   d. By default, Abaqus/CAE creates the copies in the positive direction. Click to reverse the direction in which Abaqus/CAE creates the copies.

5. To create copies in a second direction, enter a **Number** greater than 1 and specify the **Spacing**, the **Direction**, and the **Flip** direction for **Direction-2** (by default, **Direction-2** is the *Y*-direction). You must enter a **Number** greater than 1 for at least one direction.

6. In most cases you will want to preview the linear pattern that Abaqus/CAE will create as you enter values in the **Linear Pattern** dialog box. However, if you choose to create a large number of copies, the preview capability may impact the performance of Abaqus/CAE. In this case you should toggle off the **Preview** button.
7. To copy more instances, repeat the above steps beginning with Step 1.

Additional information

- *Creating patterns of instances*
- *Creating a radial pattern of instances*

Creating a radial pattern of instances

To create multiple copies of a selected instance in a radial pattern, select **Instance->Radial Pattern** from the main menu bar. You cannot edit a pattern after you create it. You can specify the following:

- The number of copies to create in the radial pattern, including the selected instances. You can create any number of instances.
- The total angle between the original instance and the last copy in the pattern.
- The position of the axis of the circular pattern.

For more information, see *Creating patterns of instances*.

1. From the main menu bar, select **Instance->Radial Pattern**.

   Tip: You can also create a radial pattern of instances using the tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

2. Select the instances that you want to copy.

   Tip: To select more than one instance, hold down the **[Shift]** key as you click each instance or drag a rectangle around the instances. To unselect an instance, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*.

3. Click mouse button 2 to indicate that you have finished selecting instances.

   Abaqus/CAE displays the **Radial Pattern** dialog box.

4. From the **Radial Pattern** dialog box, configure the radial pattern:
   a. Click the arrows to the right of **Number** to increase or decrease the number of copies to create, including the selected instances. The number of copies in the assembly updates when you click the arrows and provides a preview of the setting.

      Alternatively, you can type in a number and press **[Enter]** to preview the setting. You can enter any number greater than or equal to 2.

   b. Enter the **Total angle** between the original instances that you selected and the final copy. The angle must be between –360° and +360°. A positive angle corresponds to a counterclockwise direction.
   c. By default, Abaqus/CAE rotates the selected instance about the *Z*-axis to create the pattern. To define a new axis of rotation, click and select a line from the assembly that represents the new axis of rotation. You can also select an axis from the global coordinate system triad.

5. In most cases you will want to preview the radial pattern that Abaqus/CAE will create as you enter values in the **Radial Pattern** dialog box. However, if you choose to create a large number of copies, the preview capability may impact the performance of Abaqus/CAE. In this case you should toggle off the **Preview** button.
6. To copy more instances, repeat the above steps beginning with Step 1.

Additional information

- *Creating patterns of instances*
- *Creating a linear pattern of instances*

Translating part or model instances

Select **Instance->Translate** from the main menu bar to move selected part or model instances along a selected vector.

The direction and magnitude of the vector are arbitrary except that you can translate axisymmetric part instances only along the axis of rotation. If the translation conflicts with a previous position constraint; for example, a constraint that aligns two faces, Abaqus/CAE applies the components of translation only along the unconstrained degrees of freedom. If all of the degrees of freedom are constrained, Abaqus/CAE displays an error message and the translation fails.

When you create the first instance in your assembly, Abaqus/CAE displays a graphic indicating the origin and orientation of the assembly's default coordinate system. You can use this graphic to help you decide how to translate your part instances. In addition, you can use the Query toolset to review the sum of the translations and rotations previously applied to an instance and the distance between selected vertices. Translations and rotations are not considered features of the assembly and cannot be edited or deleted.

1. From the main menu bar, select **Instance->Translate**.

   Tip: You can also translate instances using the  tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the part or model instances to translate. You can also click **Instances** on the right of the prompt area and select the instances to translate from the dialog box that appears.

   Tip: If you are unable to select the desired instance, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple instances.

   Abaqus/CAE highlights the selected instances.

3. Specify the translation vector using one of the following methods:

   - Select an axis or linear edge from the viewport, and enter a distance for the translation. Abaqus/CAE displays an arrow along the selected axis/edge, and, if needed, you can flip the direction to define the translation vector.
   - Select the start and end points of the translation vector. You can select any existing vertices or datum points, or you can enter the coordinates in the text box in the prompt area. If you want to enter coordinates for a coordinate system other than the global coordinate system, click **Select** on the right side of the prompt area and select a local coordinate system. By default, the coordinate system applied to the start point is also applied to the end point.

   Abaqus/CAE displays a temporary image indicating the translation that will be applied to the selected instance. You cannot edit or delete a translation after it is applied. Attempting to translate an instance may result in a conflict with existing position constraints. Abaqus/CAE applies the components of translation only along the unconstrained degrees of freedom. If all of the degrees of freedom are constrained, Abaqus/CAE displays an error message and the translation fails.

4. Do one of the following:

   - If you are satisfied the translation is correct, click **OK** in the prompt area.

     Abaqus/CAE translates the instance and positions it at the same location as the temporary image of the instance.

- If you are not satisfied with the translation, click the **Previous** button () and specify a new translation vector.
- Click  to cancel the procedure.

Additional information

• *Using the Instance menu*
• *Creating the assembly*

Translating a part or model instance to another instance

Select **Instance->Translate To** from the main menu bar to position two instances by translating one instance along a vector defining the direction of motion until selected faces or edges of the movable instance are separated by a specified distance from selected faces or edges of the fixed instance.

Abaqus/CAE calculates the contact only within a tolerance based on the size of your model. As a result, contact may not be precise unless it is applied between two planar surfaces. If the selected faces or edges never contact when Abaqus/CAE translates the movable instance, the translation is not applied.

1. From the main menu, select **Instance->Translate To**.

   Tip: You can also define a **Translate To** constraint using the  tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the faces (for three-dimensional part instances) or edges (for two-dimensional part instances) from the part or model instance that will move and the instance that will remain fixed. You can select more than one face or edge from both the fixed and the movable instances. Selecting multiple faces or edges is useful if you are not sure what part of the model will come in contact when the movable instance moves along the selected vector. However, for faster processing you should select as few faces or edges as possible. You cannot select a datum plane.
3. Specify the vector that defines the direction of motion using one of the following methods. If the instances are axisymmetric, the translation vector must be parallel to the axis of revolution.
   - Select an axis or linear edge from the viewport. Abaqus/CAE displays an arrow along the selected axis/edge, and, if needed, you can flip the direction to define the translation vector.
   - Select the start and end points of the vector. You can select any existing vertices or datum points, or you can enter the coordinates in the text box in the prompt area. If you want to enter coordinates for a coordinate system other than the global coordinate system, click **Select** on the right side of the prompt area and select a local coordinate system. By default, the coordinate system applied to the start point is also applied to the end point.
4. In the text box that appears in the prompt area, enter a value for the clearance between the two selected faces; a negative value indicates overclosure. A translation is not saved as a feature, and you cannot change the clearance after you have completed the procedure.
5. From the prompt area, click **Preview**.

   Abaqus/CAE moves the movable part or model instance along the translation vector until the selected faces of the movable instance are separated by the specified clearance from the selected faces of the fixed instance.
6. If the new position of the instance is correct, click **Done** in the prompt area.

   Attempting to translate an instance may result in a conflict with existing position constraints. Abaqus/CAE applies the components of translation only along the unconstrained degrees of freedom. If all of the degrees of freedom are constrained, Abaqus/CAE displays an error message and the translation fails. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the translation.

Additional information

- *Using the Constraint menu*

- *Creating the assembly*
- *Converting constraints*

Rotating part or model instances

Select **Instance->Rotate** from the main menu bar to rotate selected part or model instances about a selected axis.

To rotate a three-dimensional instance, you must select two points that define the axis about which the instance will rotate. To rotate a two-dimensional instance, you must select a single point about which the instance will rotate. You cannot rotate axisymmetric instances. If the rotation conflicts with a previous position constraint (for example, a constraint that aligns two faces), Abaqus/CAE displays an error message and the rotation fails.

When you create the first instance in the assembly, Abaqus/CAE displays a graphic indicating the origin and orientation of the assembly's global coordinate system. You can use this graphic to help you decide how to rotate your part and model instances. In addition, you can use the Query toolset to review the sum of the translations and rotations previously applied to an instance and the distance between selected vertices. Rotations and translations are not considered features of the assembly and cannot be edited or deleted.

1. From the main menu bar, select **Instance->Rotate**.

   Tip: You can also rotate an instance using the tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. From the assembly, select the part or model instances to rotate. You can also click **Instances** on the right of the prompt area and select the instances to rotate from the dialog box that appears.

   Tip: If you are unable to select the desired instance, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple instances.

   Abaqus/CAE highlights the selected instances.

3. Specify the axis of rotation using one of the following methods:

   - Select an axis or linear edge from the viewport.
   - Select the start and end points of the rotation vector. You can select any existing vertices or datum points, or you can enter the coordinates in the text box in the prompt area. If you want to enter coordinates for a coordinate system other than the global coordinate system, click **Select** on the right side of the prompt area and select a local coordinate system. By default, the coordinate system applied to the start point is also applied to the end point.

4. In the text box that appears in the prompt area, enter the angle of rotation. A positive angle indicates a counterclockwise rotation; a negative angle indicates a clockwise rotation.

   Abaqus/CAE displays a temporary image indicating the rotation that will be applied to the selected instances. You cannot edit or delete a rotation after it is applied. Attempting to rotate an instance may result in a conflict with existing position constraints. If a conflict occurs, Abaqus/CAE displays an error message and the rotation fails.

5. Do one of the following:

   a. If you are satisfied that the rotation is correct, click **OK** in the prompt area.

      Abaqus/CAE rotates the instance and positions it at the same location as the temporary image of the instance.

   b. If you are not satisfied with the rotation, click the **Previous** button ( ) and specify a new rotation.

c. Click () to cancel the procedure.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*

Replacing an instance

Select **Instance->Replace** from the main menu bar to replace a selected instance with an instance of another part/model. Abaqus/CAE positions the new instance so that its origin is located at the origin of the original instance and their axes align. In addition, you can choose whether the new instance inherits all the constraints from the instance it replaced.

The replace operation does not change the attributes of the instance. For example, if the original instance is dependent, the instance that replaces it will also be dependent. As a result, if an independent instance of a part exists, you cannot use the replace procedure to create a dependent instance of the same part.

Replacing an instance is most useful when you are replacing an instance with one that has similar geometry. For example, the new instance might have additional detail that was not present in the original instance.

You can only replace a part instance with a part instance and a model instance with a model instance.

1. From the main menu bar, select **Instance->Replace** to replace a selected instance.
2. From the assembly, select the instance to replace. You can also click the **Instance List** button on the right of the prompt area and select the instance from the **Instance List** dialog box that appears.

   Tip: If you are unable to select the desired instance, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE displays the **Replace Instance** dialog box with a list of all the parts/models in the model.
3. From the **Replace Instance** dialog box, select the part/model that will replace the selected part/model instance in the assembly.

   Abaqus/CAE displays a temporary image of the new part instance in the assembly and positions it so that its origin is located at the origin of the original part instance and their axes align.
4. If the correct instance is selected, click **OK** in the **Replace Instance** dialog box.

   If you have not applied any position constraints to the original instance, Abaqus/CAE replaces it with the new instance.
5. If you have applied position constraints to the original instance, you must choose one of the following buttons in the prompt area:

   - Click **OK** to position the new instance in the same location as the instance it is replacing. Abaqus/CAE removes any constraints that were applied to the original instance, while maintaining its position.
   - Click **Apply previous constraints** if you want the new instance to inherit position constraints from the instance being replaced. Abaqus/CAE applies all the previous constraints that can be satisfied by the new instance; any constraints that cannot be satisfied are ignored.

   Abaqus/CAE replaces the original part instance with the new part instance and the original model instance with the new model instance.

Additional information

- *Replacing an instance*
- *Creating a part or model instance*
- *Applying constraints to part and model instances*
- *Working with part instances*

Converting constraints

To remove all the face, edge, coaxial, and contact constraints applied to a selected part or model instance while leaving the instance in its current position, select **Instance->Convert Constraints** from the main menu bar. The conversion is equivalent to applying a single translation and rotation to the instance that moves it from its original position to the current position. Any previous constraints no longer appear in the list of features and cannot be restored.

1. From the main menu bar, select **Instance->Convert Constraints** to convert any existing constraints to the current position.
2. From the assembly, select the part or model instance whose constraints you want to convert. You can also click the **Instance List** button on the right of the prompt area and select the instance from the **Instance List** dialog box that appears.

   Tip: If you are unable to select the desired instance, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   The instance does not move, but Abaqus/CAE converts any existing constraints to the current position. You cannot restore the original face, edge, coaxial, and contact constraints.

Additional information

- *How the position constraint methods differ*
- *How conflicts can arise between position constraints, translations, and rotations*

Merging or cutting part instances

To merge a selected group of part instances, select **Instance->Merge/Cut** from the main menu bar. You can merge the geometry of parts that you created in the Part module, you can merge the meshes of part instances (any native mesh becomes an orphan mesh), or you can merge both geometry and orphan mesh features. You can also merge selected nodes using the Edit Mesh toolset in the Mesh module; for more information, see *Manipulating nodes*. You can cut the geometry of a selected part instance using the geometry of one or more other part instances to make the cut; any orphan mesh nodes or elements within the instance being cut are retained in the new cut part instance.

The merge or cut operation creates both a new part instance in the assembly and a new part. You can choose to suppress the original part instances that you selected, or you can delete them from the assembly. For more information, see *Performing Boolean operations on part instances*.

You cannot use the **Merge/Cut** command on model instances.

1. From the main menu bar, select **Instance->Merge/Cut**.

   Tip: You can also merge or cut part instances using the  tool from the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays the **Merge/Cut Instances** dialog box.
2. Enter the name of the part that will be created by the operation.
3. Select the type of operation:
   - To merge the geometry of part instances, choose **Merge** and **Geometry**. Any native or orphan mesh nodes and elements are not included in the new part.
   - To merge the meshes of part instances, choose **Merge** and **Mesh**. Any geometry is not included in the new part, and any native mesh becomes an orphan mesh.
   - To merge both the geometry and mesh features of part instances, choose **Merge** and **Both**. Geometry is included in the new part, and any native mesh is deleted.
   - To cut part instances, choose **Cut geometry**.
4. Choose how you would like Abaqus/CAE to handle the original instances that are being merged or cut:
   - Choose **Suppress** to suppress the original part instances but retain them in the model database. After you complete the **Merge/Cut** operation, you can resume the original part instances if necessary (see *Suppressing and resuming objects*).
   - Choose **Delete** to delete the original part instances from the model database. You cannot recover deleted part instances.
5. If you chose a **Merge** operation, do the following:
   a. Choose the desired options:

Geometry

By default, Abaqus/CAE removes the boundaries between intersecting part instances. If you want to retain the boundaries between intersecting part instances, choose **Retain** from the bottom of the **Merge/Cut Instances** dialog box. The effect of removing and retaining the boundaries is shown in *Figure 1*.

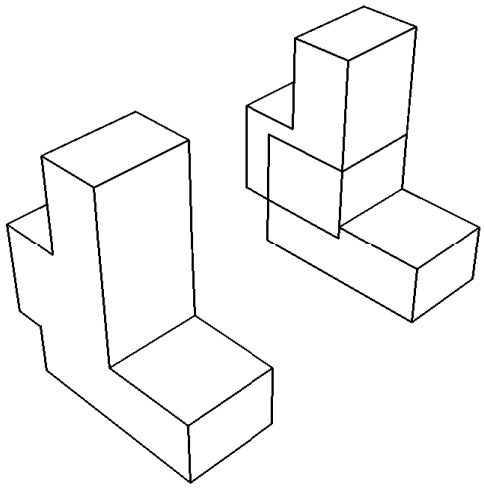


Figure 1: The effect of removing and retaining intersecting boundaries.

Mesh

Choose the method that Abaqus/CAE will use to merge nodes:

- **Boundary only**. By default, Abaqus/CAE merges the meshes only along their boundaries. Therefore, Abaqus/CAE does not check for duplicate nodes in the interior of the parts, which speeds up the merging process. You should retain this default setting if the part instances intersect at only a common face.
- **All**. Merge all the nodes in the selected part instances. By default, Abaqus/CAE removes elements that have the same connectivity as an existing element. Toggle off **Remove duplicate elements** to retain duplicate elements.
- **None**. Merge the part instances into a single part instance but retain the original nodes.

If applicable, enter the **Node merging tolerance**. Abaqus/CAE deletes nodes that are closer than the specified tolerance and replaces them with a single new node. The location of the new node is the average position of the group of nodes that were merged into the new node.

b. Click **Continue**.

c. Select the part instances to merge. You can use a combination of **[Ctrl] + Click** and **[Shift] + Click** to select multiple part instances. You can also click the **Instance List** button on the right of the prompt area and select the instances from the **Instance List** dialog box that appears.



Tip: If you are unable to select the desired part instances, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

The part instances do not have to be touching or overlapping.

d. Click mouse button 2 to indicate that you have finished selecting part instances.

e. If you are merging meshes and the value that you entered for the **Node merging tolerance** is too large, Abaqus/CAE may detect duplicate nodes from the same element. Abaqus/CAE will not merge nodes from the same element, but the large tolerance can result in a distorted mesh. If the **Node merging tolerance** is too large, Abaqus/CAE asks if you want to continue merging the part instances.

- Click **Yes** to continue.
- Click **No** to cancel the merging procedure.

f. Abaqus/CAE highlights in magenta any nodes that will be merged and asks if you wish to proceed.

- Click **Yes** to continue.
- Click **No** to cancel the merging procedure.

Abaqus/CAE merges the nodes that are closer than the specified tolerance and replaces them with a single new node. The location of the new node is the average position of the group of nodes that were merged into the new node. If no nodes are closer than the specified tolerance, Abaqus/CAE asks if you want to cancel the procedure or merge the selected instances into a single part instance.

Abaqus/CAE merges the selected instances, creates a new part instance and a new part, and modifies sets and surfaces to include the new part instance.

6. If you chose a **Cut** operation, do the following:

a. Click **Continue**.

b. Select the geometry of the part instance to be cut. You can select only one part instance, and only the geometry is selected, even if the part includes orphan mesh nodes and elements.

c. Select the part instances that will make the cut. The cutting geometry must touch or overlap the geometry of the part instance to be cut.

d. Click mouse button 2 to indicate that you have finished selecting part instances.

Abaqus/CAE cuts the selected instance and creates a new part instance and a new part. Any orphan mesh on the part being cut is copied to the new instance and part.

Additional information

• *How the position constraint methods differ*

• *How conflicts can arise between position constraints, translations, and rotations*

Applying constraints to part and model instances

This section describes how you use the Assembly module's **Constraint** menu to apply positioning constraints to part and model instances in the assembly.

In this section:

- *Using the Constraint menu*
- *Constraining two instances with parallel planar faces*
- *Constraining two instances with parallel planar faces separated by a specified distance*
- *Constraining two instances with parallel edges*
- *Constraining two instances with parallel edges separated by a specified distance*
- *Constraining two instances with coaxial faces*
- *Constraining two instances with coincident points*
- *Constraining two instances with parallel coordinate systems*

Using the Constraint menu

Use the **Constraint** menu to apply a constraint that does the following:

- ***Parallel Face***. Positions a movable part or model instance so that a selected face is parallel to a selected face of a fixed instance.
- ***Face to Face***. Positions a movable part or model instance so that a selected face is parallel to and a specified distance away from a selected face of a fixed instance.
- ***Parallel Edge***. Positions a movable part or model instance so that a selected edge is parallel to a selected edge of a fixed instance.
- ***Edge to Edge***. Positions a movable part or model instance so that a selected edge is parallel to and a specified distance away from a selected edge of a fixed instance.
- ***Coaxial***. Positions a movable part or model instance so that the axis of revolution of a selected face is coincident with the axis of revolution of a selected face of a fixed instance.
- ***Coincident Point***. Positions a movable part or model instance so that a selected point is coincident with a selected point of a fixed instance.
- ***Parallel CSYS***. Positions a movable part or model instance so that a selected datum coordinate system associated with the instance is parallel to a selected datum system of a fixed instance.

Constraints position one instance relative to another; as a result, constraints cannot be applied until your assembly contains two or more part or model instances.

You may find it more convenient to access the constraint tools using the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

Additional information

- *Creating the assembly*
- *The Assembly module*

Constraining two instances with parallel planar faces

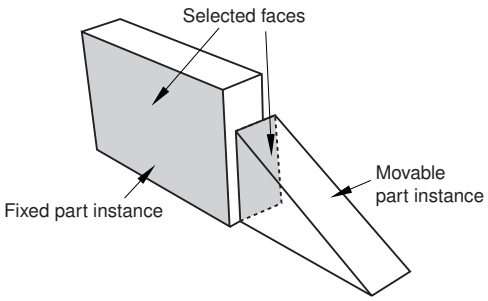
Select **Constraint->Parallel Face** from the main menu bar to apply a constraint that positions a movable instance so that a selected face is parallel to a selected face of a fixed instance. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset.

1. From the main menu, select **Constraint->Parallel Face**.

   Tip: You can also apply the parallel face constraint using the tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a planar face from the part or model instance that will move and a planar face from the instance that will remain fixed, as shown in the following figure:



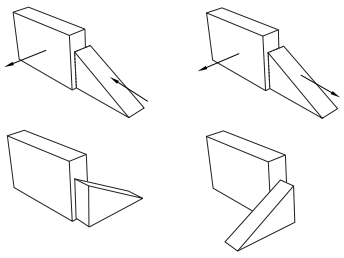
   Abaqus/CAE displays arrows normal to the selected faces.

   When Abaqus/CAE prompts you to select the face from the fixed instance, you can select a datum plane that was created in either the Part or Assembly module. In contrast, when you select the face from a movable part instance, you can select a datum plane that was created only in the Part module.

3. From the buttons in the prompt area, do one of the following:

   - Click **OK** to accept the direction of the arrow on the face of the movable instance.
   - Click **Flip** to reverse the direction of the arrow on the face of the movable instance, and click **OK**.

   Abaqus/CAE positions the movable part or model instance so that the two faces are parallel and the arrows point in the same direction. The orientation of the fixed instance remains unchanged. The effect of changing the direction of the arrow is illustrated in the following figure:



   If the parallel face constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that

will move and the instance that will remain fixed. Alternatively, you can delete the existing relative position constraints, apply absolute position constraints, and reapply the parallel face constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Constraining two instances with parallel planar faces separated by a specified distance

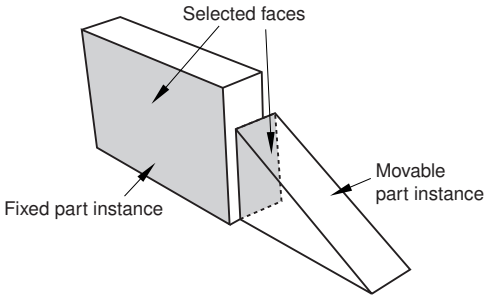
Select **Constraint->Face to Face** from the main menu bar to apply a constraint that positions a movable instance so that a selected face is parallel to and a specified distance away from a selected face of a fixed instance. The face-to-face constraint is a feature of the assembly and can be suppressed or deleted using the Feature Manipulation toolset. In addition, you can edit the clearance between the two selected faces.

1. From the main menu, select **Constraint->Face to Face**.

   Tip: You can also apply the face-to-face constraint using the tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a planar face from the part or model instance that will move and a planar face from the instance that will remain fixed, as shown in the following figure:

   Abaqus/CAE displays arrows normal to the selected faces.

   When Abaqus/CAE prompts you to select the face from the fixed instance, you can select a datum plane that was created in either the Part or Assembly module. In contrast, when you select a face from the movable part instance, you can select a datum plane that was created only in the Part module.

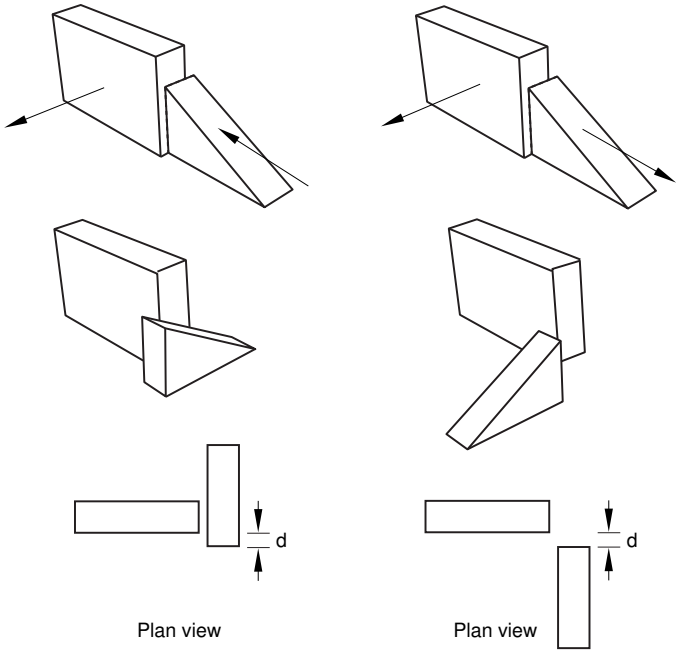
3. From the buttons in the prompt area, do one of the following:

   - Click **OK** to accept the direction of the arrow on the face of the movable instance.
   - Click **Flip** to reverse the direction of the arrow on the face of the movable instance, and click **OK**.

   The effect of changing the direction of the arrow is illustrated in the next step.

4. In the text field that appears in the prompt area, enter the distance between the selected faces, positive along the normal to the face of the fixed instance.

   Abaqus/CAE positions the movable instance so that the two faces are parallel and the arrows point in the same direction. In addition, the movable instance is translated to satisfy the clearance specified. The orientation of the fixed instance remains unchanged. The effect of specifying the distance is illustrated in the following figure:



If the face-to-face constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the face-to-face constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Constraining two instances with parallel edges

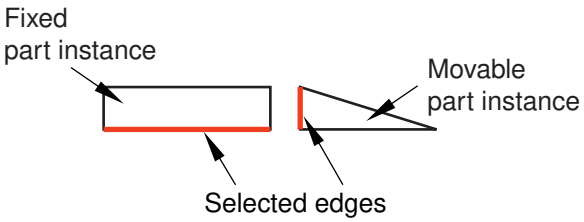
Select **Constraint->Parallel Edge** from the main menu bar to apply a constraint that positions a movable instance so that a selected edge is parallel to a selected edge of a fixed instance. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset.

1. From the main menu, select **Constraint->Parallel Edge**.

   Tip: You can also apply the parallel edge constraint using the tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a straight edge from the instance that will move and a straight edge from the instance that will remain fixed, as shown in the following figure:

   In addition to selecting an edge or a datum axis, you can also select one of the axes of a datum coordinate system.

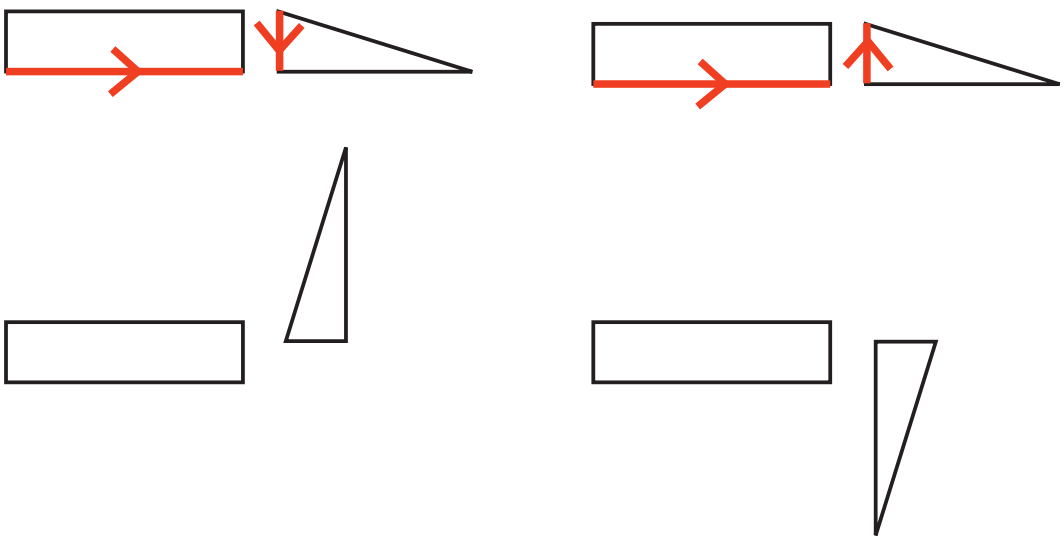
   Abaqus/CAE displays arrows along the selected edges.

   When Abaqus/CAE prompts you to select the edge from the fixed instance, you can select a datum axis that was created in either the Part or Assembly module. In contrast, when you select the edge from a movable part instance, you can select a datum axis that was created only in the Part module.

3. From the buttons in the prompt area, do one of the following:

   - Click **OK** to accept the direction of the arrow along the edge of the movable instance.
   - Click **Flip** to reverse the direction of the arrow along the edge of the movable instance, and click **OK**.

   Abaqus/CAE positions the movable instance so that the two edges are parallel and the arrows point in the same direction. The orientation of the fixed instance remains unchanged. The effect of changing the direction of the arrow is illustrated in the following figure:



If the parallel edge constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the parallel edge constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Constraining two instances with parallel edges separated by a specified distance

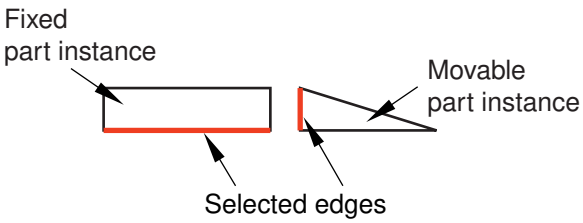
Select **Constraint->Edge to Edge** from the main menu bar to apply a constraint that positions a movable instance so that a selected edge is parallel to a selected edge of a fixed instance. In addition, if the instances are two-dimensional, you must specify the distance between the selected edges; otherwise, Abaqus/CAE makes them coincident. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset. In addition, you can edit the clearance between the two selected edges, where applicable. For more information, see *How the position constraint methods differ*.

1. From the main menu, select **Constraint->Edge to Edge**.

   Tip: You can also apply the edge-to-edge constraint using the tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a straight edge from the instance that will move and a straight edge from the instance that will remain fixed, as shown in the following figure:

   In addition to selecting an edge or a datum axis, you can also select one of the axes of a datum coordinate system.

   Abaqus/CAE displays arrows along the selected edges.

   When Abaqus/CAE prompts you to select the edge from the fixed instance, you can select a datum axis that was created in either the Part or Assembly module. In contrast, when you select the edge from a movable part instance, you can select a datum axis that was created only in the Part module.

3. From the buttons in the prompt area, do one of the following:

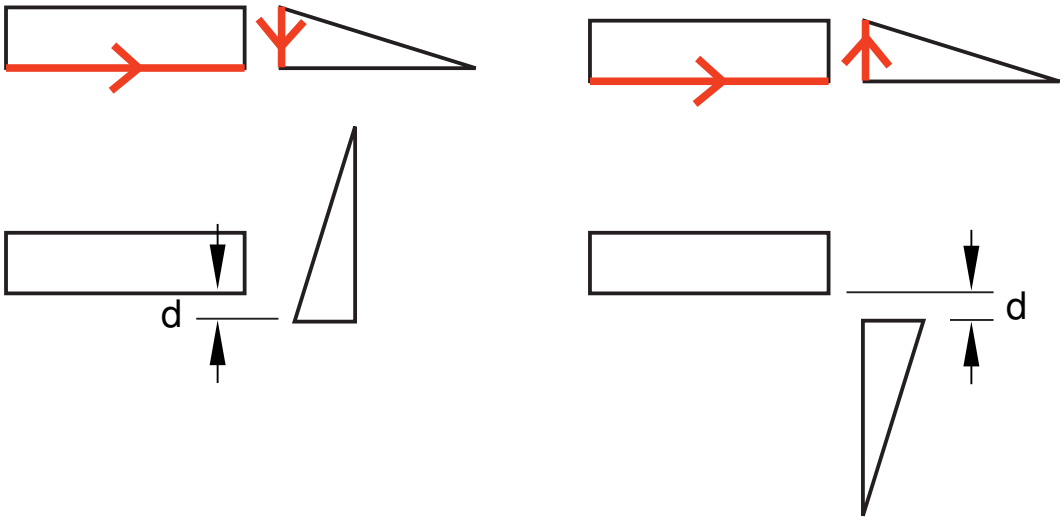
   - Click **OK** to accept the direction of the arrow along the edge of the movable instance.
   - Click **Flip** to reverse the direction of the arrow along the edge of the movable instance and click **OK**.

   The effect of changing the direction of the arrow is illustrated in the next step.

   If the instances are three-dimensional, Abaqus/CAE positions the movable instance so that the selected edges are parallel and coincident.

4. If the instances are two-dimensional, you must specify the clearance between the selected edges. In the text field that appears in the prompt area, enter the distance from the edge of the movable instance to the edge of the fixed instance, positive along the normal to the edge of the fixed instance.

   Abaqus/CAE positions the movable instance so that the two edges are parallel and the arrows point in the same direction. In addition, the movable instance is translated to satisfy the clearance specified. The orientation of the fixed instance remains unchanged. The effect of specifying the distance and changing the direction of the arrow is illustrated with two-dimensional instances in the following figure:



If the edge-to-edge constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the edge-to-edge constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Constraining two instances with coaxial faces

Select **Constraint->Coaxial** from the main menu bar to apply a constraint that positions a movable instance so that the axis of revolution of a selected face is coincident with the axis of revolution of a selected face of a fixed instance. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset.

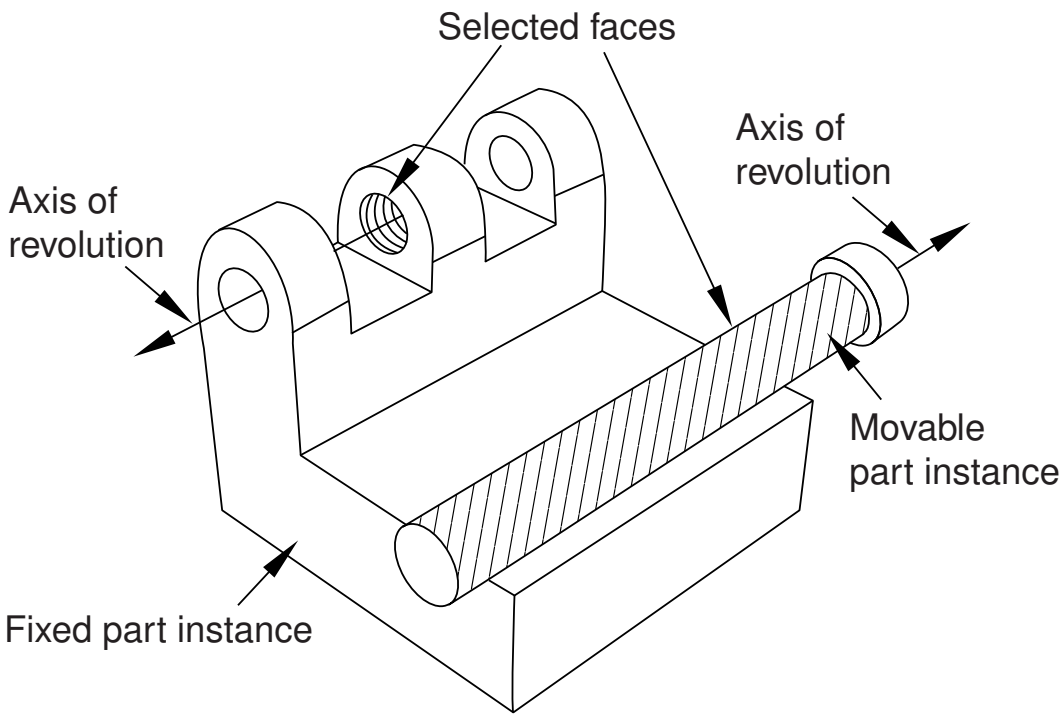
The selected faces of the movable and fixed instances must be either cylindrical or conical. In addition, the coaxial constraint can be applied only to three-dimensional instances. For more information, see *How the position constraint methods differ*.

1. From the main menu, select **Constraint->Coaxial**.

   Tip: You can also apply the coaxial constraint using the tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select cylindrical or conical faces from the instance that will move and the instance that will remain fixed, as shown in the following figure:

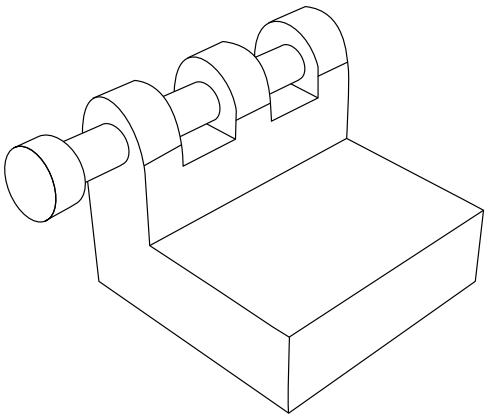


   Abaqus/CAE displays arrows along the axis of revolution of the selected faces.

3. From the buttons in the prompt area, do one of the following:

   - Click **OK** to accept the direction of the arrow along the axis of revolution of the face of the movable instance.
   - Click **Flip** to reverse the direction of the arrow along the axis of revolution of the face of the movable instance, and click **OK**.

Abaqus/CAE positions the movable instance so that the two axes are coincident and the arrows point in the same direction. The orientation of the fixed instance remains unchanged. The effect of the coaxial constraint with the arrows selected as shown above is illustrated in the following figure:



If the coaxial constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the coaxial constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*

Constraining two instances with coincident points

Select **Constraint->Coincident Point** from the main menu bar to apply a constraint that positions a movable instance so that a selected point is coincident with a selected point of a fixed instance. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset.

1. From the main menu, select **Constraint->Coincident Point**.

   Tip: You can also apply the coincident point constraint using the  tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a point from the instance that will move and a point from the instance that will remain fixed. In addition to selecting a vertex or a midpoint, you can select a datum point, a reference point, or the origin of a datum coordinate system.

   Abaqus/CAE moves the movable instance so that the selected points are coincident.

   When Abaqus/CAE prompts you to select the point from the fixed instance, you can select a datum point or a reference point that was created in either the Part or Assembly module. In contrast, when you select the point from a movable part instance, you can only select a datum point or reference point that was created in the Part module.

   If the coincident point constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the coincident point constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Constraining two instances with parallel coordinate systems

Select **Constraint->Parallel CSYS** from the main menu bar to apply a constraint that positions a movable instance so that a selected datum coordinate system is parallel to a selected datum coordinate system of a fixed instance. All position constraints are features of the assembly and can be suppressed or deleted using the Feature Manipulation toolset.

1. From the main menu, select **Constraint->Parallel CSYS**.

   Tip: You can also apply the parallel coordinate systems constraint using the  tool in the Assembly module toolbox. For a diagram of the tools in the Assembly toolbox, see *Using the Assembly module toolbox*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select a datum coordinate system from the movable instance and a datum coordinate system from the fixed instance.

   When Abaqus/CAE prompts you to select the datum coordinate system from the fixed instance, you can select a datum coordinate system that was created in either the Part or Assembly module. In contrast, when you select the datum coordinate system from a movable part instance, you can only select a datum coordinate system that was created in the Part module.

   Abaqus/CAE rotates the movable instance so that the selected coordinate systems are parallel.

   If the parallel coordinate systems constraint conflicts with existing constraints, Abaqus/CAE displays an error message and cancels the operation. To avoid the conflict, you can try reversing the selection of the instance that will move and the instance that will remain fixed. Alternatively, you can convert the existing constraints to an absolute position and reapply the parallel coordinate systems constraint.

Additional information

- *Using the Constraint menu*
- *Creating the assembly*
- *Converting constraints*

Using the Query toolset to query the assembly

Select **Tools->Query** from the main menu bar to start the Query toolset. You can use the Query toolset to request either general information or module-specific information. For a discussion of the information displayed by general queries, see *Obtaining general information about the model*. In addition, you can use the Assembly module-specific tools in the Query toolset to determine the attributes and position of a selected part or model instance.

1. From the main menu bar, select **Tools->Query**.

   
   Tip: You can also select the  tool in the Query toolset.

   Abaqus/CAE displays the **Query** dialog box.

2. From the **Query** dialog box, click on one of the following:

Instance Attributes

Select a part or model instance.

Abaqus/CAE displays the following in the message area:

- The name, modeling space, and type (deformable or rigid, dependent or independent) of the instance

Instance Position

Select a part or model instance.

Abaqus/CAE displays the following in the message area:

- Position of the origin of the instance relative to the global coordinate system
- The sum of the rotations applied to the instance relative to the assembly's global coordinate system
- A list of the constraints applied to the instance

3. Close the **Query** dialog box when you are done querying the assembly.

Additional information

• *Creating the assembly*

The Step module

You can use the Step module to create and define the analysis steps and associated output requests.

In this section:

- *Understanding the role of the Step module*
- *Entering and exiting the Step module*
- *Understanding steps*
- *Understanding output requests*
- *Understanding integrated, restart, diagnostic, and monitor output*
- *Understanding ALE adaptive meshing*
- *How can I customize the Abaqus analysis controls?*
- *Using the Step module toolbox*
- *Using the Step Manager*
- *Using the step editor*
- *Configuring analysis procedure settings*
- *Defining output requests*
- *Requesting specialized output*
- *Customizing ALE adaptive meshing*
- *Customizing the Abaqus analysis controls*

Understanding the role of the Step module

You can use the Step module to perform the following tasks:

Create analysis steps

Within a model you define a sequence of one or more analysis steps. The step sequence provides a convenient way to capture changes in the loading and boundary conditions of the model, changes in the way parts of the model interact with each other, the removal or addition of parts, and any other changes that may occur in the model during the course of the analysis. In addition, steps allow you to change the analysis procedure, the data output, and various controls. You can also use steps to define linear perturbation analyses about nonlinear base states. You can use the replace function to change the analysis procedure of an existing step.

Specify output requests

Abaqus writes output from the analysis to the output database; you specify the output by creating output requests that are propagated to subsequent analysis steps. An output request defines which variables will be output during an analysis step, from which region of the model they will be output, and at what rate they will be output. For example, you might request output of the entire model's displacement field at the end of a step and also request the history of a reaction force at a restrained point.

Specify adaptive meshing

You can define adaptive mesh regions and specify controls for adaptive meshing in those regions.

Specify analysis controls

You can customize general solution controls and solver controls.

Entering and exiting the Step module

You can enter the Step module at any time during an Abaqus/CAE session by clicking **Step** in the **Module** list located in the context bar. The **Step**, **Output**, **Other**, and **Tools** menus appear on the main menu bar. If the current viewport contains something other than the assembly, the contents of the viewport disappear when you start the Step module.

To exit the Step module, select any other module from the **Module** list. You need not save your steps or output requests before exiting the module; they will be saved automatically when you save the model database by selecting **File**->**Save** or **File**->**Save As** from the main menu bar.

Understanding steps

This section gives an overview of steps.

For more information on steps, see *Defining an Analysis*.

In this section:

- *What is a step?*
- *Linear and nonlinear procedures*
- *Step sequence restrictions*
- *What is step replacement?*
- *Replacing an Abaqus/Standard procedure with an Abaqus/Explicit procedure or vice versa*

What is a step?

An Abaqus/CAE model uses the following two types of steps:

The initial step

Abaqus/CAE creates a special initial step at the beginning of the model's step sequence and names it `Initial`. Abaqus/CAE creates only one initial step for your model, and it cannot be renamed, edited, replaced, copied, or deleted.

The initial step allows you to define boundary conditions, predefined fields, and interactions that are applicable at the very beginning of the analysis. For example, if a boundary condition or interaction is applied throughout the analysis, it is usually convenient to apply such conditions in the initial step. Likewise, when the first analysis step is a linear perturbation step, conditions applied in the initial step form part of the base state for the perturbation.

Analysis steps

The initial step is followed by one or more analysis steps. Each analysis step is associated with a specific procedure that defines the type of analysis to be performed during the step, such as a static stress analysis or a transient heat transfer analysis. You can change the analysis procedure from step to step in any meaningful way, so you have great flexibility in performing analyses. Since the state of the model (stresses, strains, temperatures, etc.) is updated throughout all general analysis steps, the effects of previous history are always included in the response for each new analysis step.

There is no limit to the number of analysis steps you can define, but there are restrictions on the step sequence. (For more information, see *Step sequence restrictions*.)

You use items from the **Step** menu to create a step, to select and define the analysis procedure used during the step, and to manage existing steps. Alternatively, you can select **Step->Manager** from the main menu bar to display the **Step Manager**.

For example, consider the following analysis of a section of a piping system:

Initial Step:

Apply boundary conditions to fix the left end of the pipe and to allow only axial movement at the right end.

Step 1: Compress

Apply a compressive force to the right end of the pipe. This step is a general analysis step.

Step 2: Eigenmodes

Calculate the frequencies and modes of vibration of the pipe in its compressed state. This step is a linear perturbation step.

Figure 1 shows the **Step Manager** after you create these steps.

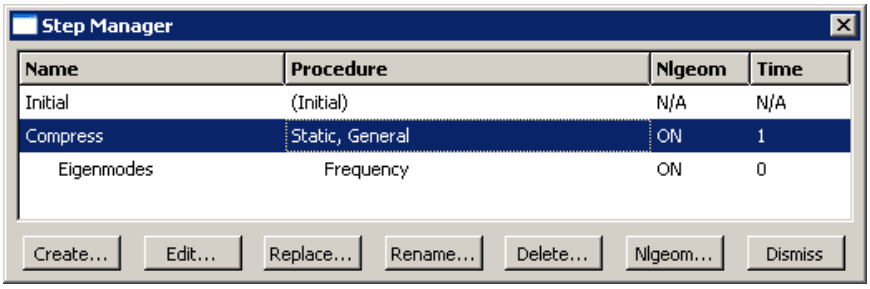


Figure 1: The Step Manager.

The manager lists all of the steps in the analysis as well as a few salient details concerning each step. Step 2, `Eigenmodes`, is indented to show that it is a linear perturbation step based on the state of the model at the end of Step 1, `Compress`.

For detailed information on creating, editing, and replacing steps, see the following sections:

- *The Step Manager*
- *Creating a step*
- *Editing a step*
- *Replacing a step*
- *Resetting the default values in the step editor*
- *The step editor*
- *The Incrementation tab*

Additional information

- *Understanding steps*
- *Defining an Analysis*

Linear and nonlinear procedures

The **Step Manager** distinguishes between general nonlinear steps and linear perturbation steps by indenting the names and procedure descriptions of linear perturbation steps. General nonlinear analysis steps define sequential events: the state of the model at the end of one general step provides the initial state for the start of the next general step. Linear perturbation analysis steps provide the linear response of the model about the state reached at the end of the last general nonlinear step. You use the **Procedure type** field to choose between **General** and **Linear perturbation** steps when you select the procedure in the **Create Step** dialog box.

For each step in the analysis the **Step Manager** also indicates whether Abaqus will account for nonlinear effects from large displacements and deformations. If the displacements in a model due to loading are relatively small during a step, the effects may be small enough to be ignored. However, in cases where the loads on a model result in large displacements, nonlinear geometric effects can become important. The **Nlgeom** setting for a step determines whether Abaqus will account for geometric nonlinearity in that step.

The **Nlgeom** setting is turned on by default for Abaqus/Explicit steps and turned off by default for Abaqus/Standard steps. The sequence of steps and the current **Nlgeom** setting determine whether you can change the **Nlgeom** setting in a particular step. For example, if Abaqus is already accounting for geometric nonlinearity, the **Nlgeom** setting is toggled on for all subsequent steps, and you cannot toggle it off. Where permissible, the following methods allow you to change the **Nlgeom** setting for a step:

- Click the **Basic** tab in the **Step Editor**, and toggle the **Nlgeom** setting.
- Select **Step->Nlgeom** from the main menu bar.
- Click **Nlgeom** in the **Step Manager**.

For more information, see *Accounting for geometric nonlinearity*, or see *General and Perturbation Procedures*.

Additional information

• *Understanding steps*

Step sequence restrictions

When you select **Step->Create** from the main menu bar, a **Create Step** dialog box appears in which you can specify the procedure type for the step that you are creating. Similarly, when you select **Step->Replace** from the main menu bar, a **Replace Step** dialog box appears in which you can specify a new procedure type for an existing step. The selection of procedure types in the **Create Step** and **Replace Step** dialog boxes depends on the following:

- The model type.
- The procedures that you have already associated with existing steps.
- The position of the new or replaced step in the analysis step sequence.

For example, when you create the first step in an analysis, you can choose from a list of valid procedure types; both Abaqus/Standard and Abaqus/Explicit procedure types appear in the list. However, once you have created the first step, the list of valid procedure types in the **Create Step** dialog box will change to include only those procedures that are compatible with the first step. For example, if the first step is an Abaqus/Standard step, Abaqus/Explicit procedures no longer appear in the list.

What is step replacement?

After you have defined your model and performed an analysis, you may want to run another analysis using a different procedure without having to redefine objects in your model, such as loads, boundary conditions, and interactions. You can use the replace function to replace the analysis procedure for an existing step with any procedure that is allowed by Abaqus/Standard or Abaqus/Explicit; for example, you can change from a **Static, General** procedure to a **Dynamic, Explicit** procedure or from a **Static, General** procedure to a **Static, Riks** procedure. After you select **Step->Replace** from the main menu bar, you select the step that you want to replace and the new analysis procedure for that step. The **Edit Step** dialog box appears with default values for the new analysis procedure. You can modify the default values and specify values for optional settings in the step editor.

When you replace a step, Abaqus/CAE copies all of the compatible step-dependent objects to the new step. If objects are incompatible with the new step, Abaqus/CAE substitutes an equivalent object, if possible, and suppresses or deletes the remaining objects. Therefore, you may want to copy the model before you replace the step. Abaqus/CAE displays a list of the objects that were suppressed or deleted during step replacement in the message area. For example, if you replace a **Static, General** procedure containing an Abaqus/Standard self-contact interaction, a pressure load, and an inertia relief load with a **Dynamic, Explicit** procedure, Abaqus/CAE does the following:

- Substitutes an Abaqus/Explicit self-contact interaction for the Abaqus/Standard self-contact interaction in the **Dynamic, Explicit** procedure.
- Copies the pressure load to the **Dynamic, Explicit** procedure.
- Suppresses the inertia relief load. Inertia relief loads apply only in Abaqus/Standard procedures.

After you replace a step, you should verify that previously defined properties, element types, jobs, and boundary conditions and predefined fields in the initial step remain valid for the model. In the Job module you can click **Write Input** in the **Job Manager** to write the input file and then check the input file for errors.

You can use the replace function to reset step settings to their default values by replacing an existing step with a step of the same procedure type.

Additional information

- *Suppressing and resuming objects*
- *Understanding steps*
- *Step sequence restrictions*
- *Replacing a step*
- *Resetting the default values in the step editor*
- *Writing the input file only*

Replacing an Abaqus/Standard procedure with an Abaqus/Explicit procedure or vice versa

If you want to replace an Abaqus/Standard analysis procedure with an Abaqus/Explicit analysis procedure or vice versa, you must have only one analysis step in the model for the desired procedure type to appear in the **Replace Step** dialog box. If your model contains multiple steps, you can use step-dependent managers to move objects to a single step. You can then delete the other steps and replace the remaining step with the new analysis procedure.

For example, if you want to change a model that contains four **Static, General** procedures from an Abaqus/Standard analysis to an Abaqus/Explicit analysis, you can use the **Load Manager** to move all of the loads into one of the four steps. Similarly, you can use the **Interaction Manager** to move the interactions. You can then delete the other three steps and replace the remaining step with a **Dynamic, Explicit** procedure. If desired, you can create additional Abaqus/Explicit steps and use the step-dependent managers to move objects that were copied during step replacement to the appropriate Abaqus/Explicit procedures.

For more information, see *Modifying the history of a step-dependent object*.

Additional information

- *Changing the status of an object in a step*
- *Step sequence restrictions*
- *What is step replacement?*
- *Replacing a step*

Understanding output requests

This section gives an overview of output requests.

In this section:

- *What is an output request?*
- *What is the difference between field output and history output?*
- *Propagation of output requests*
- *The output request managers*
- *Creating and modifying output requests*

What is an output request?

The Abaqus analysis products compute the values of many variables at every increment of a step. Usually you are interested in only a small subset of all of this computed data. You can specify the data that you want written to the output database by creating output requests. An output request consists of the following information:

- The variables or variable components of interest.
- The region of the model and the integration points from which the values are written to the output database.
- The rate at which the variable or component values are written to the output database.

When you create the first step, Abaqus/CAE selects a default set of output variables corresponding to the step's analysis procedure. By default, output is requested from every node or integration point in the model and from default section points. In addition, Abaqus/CAE selects the default rate at which the variables are written to the output database. You can edit these default output requests or create and edit new ones.

Default output requests and output requests that you modified are propagated to subsequent steps in the analysis. If you have a large model that includes the default output requests and requests output from a large number of frames, the resulting output database will be very large. You can use a C++ program to extract data from a large output database and copy only selected frames to a second output database. For more information, see *Decreasing the amount of data in an output database by retaining data at specific frames*.

When your analysis is complete, you use the Visualization module to read the output database and graphically display the data that were written to it.

For detailed instructions on creating and editing output requests, see the following sections:

- *Creating an output request*
- *Modifying field output requests*
- *Modifying history output requests*

Additional information

- *Understanding output requests*

What is the difference between field output and history output?

When you create an output request, you can choose either field output or history output.

Field output

Abaqus generates field output from data that are spatially distributed over the entire model or over a portion of it. In most cases you use the Visualization module to view field output data using deformed shape, contour, or symbol plots. The amount of field output generated by Abaqus during an analysis is often large. As a result, you typically request that Abaqus write field data to the output database at a low rate; for example, after every step or at the end of the analysis.

When you create a field output request, you can specify the output frequency in equally spaced time intervals or every time a particular length of time elapses. For an Abaqus/Standard analysis procedure, you can alternatively specify the output frequency in increments, request output after the last increment of each step, or request output according to a set of time points. For an Abaqus/Explicit analysis procedure, you can alternatively request field output for every time increment or according to a set of time points.

When you create a field output request, Abaqus writes every component of the selected variables to the output database. For example, if you were using solid elements to model a cantilever beam with a load at the tip, you could request the stress (all six components) and the displacement (all six components) data from the entire model after the last increment of the loading step. You could then use the Visualization module to view a contour plot of stresses and deflections in the final loaded state.

History output

Abaqus generates history output from data at specific points in a model. In most cases you use the Visualization module to display history output using *X*–*Y* plots. The rate of output depends on how you want to use the data that are generated by the analysis, and the rate can be very high. For example, data generated for diagnostic purposes may be written to the output database after every increment. You can also use history output for data that relate to the model or a portion of the model as a whole; for example, whole model energies.

When you create a history output request, you can specify the output frequency in equally spaced time intervals or every time a particular length of time elapses. For an Abaqus/Standard analysis procedure, you can alternatively specify the output frequency in increments, request output after the last increment of each step, or request output according to a set of time points. For an Abaqus/Explicit analysis procedure, you can alternatively request history output in time increments.

When you create a history output request, you can specify the individual components of the variables that Abaqus/CAE will write to the output database. For example, if you model the response of a cantilever beam with a load applied to the tip, you might request the following output after each increment of the loading step:

- The principal stress at a single node at the root of the beam.
- The vertical displacement at a single node at the tip of the beam.

You could then use the Visualization module to view an *X*–*Y* plot of stress at the root versus displacement at the tip with increasing load.

Propagation of output requests

When you create the first step in the analysis, Abaqus/CAE generates default field and history output requests based on the analysis procedure that you selected for the step. These default output requests are propagated to subsequent steps. The **Field Output Requests Manager** and the **History Output Requests Manager** are step-dependent managers that display the propagation and the status of output requests between steps.

The output requested in a general step is independent of the output requested in a linear perturbation step. In addition, the propagation behavior of output requests varies between general steps and linear perturbation steps.

General steps

Abaqus/CAE creates a default field output request for the first general step in your model, and that default output request propagates to all subsequent general steps. Similarly, if you create a new output request or modify the default output request, the new or modified request is propagated to subsequent general steps.

If you insert a new general step into the sequence of steps, the output request from the previous general step propagates to the new step.

Linear perturbation steps

Abaqus/CAE creates a default field output request for the first linear perturbation step in your model, and that default output request propagates to all subsequent linear perturbation steps that use the same analysis procedure; for example, all the frequency analyses. Similarly, if you create a new output request or modify the default output request, the new or modified request is propagated to subsequent steps that use the same analysis procedure.

If you insert a new linear perturbation step into the sequence of steps, the output request from the previous linear perturbation step that uses the same analysis procedure propagates to the new step. If you create a linear perturbation step that uses a different analysis procedure, Abaqus/CAE creates a new default output request. The new default output request propagates to all subsequent linear perturbation steps that use the same analysis procedure.

You should be aware of the following behavior:

- If you insert a new general step at the beginning of a sequence of existing general steps, Abaqus/CAE does not create a default output request for the new step. Similarly, if you insert a new linear perturbation step at the beginning of a sequence of existing linear perturbation steps of the same procedure type, Abaqus/CAE does not create a default output request for the new step. In both cases you must create a new output request for the new step. Alternatively, you can use the output request managers to move the output request from the following step to the new step.
- If you delete a step (general or linear perturbation) that contains a new output request, Abaqus/CAE deletes the output request from all subsequent steps into which the request had propagated.
- If a step does not contain an output request, Abaqus/CAE displays a warning in the Job module when the input file is generated.

The output request managers

Abaqus/CAE provides separate managers for field output requests and history output requests. The output request managers are step-dependent managers, which means that they contain information concerning the status of each output request in each step of the analysis and allow you to control the propagation of requests across the sequence of steps. For more information, see *What are step-dependent managers?*.

The **Field Output Requests Manager** and the **History Output Requests Manager** contain lists of all of the output requests that you have created. For example, the **Field Output Requests Manager** is shown in *Figure 1*.

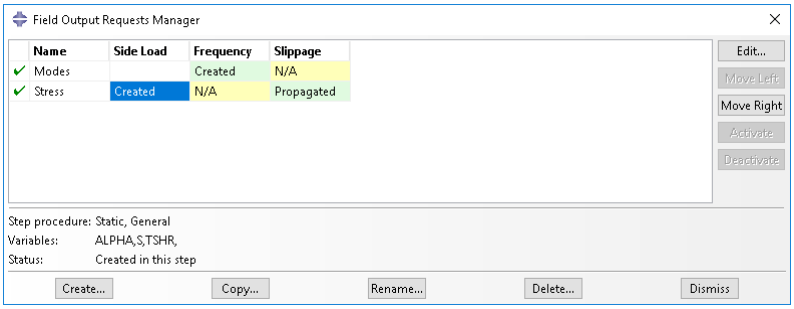


Figure 1: The Field Output Requests Manager.

After you select the step, the **Create** button in the two managers allows you to create a new output request during that step. Similarly, the **Edit**, **Copy**, **Rename**, and **Delete** buttons allow you to edit, copy, rename, and delete the selected output request. You can also initiate the create, edit, copy, rename, and delete procedures using the **Output->Field Output Requests** and **Output->History Output Requests** submenus in the main menu bar.

You can use the **Copy** button in the **Field Output Requests Manager** and the **History Output Requests Manager** (or the corresponding menu commands or Model Tree) to copy an output request. You can copy an output request from any step to any valid step, with some restrictions. For more details, see *Copying step-dependent objects using manager dialog boxes*.

The **Move Left**, **Move Right**, **Activate**, and **Deactivate** buttons allow you to control the propagation of output requests over the course of an analysis. For more information, see *Modifying the history of a step-dependent object*.

You can use the icons in the column along the left side of the managers to suppress output requests or to resume previously suppressed output requests for an analysis. The suppress and resume procedures are also available from the **Output->Field Output Requests** and **Output->History Output Requests** submenus in the main menu bar. For more information, see *Suppressing and resuming objects*.

Additional information

- *What are step-dependent managers?*
- *What is the difference between field output and history output?*

Creating and modifying output requests

To create an output request, select **Output->Field Output Requests->Create** or **Output->History Output Requests->Create** from the main menu bar.
An editor appears in which you can enter all of the information necessary to define the output request. The top of the editor displays the following:

- The name of the output request.
- The name of the step in which you are creating or modifying the output request.
- The name of the analysis procedure associated with the step.

For example, the **Field Output Request** editor is shown in *Figure 1*.

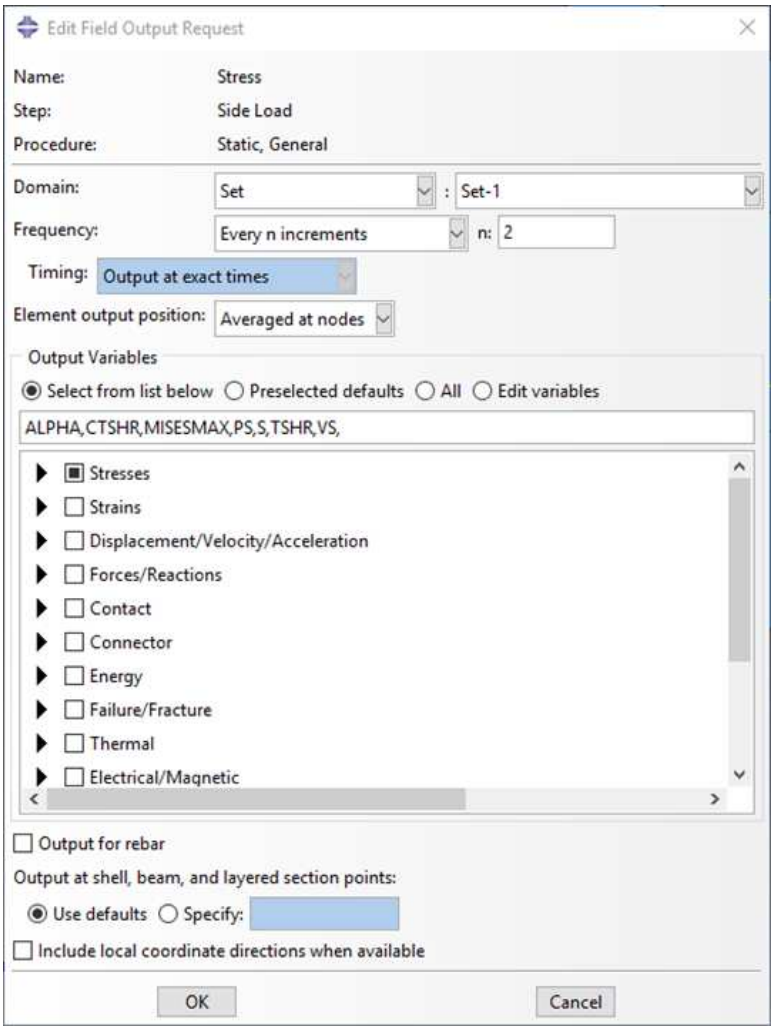


Figure 1: The Field Output Request editor.

The **Domain** section of the editor allows you to choose the region from which output will be generated. You can request that Abaqus write field data to the output database for the following:

- Whole model
- Whole model, only exterior nodes and elements (three-dimensional models in Abaqus/Standard or Abaqus/Explicit analyses)

- A set
- A bolt load
- A skin
- A stringer
- A fastener
- An assembled fastener set
- An interaction
- A composite layup
- A substructure

Similarly, you can request that Abaqus write history data to the output database for the following:

- Whole model
- A set
- A bolt load
- A skin
- A stringer
- A fastener
- An assembled fastener set
- A contour integral
- A general contact surface (Abaqus/Explicit steps only)
- An integrated output section (Abaqus/Explicit steps only)
- An interaction
- Springs/dashpots
- A composite layup

The **Frequency** section of the editor allows you to specify the frequency at which the output is written to the output database. Choose one of the following:

- **Last increment** to request output only after the last increment of the step. This output frequency is available only when you choose an Abaqus/Standard analysis procedure.
- **Every *n* increments** to request output after a specified number of increments. If you specify the frequency in increments, Abaqus also writes output after the last increment of the step. This output frequency is available when you choose an Abaqus/Standard analysis procedure.
- **Every time increment** to request output at every time increment. This output frequency is available for field output when you choose an Abaqus/Explicit analysis procedure.
- **Every *n* time increments** to request output at a specified number of time increments. This output frequency is available for history output when you choose an Abaqus/Explicit analysis procedure.
- **Evenly spaced time intervals** to request output at a number of evenly spaced time intervals.
- **Every *x* units of time** to request output after a particular length of time elapses.
- **From time points** to request output according to a set of time points that you specify. This output frequency is available for field and history output when you choose an Abaqus/Standard analysis procedure and for field output when you choose an Abaqus/Explicit analysis procedure.

The **Element output position** section of the editor allows you to choose the position where selected field output values are written. Choose one of the following:

- **Averaged at nodes** (Abaqus/Standard steps only)
- **Centroidal**
- **Integration points** (default)
- **Nodes**

The **Output Variables** section of the editor contains a list of the variable categories that are applicable to the step procedure and the selected domain. Choose one of the following:

- **Select from list below** to request the variables from the list of check boxes below. You can click the check box next to a category name to select all of the variables within that category, or you can click the arrow next to a category name to display the list of variables in that category and then select individual variables.
- **Preselected defaults** to request the default output variables for the procedure.
- **All** to request all output variables for the procedure.
- **Edit variables** to request variables from the text field below. You can manually edit this field and type or delete variable names.



Note:

In addition to the current analysis procedure, other aspects of the model such as the region specified might affect the output variables. For example, if an output variable is valid for the analysis procedure but is not valid for the element type used in the mesh, Abaqus will remove that variable during the analysis.

If you use the **Field Output Request** editor to select a vector or tensor variable to be included in a field output request, Abaqus automatically writes all components of that variable to the output database during the step. For example, if you select the vector U in a three-dimensional model, Abaqus outputs the three displacement components U1, U2, and U3 to the output database along with the three rotation components UR1, UR2, and UR3.

In contrast, if you use the **History Output Request** editor to select a vector or tensor variable to be included in a history output request, the **History Output Request** editor allows you to select individual components of the variable. It is useful to specify individual components in a history output request because these variables are typically output very frequently—possibly as often as every increment.

If your model contains rebar, you must toggle on **Output for rebar** to include rebar output in the data that Abaqus writes to the output database and to view plots of the rebar orientations in the Visualization module. For more information, see *Understanding rebar in shell sections*.

The editor also allows you to specify the section points from which output will be obtained. If you request output from a composite layup, you can specify the section points from which output will be obtained for each ply of the layup. For more information, see *Requesting output from a composite layup*.

For example, in *Figure 1* the user is editing a field output request that is associated with a **Static, General** analysis procedure. The user has selected all of the variables in the **Stresses** category. These variables will be included in the output request during the step named `Side Load`. Abaqus will write output from the default section points at every increment.

For detailed instructions on selecting output variables and components, see the following sections:

- *Modifying field output requests*
- *Modifying history output requests*

Once you have created an output request, you can modify it in the following ways:

- Select **Output->Field Output Requests->Edit** or **Output->History Output Requests->Edit** to display the field or history output request editor.
- Select **Output->Field Output Requests->Manager** or **Output->History Output Requests->Manager** to display the field or history output requests manager. Use the manager to modify the stepwise history of the output request. (See *What are step-dependent managers?*, for more information.)

If you modify an output request during the step in which you created the request, you can modify the domain, the output variables, the output for rebar option, the section points, and the output frequency. However, if you modify an output request during a step into which it was propagated, you can modify only the output variables and the output frequency.

When you request output from a contour integral, the **History Output Request** editor allows you to select only the frequency of output, the number of contour integrals, and the type of contour integral calculation. For more information, see *Requesting contour integral output*.

Additional information

• *Understanding output requests*

Understanding integrated, restart, diagnostic, and monitor output

This section explains the additional output controls available in the Step module.

In this section:

- *Integrated output requests*
- *Restart output requests*
- *Diagnostic printing*
- *Degree of freedom monitor requests*

Integrated output requests

To obtain history output of variables such as the forces summed over an exterior surface in contact or transmitted through a tie constraint between surfaces, you must refer to an integrated output section to identify the surface where output is needed.
(See *Integrated Output*.) In addition, the integrated output section definition can provide a local coordinate system in which to express the vector output quantities and/or a reference node as an anchor point about which the total moment across the surface is computed.

By default, an integrated output section is anchored at the global origin and does not follow the motion of the surface on which it is defined. You can define a reference point at which the output section is anchored and specify how this reference point tracks with the average motion of the surface. The reference point must not be connected to any other part of the finite element model.

Integrated output sections associated with a coordinate system and/or a reference node can be used independent of integrated output requests to track the average motion of a surface.

You define integrated output sections by selecting **Output->Integrated Output Sections->Create** from the main menu bar. For detailed instructions, see *Defining integrated output sections*. For information on requesting output for an integrated output section, see *Modifying history output requests*.

Additional information

- *Defining integrated output sections*
- *Understanding output requests*
- *Understanding integrated, restart, diagnostic, and monitor output*

Restart output requests

You can use the restart files created by Abaqus to continue an analysis from a specified step of a previous analysis. This section describes how you control the output of restart data.
For a discussion of how you use the restart data in a subsequent job, see *Restarting an analysis*, and *What are the model attributes?*.

By default, no restart information is written for an Abaqus/Standard analysis and restart information is written only at the beginning and end of each step for an Abaqus/Explicit analysis. However, default restart requests are created automatically for every step in an analysis. The **Edit Restart Requests** dialog box, invoked by selecting **Output->Restart Requests** from the main menu bar in the Step module, allows you to specify how often you want the restart information to be written.

You can specify the frequency at which Abaqus writes data to the restart files; however, the behavior of restart differs between analysis products.

Abaqus/Standard

You can request the frequency in increments or in time intervals. For an Abaqus/Standard step, you can choose whether the output is written at the exact time interval or at the closest approximation.

Abaqus/Explicit

For an Abaqus/Explicit analysis, you specify the number of equally spaced time intervals at which Abaqus writes data to the restart files. In addition, for an Abaqus/Explicit step you can choose whether the output is written at the exact time interval or at the closest approximation. However, you cannot avoid writing information to the restart files for Abaqus/Explicit steps; the number of time intervals must be set to one or greater.

For an Abaqus/Standard or an Abaqus/Explicit analysis, you can request that data written to the restart files overlay data from the previous increment. If you select this option, Abaqus retains the information from only one increment of each step in the restart files, thus minimizing the size of the files. By default, Abaqus does not overlay data.

For more information, see *Restarting an analysis*, and *Restarting an Analysis*. For detailed instructions on requesting restart output, see *Configuring restart output requests*.

You can use the **abaqus restartjoin** execution procedure to extract data from the output database created by a restart analysis and append the data to a second output database. For more information, see *Joining Output Database (.odb) Files from Restarted Analyses*.

Additional information

- *Understanding output requests*
- *Understanding integrated, restart, diagnostic, and monitor output*

Diagnostic printing

If the analysis of your model fails or produces unexpected results, you can examine its iteration-by-iteration progress by looking at selected diagnostic information that is written to the following files:

For Abaqus/Standard analyses:

Diagnostic information is written to the message (`.msg`) file, and a subset of the information is written to the output database (`.odb`) file. You can view the diagnostic information in the output database in the Visualization module (for more information, see *Viewing diagnostic output*). By default, the information is written during every iteration; you can request that Abaqus discontinue writing diagnostic information to the message file by specifying an output frequency of zero.

For Abaqus/Explicit analyses:

Diagnostic information is written to the status (`.sta`) file. For information on the frequency at which this information is written, see *About Output*.

You display the **Edit Diagnostic Print** dialog box by selecting **Output->Diagnostic Print** from the main menu bar.

For detailed instructions on requesting diagnostic printing, see *Configuring diagnostic printing*.



Note:

Changes to the diagnostic print requests do not affect the diagnostic information written to the output database during Abaqus/Standard analyses.

Additional information

- *Understanding output requests*
- *Understanding integrated, restart, diagnostic, and monitor output*
- *Viewing diagnostic output*

Degree of freedom monitor requests

You can request that Abaqus write the values of a degree of freedom at one selected point to the status (`.sta`) file and, for Abaqus/Standard analyses, to the message (`.msg`) file at specific increments during the course of an analysis. In addition, a plot of the degree of freedom value over time appears in a new viewport that is generated automatically when you submit the analysis. (For more information, see *Monitoring the progress of an analysis job*.) You can use this information to monitor the progress of the solution.

You must specify the vertex or node you want to monitor by selecting an existing geometry or node set or by selecting a point in the viewport. Once you have specified the point, you must indicate which degree of freedom you want to monitor at that vertex or node, how often you want the information displayed in a viewport, and how often you want it printed to the status and message files.

For detailed instructions on monitoring a degree of freedom, see *Configuring monitor requests*.

Additional information

- *Configuring monitor requests*
- *Understanding output requests*
- *Understanding integrated, restart, diagnostic, and monitor output*

Understanding ALE adaptive meshing

Arbitrary Lagrangian-Eulerian (ALE) adaptive meshing allows you to maintain a high-quality mesh throughout an analysis, even when large deformations or losses of material occur, by allowing the mesh to move independently of the material. Adaptive meshing moves only nodes; the mesh topology remains unchanged. Adaptive meshing is available only for **Coupled temp-displacement**; **Dynamic, Explicit**; **Dynamic, Temp-disp, Explicit**; **Soils**; and **Static, General** steps.

You can define regions of the model where you want adaptive meshing by selecting **Other->ALE Adaptive Mesh Domain** from the main menu bar. If necessary, you can select **Other->ALE Adaptive Mesh Controls** or **Other->ALE Adaptive Mesh Constraint** to customize the adaptive mesh controls or to add regional adaptive mesh constraints, respectively. Currently, you can define only one ALE adaptive mesh domain for any particular step.

For detailed information on adaptive meshing, see *ALE Adaptive Meshing*.

For detailed instructions on defining adaptive mesh regions, see the following sections:

- *Defining an ALE adaptive mesh region*
- *Specifying ALE adaptive mesh constraints*
- *Specifying controls for ALE adaptive meshing*

How can I customize the Abaqus analysis controls?

This section explains how you can adjust the parameters that control the Abaqus analysis.

In this section:

- *General solution controls*
- *Solver controls*

General solution controls

You can customize the numerous variables that control the convergence and time integration accuracy algorithms in Abaqus. The default solution controls usually work well, but customizing these controls may result in a more cost-effective solution or help you to obtain a solution for particularly difficult analyses.



Note:

These options are available only for general Abaqus/Standard analysis steps.

You can access the solution controls by selecting **Other->General Solution Controls** from the main menu bar. For more information, see *Analysis Convergence Controls*.



Warning:

Solution controls are intended for experienced analysts and should be used with great care. The default settings of these controls are appropriate for most nonlinear analyses. Changing these values inappropriately may greatly increase the computational time of your analysis or produce inaccurate results.

For detailed instructions on setting general solution controls, see *Customizing general solution controls*.

Solver controls

You can customize the variables that control the iterative linear equation solver.



Note:

You can use the iterative linear equation solver only for **Static, General**; **Static, Linear perturbation**; **Visco**; **Heat transfer**; **Geostatic**; and **Soils** analysis steps.

You can access the solver controls by selecting **Other->Solver Controls** from the main menu bar. For more information, see *Iterative Linear Equation Solver*.

For detailed instructions on setting solver controls, see *Customizing solver controls*.

Using the Step module toolbox

You can access all the Step module tools through the main menu bar; in addition, you can also access the tools through the Step module toolbox. *Figure 1* shows the icons for the tools in the Step module toolbox.

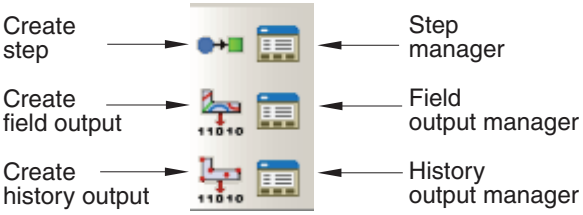


Figure 1: The Step module toolbox.

Using the Step Manager

This section describes how you can use the **Step Manager** to create, edit, and manipulate steps. (For general information on managers, see *Managing objects*.)

In this section:

- *The Step Manager*
- *Creating a step*
- *Editing a step*
- *Replacing a step*
- *Resetting the default values in the step editor*
- *Accounting for geometric nonlinearity*

The Step Manager

You use the **Step Manager** to create, edit, and manipulate the analysis steps associated with the current model. To start the **Step Manager**, select **Step->Manager** from the main menu bar. Columns in the **Step Manager** dialog box display the following information about each step:

Name

The name of the step. Names of linear perturbation steps are indented relative to names of general steps.

Procedure

The analysis procedure that you selected for this step when the step was created. You can change the analysis procedure after creating a step. Click **Replace** to select a new procedure type for the selected step. The **Procedure** column also indicates whether thermal and soils steps assume steady-state or transient conditions or if neither is applicable.

Nlgeom

Whether the analysis step accounts for geometric nonlinearities. You use the **Nlgeom** button to control the **Nlgeom** setting for a particular step. Once you have set the **Nlgeom** option for a step, your setting remains in effect for all subsequent steps.

Time

The time period for the step. The default value for the time period is 1.0 time unit. Click **Edit** to display the step editor so that you can modify the time period.

You use the buttons across the bottom of the **Step Manager** dialog box to create a step that follows the selected step or to manipulate the selected step. You use the **Dismiss** button to close the **Step Manager** dialog box. You can perform the same tasks using the pull-down menus available from the **Step** menu, located in the main menu bar.

You can suppress an analysis step to exclude the procedure from the analysis. The suppressed step is removed from the context bar, the restart request dialog box, and the diagnostic print dialog box. Any step-dependent or propagating attributes created in the step are automatically suppressed and ignored during the analysis. Upon resuming the step, the status of each attribute will return to the original state. For example, suppressing and resuming a step will not resume an associated load that was previously suppressed. You can suppress or resume a step as long as the step sequence remains valid.



Warning:

If you use the **Step Manager** or the **Step** menu to delete a step, objects associated with that step, such as prescribed conditions or output requests, are also deleted. If you use the **Step Manager** or the **Step** menu to replace a step, objects that are incompatible with the new analysis procedure are substituted with an equivalent object, if possible, or deleted.

Additional information

- *Suppressing and resuming objects*
- *Understanding steps*
- *Using the Step Manager*

Creating a step

You can create any sequence of procedures that is allowed by the Abaqus analysis products; the procedure list in the **Create Step** dialog box is updated to show only the available procedures for the new step. For example, if your first step contains a static stress/displacement procedure, you cannot follow it with a new step containing a heat transfer procedure.

1. From the main menu bar, select **Step->Create**.

   The **Create Step** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Step Manager**. (You can display the **Step Manager** by selecting **Step->Manager** from the main menu bar.)
   - Click the tool in the Step module toolbox.

2. If desired, use the **Name** text field to change the name of the new step.

   All steps must have unique names, and you cannot name a step “Initial”.

3. From the list of existing steps, select the step after which the new step will be inserted.
4. Click the arrow next to the **Procedure type** field, and select either **General** or **Linear perturbation** from the list that appears.

   The lower half of the dialog box displays a list of available procedures.

5. Select the desired procedure and click **Continue**.

   The **Edit Step** dialog box appears.

6. Use the **Edit Step** dialog box to modify the settings from their default values and to provide values for optional settings. (For detailed help on a particular editor feature, select **Help->On Context** from the main menu bar and then click the feature of interest.)
7. Click **OK**.

   Abaqus/CAE closes the **Edit Step** dialog box, and the new step appears in the **Step Manager**.

Additional information

- *Understanding steps*
- *General and Perturbation Procedures*

Editing a step

You can use the step editor to edit the analysis procedure settings associated with an existing step.

1. From the main menu bar, select **Step->Edit->*step name***.

   The step editor appears.

   Tip: You can also select the step name in the **Step Manager** and click **Edit**.

2. Use the tabs within the step editor to modify the settings. (For detailed help on a particular editor feature, select **Help->On Context** from the main menu bar and then click the feature of interest.)
3. Click **OK** to close the step editor and save the new settings.

Additional information

- *Understanding steps*

Replacing a step

You can replace an existing procedure with any procedure that is allowed by the Abaqus analysis products; the procedure list in the **Replace Step** dialog box is updated to show only the available procedures for the revised step. For example, you can change from a **Static, General** procedure to a **Static, Riks** procedure. Abaqus/CAE copies compatible step-dependent objects to the new step, substitutes equivalent objects, if possible, and deletes the remaining objects.

After you replace a step, you should verify that previously defined properties, element types, jobs, and boundary conditions and fields in the inital step remain valid for the model. For more information, see *What is step replacement?*.

1. From the main menu bar, select **Step->Replace->*step name***.

   The **Replace Step** dialog box appears.

   Tip: You can also select the step name in the **Step Manager** and click **Replace**.

2. Click the arrow next to the **New procedure type** field, and select either **General** or **Linear perturbation** from the list that appears.

   The lower half of the dialog box displays a list of available procedures.

3. Select the new procedure, and click **Continue**.

   The **Edit Step** dialog box appears.

4. Use the **Edit Step** dialog box to modify the settings from their default values and to provide values for optional settings. (For detailed help on a particular editor feature, select **Help->On Context** from the main menu bar and then click the feature of interest.)

5. Click **OK**.

   If step-dependent objects are not compatible with the new step, Abaqus/CAE displays a list of the objects that were deleted during step replacement in the message area and closes the **Edit Step** dialog box.

Additional information

- *Understanding steps*

Resetting the default values in the step editor

When you create, edit, or replace a step, you use the step editor to configure the analysis procedure settings. You can use the replace function to reset the settings in the step editor to their default values by replacing an existing step with a step of the same procedure type.

1. From the main menu bar, select **Step->Replace->*step name***.

   The **Replace Step** dialog box appears with the current procedure highlighted in the list of available procedures.

   Tip: You can also select the step name in the **Step Manager** and click **Replace**.

2. Click **Continue**.

   The **Edit Step** dialog box appears with default values for the procedure settings.

3. Use the **Edit Step** dialog box to modify the settings from their default values and to provide values for optional settings. (For detailed help on a particular editor feature, select **Help->On Context** from the main menu bar and then click the feature of interest.)

4. Click **OK**.

   Abaqus/CAE copies step-dependent objects to the new step and closes the **Edit Step** dialog box.

Accounting for geometric nonlinearity

The **Nlgeom** setting for a step determines whether Abaqus will account for geometric nonlinearity in that step. The **Nlgeom** setting is turned on by default for Abaqus/Explicit steps and turned off by default for Abaqus/Standard steps.

The sequence of steps and the current **Nlgeom** setting determine whether you can change the **Nlgeom** setting in a particular step. For example, if Abaqus is already accounting for geometric nonlinearity, the **Nlgeom** setting is toggled on for all subsequent steps, and you cannot toggle it off. Similarly, you cannot change the **Nlgeom** setting during a linear perturbation step. For more information, see *Linear and nonlinear procedures*.



Note:

When you create a step, you can click the **Basic** tab in the **Step Editor** and select **On** or **Off** as the **Nlgeom** setting.

1. To display the **Edit Nlgeom** dialog box and to change the setting where applicable, do one of the following:
   - From the main menu bar, select **Step->Nlgeom**.
   - From the main menu bar, select **Step->Edit->*step name***.

     The **Step Editor** appears. From the **Nlgeom** field on the **Basic** tabbed page, click .
   - From the main menu bar, select **Step->Manager**.

     The **Step manager** appears. From the buttons along the bottom of the manager, click **Nlgeom**.
2. From the **Edit Nlgeom** dialog box, click the step name of interest to turn **Nlgeom** on or off for that step.

   If **Nlgeom** is turned on for a step, a check mark appears in the **Nlgeom** column. If **Nlgeom** is turned off for a step, no tickmark appears.
3. Click **OK** to close the **Edit Nlgeom** dialog box.

Additional information

- *Understanding steps*

Using the step editor

This section describes the step editor and the options that appear in the step editor.

In this section:

- *The step editor*
- *The Incrementation tab*

The step editor

When you create, edit, or replace a step, the step editor displays a set of tabbed pages that allow you to configure the settings for the procedure you selected. The pages are unique for each procedure; for example, when you configure a **Static, General** procedure, the step editor displays the **Basic**, **Incrementation**, and **Other** tabs. Settings you can configure with these tabbed pages include the time period for the step, the maximum number of increments, the increment size, the default load variation with time, and whether to account for geometric nonlinearity.

Abaqus stores the text that you enter in the **Description** field on the **Basic** tabbed page in the output database, and it is displayed in the state block by the Visualization module.

If you want to reset the procedure settings to their default values, you can replace an existing step with a step of the same procedure type. For more information, see *Resetting the default values in the step editor*.

For detailed help on a specific feature of the editor, select **Help->On Context** and then click the feature of interest.

Additional information

- *Understanding steps*
- *Using the step editor*

The Incrementation tab

When you configure general procedures, you use the **Basic** tab in the step editor to enter the total time period for the step. You use the **Incrementation** tab to configure the approach that Abaqus will use to divide the total time period for the step into increments. For a general, static step as well as for many other kinds of steps you can set the following options on the **Incrementation** tabbed page:

Time incrementation

- When you choose **Automatic** time incrementation, Abaqus starts the incrementation using the value entered for the initial increment size. The size of subsequent time increments are adjusted based on how quickly the solution converges. This option is the default selection.
- When you choose **Fixed** time incrementation, Abaqus uses the value entered for the initial increment size throughout the step.



Warning:

Choosing **Fixed** time incrementation may prevent the solution from converging and is not recommended.

Maximum number of increments

Abaqus limits the number of increments in a step to the value that you enter for the maximum number of increments. If the step exceeds this number of increments, the analysis stops, and diagnostic information is reported to the Job module and written to the message file. By default, Abaqus/CAE sets the maximum number of increments to 100.

Initial increment size

Abaqus starts the step using the value entered for the initial increment size.

Minimum increment size

Abaqus checks for the minimum increment size only when you analyze your model using automatic time incrementation. If Abaqus needs a smaller time increment than this value to reach a convergent solution, it terminates the analysis, reports to the Job module, and writes diagnostic information to the message file. If you do not enter a minimum increment size, Abaqus uses 10^{-5} times the total time period.

Maximum increment size

Abaqus checks for the maximum increment size only when you analyze your model using automatic time incrementation. Abaqus will not increase the increment size beyond this value during the analysis. If you do not specify this value, Abaqus/CAE sets the value to that of the total time period (with the exception of dynamic, implicit procedures, in which the default maximum increment size depends on a variety of analysis settings; see *Configuring a dynamic, implicit procedure*).



Note:

A value must be entered for each of the incrementation options described above. Abaqus/CAE does not allow you to create the step if you delete the default value for an incrementation option but fail to provide another.

For detailed information on other items in the **Incrementation** tabbed page, click **Help->On Context** and then click the item of interest.

Additional information

- *Understanding steps*
- *Using the step editor*

Configuring analysis procedure settings

The **Edit Step** dialog box allows you to configure the analysis procedure settings for a particular step. This section provides instructions for each analysis procedure.

In this section:

- *Configuring general analysis procedures*
- *Configuring linear perturbation analysis procedures*

Configuring general analysis procedures

You can configure general analysis procedures to analyze linear or nonlinear response. You can include general analysis procedures in Abaqus/Standard or Abaqus/Explicit analyses.
For more information, see *General and Perturbation Procedures*.

This section provides instructions for using the step editor to configure different types of general analysis procedures.

In this section:

- *Configuring a static, general procedure*
- *Configuring a static, Riks procedure*
- *Configuring a dynamic, explicit procedure*
- *Configuring a heat transfer procedure*
- *Configuring a dynamic, implicit procedure*
- *Configuring a fully coupled, simultaneous heat transfer and stress procedure*
- *Configuring a fully coupled, simultaneous heat transfer and electrical procedure*
- *Configuring a fully coupled, simultaneous heat transfer, electrical, and structural procedure*
- *Configuring a direct cyclic procedure*
- *Configuring a dynamic fully coupled thermal-stress procedure using explicit integration*
- *Configuring a geostatic stress field procedure*
- *Configuring a mass diffusion procedure*
- *Configuring an effective stress analysis for fluid-filled porous media*
- *Configuring a transient, static, stress/displacement analysis with time-dependent material response*
- *Configuring an annealing procedure*

Configuring a static, general procedure

A static stress procedure is one in which inertia effects are neglected. The analysis can be linear or nonlinear and ignores time-dependent material effects. For more information, see *Static Stress Analysis*.

Create or edit a static, general procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Static, General`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, the maximum number of increments, the increment size, the default load variation with time, and whether to account for geometric nonlinearity as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Time period** field, enter the time period of the step. For more information, see *Time Period*.
4. Select an **Nlgeom** option:

   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
5. Select an automatic stabilization method if you expect the problem to have local instabilities such as surface wrinkling, material instability, or local buckling. Abaqus/Standard can stabilize this class of problems by applying damping throughout the model. For more information, see *Unstable Problems*, and *Automatic Stabilization of Static Problems with a Constant Damping Factor*

   Click the arrow to the right of **Automatic stabilization**, and select a method for defining the damping factor:

   - Select **Specify dissipated energy fraction** to allow Abaqus/Standard to calculate the damping factor from a dissipated energy fraction that you provide. Enter a value for the dissipated energy fraction in the adjacent field (the default is 2.0×10^{-4}). For more information, see *Calculating the Damping Factor Based on the Dissipated Energy Fraction*.
   - Select **Specify damping factor** to enter the damping factor directly. Enter a value for the damping factor in the adjacent field. For more information, see *Directly Specifying the Damping Factor*.
   - Select **Use damping factors from previous general step** to use the damping factors at the end of the previous step as the initial factors in the current step's variable damping scheme. These factors override any initial damping factors that are calculated or specified directly in the current step. If there are no damping factors associated with the previous general step (for example, if the previous step does not use any stabilization or the current step is the first step of the analysis), Abaqus uses adaptive stabilization to determine the required damping factors.
6. When using automatic stabilization, Abaqus can use the same damping factor over the course of a step, or it can vary the damping factor spatially and temporally during a step based on the convergence history and the ratio of the energy dissipated by damping to the total strain energy. For more information, see *Adaptive Automatic Stabilization Scheme*. If you selected **Specify dissipated energy fraction**, adaptive stabilization is optional and turned on by default. If you selected **Specify damping factor**,

adaptive stabilization is optional and turned off by default. If you selected **Use damping factors from previous general step**, adaptive stabilization is required.

To use adaptive stabilization, toggle on **Use adaptive stabilization with max. ratio of stabilization to strain energy** (if necessary), and enter a value in the adjacent field for the allowable accuracy tolerance for the ratio of energy dissipated by damping to total strain energy in each increment. The default value of 0.05 should be suitable in most cases.

7. Toggle on **Include adiabatic heating effects** if you are performing an adiabatic stress analysis. This option is relevant only for isotropic metal plasticity materials with a Mises yield surface. For more information, see *Adiabatic Analysis*.
8. When you have finished configuring settings for the static, general step, click **OK** to close the **Edit Step** dialog box.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:
   - Choose **Automatic** to allow Abaqus/Standard to choose the size of the time increments based on computational efficiency.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.
3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.
5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. When you have finished configuring settings for the static, general step, click **OK** to close the **Edit Step** dialog box.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose an **Equation Solver Method** option:
   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.
3. Choose a **Matrix storage** option:

- Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
- Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
- Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

4. Choose a **Solution technique**:

   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

     If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

     For more information, see *Quasi-Newton solution technique*.

5. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

6. Choose an option for **Default load variation with time**:

   - Choose **Instantaneous** if you want loads to be applied instantaneously at the start of the step and remain constant throughout the step.
   - Choose **Ramp linearly over step** if the load magnitude is to vary linearly over the step, from the value at the end of the previous step to the full magnitude of the load.

7. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

For more information, see *Extrapolation of the Solution*.

8. Toggle on **Stop when region *region name* is fully plastic** if "fully plastic" analysis is required with deformation theory plasticity. If you toggle on this option, enter the name of the region being monitored for fully plastic behavior.

   The step ends when the solutions at all constitutive calculation points in the element set are fully plastic (defined by the equivalent strain being 10 times the offset yield strain). However, the step can end before this point if either the maximum number of increments that you specified on the **Incrementation** tabbed page or the time period that you specified on the **Basic** tabbed page is exceeded.

9. If you selected **Fixed** time incrementation on the **Incrementation** tabbed page, you can toggle on **Accept solution after reaching maximum number of iterations**. This option directs Abaqus/Standard to accept the solution to an increment after the maximum number of iterations allowed has been completed, even if the equilibrium tolerances are not satisfied. Very small increments and a minimum of two iterations are usually necessary if you use this option.

   Warning:

   This approach is not recommended; you should use it only in special cases when you have a thorough understanding of how to interpret results obtained in this way.

10. Toggle on **Obtain long-term solution with time-domain material properties** to obtain the fully relaxed long-term elastic solution with time-domain viscoelasticity or the long-term elastic-plastic solution for two-layer viscoplasticity. This parameter is relevant only for time-domain viscoelastic and two-layer viscoplastic materials.
11. When you have finished configuring settings for the static, general step, click **OK** to close the **Edit Step** dialog box.

Configuring a static, Riks procedure

Geometrically nonlinear static problems sometimes involve buckling or collapse behavior, where the load-displacement response shows a negative stiffness, and the structure must release strain energy to remain in equilibrium. The modified Riks method allows you to find static equilibrium states during the unstable phase of the response.

You can use this method for cases where the load magnitudes are governed by a single scalar parameter. It is also useful for solving ill-conditioned problems such as limit load problems or almost unstable problems that exhibit softening. For more information, see *Unstable Collapse and Postbuckling Analysis*.

Create or edit a static, Riks procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Static, Riks`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as stopping criteria, the maximum number of increments, the arc increment length, and whether to account for geometric nonlinearity as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Select an **Nlgeom** option:
   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
4. Toggle on **Include adiabatic heating effects** if you are performing an adiabatic stress analysis. This option is relevant only for isotropic metal plasticity materials with a Mises yield surface. For more information, see *Adiabatic Analysis*.
5. Since the loading magnitude is part of the solution, you need a method to specify when the step is completed. Choose one or both of the following options:
   - Toggle on **Maximum load proportionality factor** to enter a maximum value for the load proportionality factor, λ_{end}. Abaqus/Standard uses this value to terminate the step when the load exceeds a certain magnitude. For more information, see *Proportional Loading*
   - Toggle on **Maximum displacement** to enter a maximum displacement value at a specific degree of freedom (**DOF**). You must also specify the **Node Region** that Abaqus/Standard will monitor for finishing displacement. If this maximum displacement is exceeded, Abaqus/Standard terminates the step.

   If you leave both of these finishing conditions unspecified, the analysis continues for the number of increments that you specify on the **Incrementation** tabbed page.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:

- Choose **Automatic** to allow Abaqus/Standard to choose the size of the arc length increments based on computational efficiency.
- Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an arc length increment that you specify as the constant increment size throughout the step. This method is not recommended for a Riks analysis since it prevents Abaqus/Standard from reducing the arc length when a severe nonlinearity is encountered.

For more information, see *Incrementation*.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Arc length increment**:
   a. In the **Initial** field, enter the initial increment in arc length along the static equilibrium path in scaled load-displacement space, Δl_{in}.
   b. In the **Minimum** field, enter the minimum arc length increment, Δl_{min}. If you enter zero, Abaqus assumes a default value of the smaller of the suggested initial arc length or 10^{-5} times the total arc length.
   c. In the **Maximum** field, enter the maximum arc length increment, Δl_{max}. If this value is not specified, no upper limit is imposed.
   d. In the **Estimated total arc length** field, enter the total arc length scale factor associated with this step, l_{period}. If this entry is zero or is unspecified, Abaqus/Standard assumes a default value of **1.0**.
5. If you selected **Fixed** in Step 2, enter a value for the constant arc length increment in the **Arc length increment** field.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
3. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:
   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

4. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic, and Abaqus/Standard should use a 1% linear extrapolation of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   (The **Parabolic** option is not relevant for Riks analyses.) For more information, see *Extrapolation of the Solution*.

5. Toggle on **Stop when region *region name* is fully plastic** if "fully plastic" analysis is required with deformation theory plasticity. If you toggle on this option, enter the name of the region being monitored for fully plastic behavior.

   The step ends when the solutions at all constitutive calculation points in the element set are fully plastic (defined by the equivalent strain being 10 times the offset yield strain). However, the step can end before this point if the maximum number of increments that you specified on the **Incrementation** tabbed page is exceeded.

6. If you selected **Fixed** time incrementation on the **Incrementation** tabbed page, you can toggle on **Accept solution after reaching maximum number of iterations**. This option directs Abaqus/Standard to accept the solution to an increment after the maximum number of iterations allowed has been completed, even if the equilibrium tolerances are not satisfied. Very small increments and a minimum of two iterations are usually necessary if you use this option.

   Warning:

   This approach is not recommended; you should use it only in special cases when you have a thorough understanding of how to interpret results obtained in this way.

7. Toggle on **Obtain long-term solution with time-domain material properties** to obtain the fully relaxed long-term elastic solution with time-domain viscoelasticity or the long-term elastic-plastic solution for two-layer viscoplasticity. This parameter is relevant only for time-domain viscoelastic and two-layer viscoplastic materials.

When you have finished configuring settings for the static, Riks step, click **OK** to close the **Edit Step** dialog box.

Configuring a dynamic, explicit procedure

An explicit, dynamic analysis is computationally efficient for the analysis of large models with relatively short dynamic response times and for the analysis of extremely discontinuous events or processes. This type of analysis allows for the definition of very general contact conditions and uses a consistent, large-deformation theory. For more information, see *Explicit Dynamic Analysis*.

Create or edit a dynamic, explicit procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Dynamic, Explicit`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, **Mass scaling**, and **Other** tabbed pages, configure settings such as the time period for the step, the maximum time increment, the increment size, mass scaling definitions, and bulk viscosity parameters as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Time period** field, enter the time period of the step.
4. Select an **Nlgeom** option:
   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Explicit should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
5. Toggle on **Include adiabatic heating effects** if you are performing an adiabatic stress analysis. This option is relevant only for metal plasticity. For more information, see *Adiabatic Analysis*.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:
   - Choose **Automatic** to allow Abaqus/Explicit to determine the time incrementation automatically. For more information, see *Automatic Time Incrementation*.
   - Choose **Fixed** to use a fixed time incrementation scheme. The fixed time increment size is determined either by the initial element stability estimate for the step or by a user-specified time increment. For more information, see *Fixed Time Incrementation*.
3. If you selected **Automatic** time incrementation, perform the following steps:
   a. Choose a **Stable increment estimator** option:
      - Choose **Global** to allow the global estimator to determine the stability limit as the step proceeds. The adaptive, global estimation algorithm determines the maximum frequency of the entire model using the current dilatational wave speed. This algorithm continuously updates the estimate for the maximum frequency. The global estimator will usually allow time increments that exceed the element-by-element values.

- Choose **Element-by-element** to allow Abaqus/Explicit to determine an element-by-element estimate using the current dilatational wave speed in each element.

  The element-by-element estimate is conservative; it will give a smaller stable time increment than the true stability limit that is based upon the maximum frequency of the entire model. In general, constraints such as boundary conditions and kinematic contact have the effect of compressing the eigenvalue spectrum, and the element-by-element estimates do not take this into account.

b. By default, the "improved" method to estimate the element stable time increment for three-dimensional continuum elements and elements with plane stress formulations is used. This method usually results in a larger element stable time increment than a more traditional method. Toggle off **Improved Dt Method** to deactivate the "improved" method.

c. Choose a **Max. time increment** option:

- Choose **Unlimited** if you do not want to impose an upper limit to time incrementation.
- Choose **Value** to enter a value for the maximum time increment allowed. Enter the value in the field provided.

For more information, see *Automatic Time Incrementation*.

4. If you selected **Fixed** time incrementation, choose an option for determining increment size:

- Choose **User-defined time increment** to specify a time increment size directly. Enter that time increment size in the field provided.
- Choose **Use element-by-element time increment estimator** to use time increments the size of the initial element-by-element stability limit throughout the step. The dilatational wave speed in each element at the beginning of the step is used to compute the fixed time increment size.

For more information, see *Fixed Time Incrementation*.

5. If desired, enter a **Time scaling factor** to adjust the stable time increment computed by Abaqus/Explicit. (This option is unavailable if you have specified a **User-defined time increment** for the **Fixed** time incrementation scheme.) For more information, see *Scaling the Time Increment*.

*Configure settings on the **Mass scaling** tabbed page*

1. In the **Edit Step** dialog box, display the **Mass scaling** tabbed page. For background information on mass scaling, see *Mass Scaling*.

(For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose one of the following options for specifying mass scaling:

- Choose **Use scaled mass and “throughout step” definitions from the previous step** if you want mass scaling definitions from the previous step to propagate through the current step. If you choose this option, you can skip the remaining steps in this procedure.
- Choose **Use scaling definitions below** to create one or more new mass scaling definitions for this step. If you choose this option, complete the remaining steps in this procedure.

3. At the bottom of the **Data** table, click **Create**.

An **Edit mass scaling** dialog box appears.

4. Specify which type of mass scaling definition you want to create:

- Choose **Semi-automatic mass scaling** to define mass scaling for any type of analysis except bulk metal rolling.

- Choose **Automatic mass scaling** to define mass scaling for a bulk metal rolling analysis. For more information, see *Automatic Mass Scaling for Analysis of Bulk Metal Rolling*.
- Choose **Reinitialize mass** to reinitialize masses of elements to their original values. This option allows you to prevent the scaled mass from a previous step from being used in the current step. For more information, see *Reverting the Mass Matrix to the Original State*.
- Choose **Disable mass scaling thoughout step** to disable in this step all variable mass scaling definitions from previous steps. For more information, see *Continuous Mass Matrix with No Further Scaling*.

5. If you selected **Semi-automatic mass scaling**, **Automatic mass scaling**, or **Reinitialize mass**, indicate the region to which you want the mass scaling definition applied:
   - Choose **Whole model** to apply the mass scaling definition to all elements in the model.
   - Choose **Set** to apply the mass scaling definition to a particular set of elements. Enter the set name in the field provided.

6. If you selected **Semi-automatic mass scaling**, indicate when, during the step, you want Abaqus/Explicit to scale the element masses:
   - Choose **At beginning of step** to perform fixed mass scaling only at the beginning of the step. For more information, see *Fixed Mass Scaling*.
   - Choose **Throughout step** to scale the mass of elements periodically during the step. For more information, see *Variable Mass Scaling*.

7. If you selected **Semi-automatic mass scaling**, indicate how you want Abaqus/Explicit to scale the element masses:
   - Toggle on **Scale by factor** to scale the elements once at the beginning of the step by the value you enter in the field provided. For more information, see *Defining a Scale Factor Directly*.
   - Toggle on **Scale to target time increment of *n*** to enter a desired element stable time increment in the field provided. Click the arrow to the right of the **Scale element mass** field, and select how you want Abaqus/Explicit to apply that target time increment:
     - Select **Uniformly to satisfy target** to scale the masses of the elements equally so that the smallest element stable time increment of the scaled elements equals the target value.
     - Select **If below minimum target** to scale the masses of only the elements whose element stable time increments are less than the target value.
     - Select **Nonuniformly to equal target** to scale the masses of all elements so that they all have the same element stable time increment equal to the target value.

     For more information, see *Defining a Desired Element-by-Element Stable Time Increment*.

   If you toggle on both **Scale by factor** and **Scale to target time increment**, Abaqus/Explicit first scales the masses by the factor value that you enter and then possibly scales them again, depending on the value you enter for target time increment and the option you select for applying that target.

8. If you selected **Automatic mass scaling**, enter the following values:
   - In the **Feed rate** field, enter the estimated average velocity of the workpiece in the rolling direction at steady-state conditions.
   - In the **Extruded element length** field, enter the average element length in the rolling direction.
   - In the **Nodes in cross-section** field, enter the number of nodes in the cross-section of the workpiece. Increasing this value decreases the amount of mass scaling.

For more information, see *Automatic Mass Scaling for Analysis of Bulk Metal Rolling*.

9. If you selected **Semi-automatic mass scaling** throughout the step or **Automatic mass scaling**, specify when, during the step, you want Abaqus/Explicit to perform mass scaling calculations:

   - Choose **Every *n* increments** to specify the frequency, in increments, at which Abaqus/Explicit is to perform mass scaling calculations. Enter the desired frequency in the field provided.

     For example, if you enter a value of 5, Abaqus/Explicit scales the mass at the beginning of the step and at increments 5, 10, 15, etc.

   - Choose **At *n* equal intervals** to specify the number of intervals during the step at which Abaqus/Explicit is to perform mass scaling calculations. Enter the desired value in the field provided.

     For example, if you enter a value of 2, Abaqus/Explicit scales the mass at the beginning of the step, the increment immediately following the half-way point in the step, and the final increment in the step.

10. Click **OK** to close the **Edit mass scaling** dialog box and return to the **Mass scaling** tabbed page of the **Edit Step** dialog box.

    The mass scaling definition that you have just created appears in the **Data** table.

11. If desired, repeat Steps 3 to 10 to create additional mass scaling definitions.
12. Once you have created one or more mass scaling definitions, you can edit or delete them if desired. Select a particular mass scaling definition in the **Data** table, and click **Edit** or **Delete** at the bottom of the **Data** table.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Enter a value for the **Linear bulk viscosity parameter**. Linear bulk viscosity is included by default in Abaqus/Explicit.
3. Enter a value for the **Quadratic bulk viscosity parameter**. This form of bulk viscosity pressure is found only in solid continuum element and is applied only if the volumetric strain rate is compressive.

   For more information, see *Bulk Viscosity*.

When you have finished configuring settings for the dynamic, explicit step, click **OK** to close the **Edit Step** dialog box.

Configuring a heat transfer procedure

You can perform an uncoupled heat transfer analysis to model solid body heat conduction with general, temperature-dependent conductivity, internal energy (including latent heat effects), and general convection and radiation boundary conditions, including cavity radiation. For more information, see *Uncoupled Heat Transfer Analysis*.

Create or edit a heat transfer procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Heat transfer`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, the maximum allowable temperature change per increment, and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Response** option:
   - Choose **Steady-state** to omit the internal energy term (the specific heat term) in the governing heat transfer equation. For more information, see *Steady-State Analysis*.
   - Choose **Transient** to perform time integration with the backward Euler method in the pure conduction elements. This method is unconditionally stable for linear problems. For more information, see *Transient Analysis*.

   Note:

   After you have selected a **Response** option, a message appears informing you that Abaqus/Standard has selected the **Default load variation with time** option (located on the **Other** tabbed page) that corresponds to your **Response** selection. Click **Dismiss** to close the message dialog box.

4. In the **Time period** field, enter the time period of the step.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:
   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.
3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.

b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.

c. In the **Maximum** field, enter the maximum time increment allowed.

5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. If you selected **Transient** analysis on the **Basic** tabbed page, do the following:

   a. Toggle on **End step when temperature change is less than *n*** if you want the analysis to end when the temperature at every temperature degree of freedom changes at a rate that is less than a rate that you specify. If you toggle on this option, enter the desired temperature change rate in the field provided.

   b. If you selected **Automatic** in Step 2, enter a value for the **Max. allowable temperature change per increment**. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node (except nodes whose temperature degree of freedom is constrained via boundary conditions, MPCs, etc.) during any increment of the step.
7. If you selected **Automatic** in Step 2 and you are performing a cavity radiation analysis, enter a value for **Max. allowable emissivity change per increment** or accept the default of 0.1. If this value is exceeded, Abaqus/Standard cuts back the increment until the maximum change in emissivity is less than the specified value. See *Cavity Radiation in Abaqus/Standard*, for more information.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose an **Equation Solver Method** option:

   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.
3. Choose a **Matrix storage** option:

   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
4. Choose a **Solution technique** option:

   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

     If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

     For more information, see *Quasi-Newton solution technique*.

5. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

6. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.
7. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the heat transfer step, click **OK** to close the **Edit Step** dialog box.

Configuring a dynamic, implicit procedure

General linear or nonlinear dynamic analysis in Abaqus/Standard uses implicit time integration to calculate the transient dynamic response of a system. See *Implicit Dynamic Analysis Using Direct Integration*, or *Implicit dynamic analysis*, for details on implicit dynamic analysis.

Create or edit a dynamic, implicit procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Dynamic, Implicit`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, increment size, and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Time period** field, enter the time period of the step.
4. Select an **Nlgeom** option:

   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
5. Select an **Application** option. The application setting adjusts various numerical settings (such as damping and time incrementation) to most efficiently and accurately capture the intended behavior of your analysis.

   - **Transient fidelity** applications—such as an analysis of satellite systems—use small time increments to accurately resolve the vibrational response of the structure, and numerical energy dissipation is kept at a minimum.
   - **Moderate dissipation** applications—including various insertion, impact, and forming analyses—use some energy dissipation (via plasticity, viscous damping, or numerical effects) to reduce solution noise and improve convergence behavior without significantly degrading solution accuracy.
   - **Quasi-static** applications introduce inertia effects primarily to regularize unstable behavior in analyses whose main focus is a final static response. Large time increments are taken when possible to minimize computational cost, and considerable numerical dissipation may be used to obtain convergence during certain stages of the loading history.
   - The **Analysis product default** depends on the presence of contact in the model: analyses involving contact are treated as moderate dissipation applications; analyses without contact are treated as transient fidelity applications.
6. Toggle on **Include adiabatic heating effects** if you are performing an adiabatic stress analysis. This option is relevant only for isotropic metal plasticity materials with a Mises yield surface. For more information, see *Adiabatic Analysis*.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

(For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Type** option:
   - Choose **Automatic** to allow Abaqus/Standard to choose the size of the increments based on computational efficiency.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

     Warning:

     Fixed incrementation is not generally recommended; it should be used only in special cases when you have a thorough understanding of how to interpret results obtained in this way. Impact events are particularly difficult to solve using fixed time increments.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, do the following:
   a. Enter values for **Increment size**:
      - In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
      - In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.

   b. Specify the **Maximum increment size**:
      - Choose **Specify** to enter the maximum increment size directly.
      - Choose **Analysis application default** to set the maximum increment size automatically based on the application setting:
        - For transient fidelity applications, the default maximum increment is the time period of the step divided by 100.
        - For moderate dissipation applications, the default maximum increment is the time period of the step divided by 10.
        - For quasi-static applications, the default maximum increment is the time period of the step.

   c. The half-increment residual tolerance represents the equilibrium residual error (out-of-balance forces) halfway through a time increment. If the half-increment residual is small, it indicates that the accuracy of the solution is high and that the time step can be increased safely; conversely, if the half-increment residual is large, the time step used in the solution should be reduced. For more information, see *Numerical Details*.

      You must specify an appropriate **Half-increment Residual**:
      - Toggle on **Suppress calculation** to reduce the solution cost by skipping half-increment residual tolerance checks.
      - Choose **Analysis product default** to set a half-increment residual tolerance automatically based on the application setting:
        - For transient fidelity applications involving contact, the default half-increment residual tolerance is 10,000 times the time average force and moment values.

- For transient fidelity applications without contact, the default half-increment residual tolerance is 1000 times the time average force and moment values.
- For moderate dissipation and quasi-static applications, the half-increment residual tolerance checks are suppressed.

- Choose **Specify scale factor** to enter the half-increment residual tolerance as a scale factor applied to the time average force and moment values.
- Choose **Specify value** to enter the half-increment residual tolerance value directly.

5. If you selected **Fixed** in Step 2, do the following:
   a. Enter a value for the constant time increment in the **Increment size** field.
   b. If desired, toggle on **Suppress calculation** to skip half-increment residual tolerance checks and reduce the solution cost.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
3. Choose a **Solution technique**:
   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

     If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

     For more information, see *Quasi-Newton solution technique*.
4. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:
   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.

- Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

5. Choose an option for **Default load variation with time**:
   - Choose **Instantaneous** if you want loads to be applied instantaneously at the start of the step and remain constant throughout the step.
   - Choose **Ramp linearly over step** if the load magnitude is to vary linearly over the step, from the value at the end of the previous step to the full magnitude of the load.
6. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:
   - Select **None** to suppress any extrapolation.
   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic displacement-based extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **Velocity parabolic** to indicate that the process should use a quadratic velocity-based extrapolation, in time, of the previous incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **Analysis product default** to select the extrapolation method automatically based on the application setting:
     - For transient fidelity applications, Abaqus/Standard uses the velocity-based parabolic extrapolation method.
     - For moderate dissipation and quasi-static applications, Abaqus/Standard uses the linear extrapolation method.

   For more information, see *Extrapolation of the Solution*.
7. For transient fidelity applications, indicate **Alpha**, the numerical (artificial) damping control parameter in the implicit operator:
   - Choose **Analysis product default** to set $\alpha = -0.05$ for slight numerical damping.
   - Choose **Specify** to enter a nondefault value for α. Allowable values are zero (no damping) to -0.5 ($\alpha = -0.333$ provides maximum damping).

   For moderate dissipation applications, α cannot be modified from the default value of -0.41421. The α parameter is not used in quasi-static applications.
8. Indicate how Abaqus/Standard should handle **Initial acceleration calculations at beginning of step**:
   - Choose **Allow** to calculate the actual accelerations in a model at the beginning of the dynamic step.
   - Choose **Bypass** to set the initial accelerations based on the following criteria:
     - If the current step is the first dynamic step, Abaqus/Standard assumes that the initial accelerations for the current step are zero.
     - If the immediately preceding step was also a dynamic step, Abaqus/Standard uses the accelerations from the end of the previous step to continue the new step.

This approach is appropriate only if the loading does not change suddenly at the start of the new step. For more information, see *Controlling Calculation of Accelerations at the Beginning of a Dynamic Step*.

- Choose **Analysis product default** to determine the initial accelerations based on the application setting used for the step (this option is available only if the **Application** option on the **Basic** tabbed page is also set to **Analysis product default**):
  - For transient fidelity applications, the actual initial accelerations are calculated.
  - For moderate dissipation applications, the actual initial accelerations are set based on the criteria described above for the **Bypass** option.

9. If you selected **Fixed** time incrementation on the **Incrementation** tabbed page, you can toggle on **Accept solution after reaching maximum number of iterations**. This option directs Abaqus/Standard to accept the solution to an increment after the maximum number of iterations allowed has been completed, even if the equilibrium tolerances are not satisfied. Very small increments and a minimum of two iterations are usually necessary if you use this option.

   Warning:

   This approach is not recommended; you should use it only in special cases when you have a thorough understanding of how to interpret results obtained in this way.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a fully coupled, simultaneous heat transfer and stress procedure

You must configure a fully coupled temperature-displacement analysis when the stress analysis is dependent on the temperature distribution and the temperature distribution depends on the stress solution. For example, metalworking problems may include significant heating due to inelastic deformation of the material which, in turn, changes the material properties. For such cases the thermal and mechanical solutions must be obtained simultaneously rather than sequentially. For more information, see *Fully Coupled Thermal-Stress Analysis*.

Create or edit a coupled temperature-displacement procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Coupled temp-displacement`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, increment size, and solution technique preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Indicate whether you want **Steady-state** or **Transient** response. See the following sections for more information:
   - *Steady-State Analysis*
   - *Transient Analysis*

   Note:

   After you have selected a **Response** option, a message appears informing you that Abaqus/Standard has selected the **Default load variation with time** option (located on the **Other** tabbed page) that corresponds to your **Response** selection. Click **Dismiss** to close the message dialog box.
4. In the **Time period** field, enter the time period of the step.
5. Choose an **Nlgeom** option:
   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
6. Select an automatic stabilization method if you expect the problem to have local instabilities such as surface wrinkling, material instability, or local buckling. Abaqus/Standard can stabilize this class of problems by applying damping throughout the model. For more information, see *Unstable Problems*, and *Automatic Stabilization of Static Problems with a Constant Damping Factor*.

   Click the arrow to the right of **Automatic stabilization**, and select a method for defining the damping factor:
   - Select **Specify dissipated energy fraction** to allow Abaqus/Standard to calculate the damping factor from a dissipated energy fraction that you provide. Enter a value for the dissipated energy fraction in the adjacent field (the default is 2.0×10^{-4}). For more information, see *Calculating the Damping Factor Based on the Dissipated Energy Fraction*.

- Select **Specify damping factor** to enter the damping factor directly. Enter a value for the damping factor in the adjacent field. For more information, see *Directly Specifying the Damping Factor*.
- Select **Use damping factors from previous general step** to use the damping factors at the end of the previous step as the initial factors in the current step's variable damping scheme. These factors override any initial damping factors that are calculated or specified directly in the current step. If there are no damping factors associated with the previous general step (for example, if the previous step does not use any stabilization or the current step is the first step of the analysis), Abaqus uses adaptive stabilization to determine the required damping factors.

7. When using automatic stabilization, Abaqus can use the same damping factor over the course of a step, or it can vary the damping factor spatially and temporally during a step based on the convergence history and the ratio of the energy dissipated by damping to the total strain energy. For more information, see *Adaptive Automatic Stabilization Scheme*. If you selected **Specify dissipated energy fraction**, adaptive stabilization is optional and turned on by default. If you selected **Specify damping factor**, adaptive stabilization is optional and turned off by default. If you selected **Use damping factors from previous general step**, adaptive stabilization is required.

   To use adaptive stabilization, toggle on **Use adaptive stabilization with max. ratio of stabilization to strain energy** (if necessary), and enter a value in the adjacent field for the allowable accuracy tolerance for the ratio of energy dissipated by damping to total strain energy in each increment. The default value of 0.05 should be suitable in most cases.

8. If desired, toggle on **Include creep/swelling/viscoelastic behavior**. If you leave this option toggled off, you indicate that there is no creep or viscoelastic response occurring during this step even if creep or viscoelastic material properties have been defined.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.
5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. If you selected **Automatic** in Step 2 and if you selected **Transient** response on the **Basic** tabbed page, do the following:
   a. Enter a value for the **Max. allowable temperature change per increment**. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node during any increment of the step.

b. If you toggled on **Include creep/swelling/viscoelastic behavior** on the **Basic** tabbed page, toggle on **Creep/swelling/viscoelastic strain error tolerance** to enter the maximum difference in the creep strain increment calculated from the creep strain rates at the beginning and at the end of the increment. This value controls the accuracy of the creep integration. For more information, see *Automatic Incrementation Controlled by the Creep Response*.

7. If you toggled on **Include creep/swelling/viscoelastic behavior** on the **Basic** tabbed page, choose a **Creep/swelling/viscoelastic integration** option:

- Choose **Explicit/Implicit** if you want to allow Abaqus/Standard to invoke the implicit integration scheme. For most coupled thermal-stress analyses, the unconditional stability of the backward difference operator (implicit method) is desirable.
- Choose **Explicit** if you want to restrict Abaqus/Standard to using explicit integration. Explicit integration can be less expensive computationally and simplifies implementation of user-defined creep laws in user subroutine *CREEP*.

For more information, see *Automatic Incrementation Controlled by the Creep Response*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

(For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Matrix storage** option:

- Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
- Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme. (This is the only matrix storage option available if you choose the **Full Newton** solution technique.)
- Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

3. Choose a **Solution technique**:

- Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
- Choose **Separated** to specify that linearized equations for the individual fields in the fully coupled procedure are to be decoupled and solved separately for each field. This option provides a less costly solution for an analysis that is fully coupled in the sense that the mechanical and thermal solutions evolve simultaneously, but with a weak coupling between the two solutions. For more information, see *Approximate Implementation*.

4. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

- Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
- Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.

- Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

5. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.
6. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a fully coupled, simultaneous heat transfer and electrical procedure

Joule heating arises when the energy dissipated by an electrical current flowing through a conductor is converted into thermal energy. Abaqus/Standard provides a fully coupled thermal-electrical procedure for analyzing this type of problem; the coupled thermal-electrical equations are solved simultaneously for both temperature and electrical potential at the nodes. For more information, see *Coupled Thermal-Electrical Analysis*.

Create or edit a coupled thermal-electrical procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Coupled thermal-electric`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, increment size, and solution technique preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Response** option:

   - Choose **Steady-state** to omit the internal energy term (the specific heat term) in the governing heat transfer equation. Only direct current is considered in the electrical problem, and it is assumed that the system has negligible capacitance. (Electrical transient effects are so rapid that they can be neglected.) For more information, see *Steady-State Analysis*.
   - Choose **Transient** to perform time integration with the same backward Euler method used in uncoupled heat transfer analyses. This method is unconditionally stable for linear problems. For more information, see *Transient Analysis*.

   Note:

   After you have selected a **Response** option, a message appears informing you that Abaqus/Standard has selected the **Default load variation with time** option (located on the **Other** tabbed page) that corresponds to your **Response** selection. Click **Dismiss** to close the message dialog box.

4. In the **Time period** field, enter the time period of the step.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:

   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.

5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. If you selected **Transient** analysis on the **Basic** tabbed page, do the following:

   - Toggle on **End step when temperature change is less than *n*** if you want the analysis to end when the temperature at every temperature degree of freedom changes at a rate that is less than a rate that you specify. If you toggle on this option, enter the desired temperature change rate in the field provided.
   - If you selected **Automatic** in Step 2, enter a value for the **Max. allowable temperature change per increment**. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node (except nodes with boundary conditions) during any increment of the step.

7. If you selected **Automatic** in Step 2 and you are performing a cavity radiation analysis, enter a value for **Max. allowable emissivity change per increment**, or accept the default of 0.1. If this value is exceeded, Abaqus/Standard cuts back the increment until the maximum change in emissivity is less than the specified value. See *Cavity Radiation in Abaqus/Standard*, for more information.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:

   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme. (This is the only matrix storage option available if you choose the **Full Newton** solution technique.)
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
3. Choose a **Solution technique**:

   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Separated** to specify that linearized equations for the individual fields in the fully coupled procedure are to be decoupled and solved separately for each field. This option provides a less costly solution for an analysis that is fully coupled in the sense that the electrical and thermal solutions evolve simultaneously, but with a weak coupling between the two solutions. For more information, see *Approximate Implementation*.

4. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.

- Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
- Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

5. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.
6. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a fully coupled, simultaneous heat transfer, electrical, and structural procedure

A fully coupled thermal-electrical-structural analysis is the union of a coupled thermal-displacement analysis and a coupled thermal-electrical analysis. Coupling between the temperature and electrical degrees of freedom arises from temperature-dependent electrical conductivity and internal heat generation (Joule heating), which is a function of the electrical current density. Coupling between the temperature and displacement degrees of freedom arises from temperature-dependent material properties, thermal expansion, and internal heat generation, which is a function of inelastic deformation of the material. Coupling between the electrical and displacement degrees of freedom arises in problems where electricity flows between contact surfaces.

For more information, see *Fully Coupled Thermal-Electrical-Structural Analysis*.

Create or edit a coupled thermal-electrical-structural procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Coupled thermal-electric-structural`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period for the step, type of incrementation, and solution technique preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Response** option:
   - Choose **Steady-state** to omit the internal energy term (the specific heat term) in the governing heat transfer equation. A static displacement solution is assumed. Only direct current is considered in the electrical problem, and it is assumed that the system has negligible capacitance. (Electrical transient effects are so rapid that they can be neglected.) For more information, see *Steady-State Analysis*.
   - Choose **Transient** to perform a transient analysis. As with the steady-state response, electrical transient effects are neglected and a static displacement solution is assumed. You can control the time incrementation in a transient analysis directly, or Abaqus/Standard can control it automatically. Automatic time incrementation is generally preferred. For more information, see *Transient Analysis*.
4. In the **Time period** field, enter the time period of the step.
5. Choose an **Nlgeom** option:
   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.
6. Select an automatic stabilization method if you expect the problem to have local instabilities such as surface wrinkling, material instability, or local buckling. Abaqus/Standard can stabilize this class of problems by applying damping throughout the model. For more information, see *Unstable Problems*, and *Automatic Stabilization of Static Problems with a Constant Damping Factor*.

   Click the arrow to the right of **Automatic stabilization**, and select a method for defining the damping factor:

- Select **Specify dissipated energy fraction** to allow Abaqus/Standard to calculate the damping factor from a dissipated energy fraction that you provide. Enter a value for the dissipated energy fraction in the adjacent field (the default is 2.0×10^{-4}). For more information, see *Calculating the Damping Factor Based on the Dissipated Energy Fraction*.
- Select **Specify damping factor** to enter the damping factor directly. Enter a value for the damping factor in the adjacent field. For more information, see *Directly Specifying the Damping Factor*.
- Select **Use damping factors from previous general step** to use the damping factors at the end of the previous step as the initial factors in the current step's variable damping scheme. These factors override any initial damping factors that are calculated or specified directly in the current step. If there are no damping factors associated with the previous general step (for example, if the previous step does not use any stabilization or the current step is the first step of the analysis), Abaqus uses adaptive stabilization to determine the required damping factors.

7. When using automatic stabilization, Abaqus can use the same damping factor over the course of a step, or it can vary the damping factor spatially and temporally during a step based on the convergence history and the ratio of the energy dissipated by damping to the total strain energy. For more information, see *Adaptive Automatic Stabilization Scheme*. If you selected **Specify dissipated energy fraction**, adaptive stabilization is optional and turned on by default. If you selected **Specify damping factor**, adaptive stabilization is optional and turned off by default. If you selected **Use damping factors from previous general step**, adaptive stabilization is required.

   To use adaptive stabilization, toggle on **Use adaptive stabilization with max. ratio of stabilization to strain energy** (if necessary), and enter a value in the adjacent field for the allowable accuracy tolerance for the ratio of energy dissipated by damping to total strain energy in each increment. The default value of 0.05 should be suitable in most cases.

8. If desired, toggle on **Include creep/swelling/viscoelastic behavior**. If you leave this option toggled off, you indicate that there is no creep or viscoelastic response occurring during this step even if creep or viscoelastic material properties have been defined.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.
5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.

6. If you selected **Automatic** in Step 2 and you selected **Transient** response on the **Basic** tabbed page, do the following:
   a. Enter a value for the **Max. allowable temperature change per increment**. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node (except nodes with boundary conditions) during any increment of the step.
   b. If you toggled on **Include creep/swelling/viscoelastic behavior** on the **Basic** tabbed page, toggle on **Creep/swelling/viscoelastic strain error tolerance** to enter the maximum difference in the creep strain increment calculated from the creep strain rates at the beginning and at the end of the increment. This value controls the accuracy of the creep integration. For more information, see *Automatic Incrementation Controlled by the Creep Response*.
7. If you toggled on **Include creep/swelling/viscoelastic behavior** on the **Basic** tabbed page, choose a **Creep/swelling/viscoelastic integration** option:
   - Choose **Explicit/Implicit** if you want to allow Abaqus/Standard to invoke the implicit integration scheme. For most coupled thermal-stress analyses, the unconditional stability of the backward difference operator (implicit method) is desirable.
   - Choose **Explicit** if you want to restrict Abaqus/Standard to using explicit integration. Explicit integration can be less expensive computationally and simplifies implementation of user-defined creep laws in user subroutine *CREEP*.

   For more information, see *Automatic Incrementation Controlled by the Creep Response*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
3. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:
   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

4. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.
5. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a direct cyclic procedure

A direct cyclic procedure is a quasi-static analysis that uses a combination of Fourier series and time integration of the nonlinear material behavior to obtain the stabilized cyclic response of the structure iteratively. To avoid the considerable numerical expense associated with a transient analysis, a direct cyclic procedure can be used to calculate the cyclic response of a structure directly. The basis of this method is to construct a displacement function $\overline{u}(t)$ that describes the response of the structure at all times t during a load cycle with period T. For more information, see *Direct Cyclic Analysis*.

Abaqus/Standard assumes geometrically linear behavior for a direct cyclic procedure. For more information, see *Linear and nonlinear procedures*.

Create or edit a direct cyclic procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Direct cyclic`) or *Editing a step*.
2. On the **Basic**, **Incrementation**, **Fatigue**, and **Other** tabbed pages, configure settings such as the cycle time period, maximum number of increments, increment size, low-cycle fatigue options, and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Cycle time period** field, enter the time of a single loading cycle.
4. Toggle on **Use displacement Fourier coefficients from previous direct cyclic step** to indicate that the current step is a continuation of the previous direct cyclic step. See *Direct Cyclic Analysis* for more details.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step* or *Editing a step*.)
2. Choose a **Type** option:
   - Choose **Automatic** to allow Abaqus/Standard to choose the size of the time increments based on computational efficiency.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.
3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in a single loading cycle. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step. See *Controlling the Incrementation during the Cyclic Time Period* for more details.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.

5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. In the **Maximum number of iterations** field, enter an upper limit for the number of cyclic iterations. See *Controlling the Iterations in the Modified Newton Method* for more details.
7. In the **Number of Fourier terms** fields, enter values for the **Initial** and **Maximum** number of Fourier terms and the **Increment** in the number of terms. The number of Fourier terms required to obtain an accurate solution depends on the variation of the load as well as the variation of the structural response over the period. More Fourier terms usually provide a more accurate solution but at the expense of additional data storage and computational time. Each of these values must be greater than 0 and less than 100. For more information, see *Controlling the Fourier Representations*.
8. If you selected **Automatic** in Step 2, choose one or both of the following options:

   - Toggle on **Max. allowable temperature change per increment** to enter the maximum temperature change to be allowed in an increment. Abaqus/Standard will restrict the time increment to ensure that this value is not exceeded at any node during any increment of the step.
   - Toggle on **Creep/swelling/viscoelastic strain error tolerance** to enter the maximum difference in the creep strain increment calculated from the creep strain rates based on conditions at the beginning and end of the increment, thus controlling the time integration accuracy of the creep integration.

   For more details about these options, see *Automatic Incrementation*.
9. Toggle on **Evaluate structure response at time points** to define specific times at which the response should be evaluated. Click the arrow to the right of this field, and select a set of time points from the list that appears. Otherwise, click  to define a new set of time points. See *Defining time points* and *Defining the Time Points at Which the Response Must Be Evaluated* for more details.

*Configure settings on the **Fatigue** tabbed page*

1. In the **Edit Step** dialog box, display the **Fatigue** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step* or *Editing a step*.)
2. Toggle on **Include low-cycle fatigue analysis** to use the direct cyclic approach to obtain the stabilized response of a structure subjected to periodic loading. Multiple cycles can be included in a single direct cyclic analysis. The analysis models progressive damage and failure on constitutive points in the bulk materials based on a continuum damage approach. It can also be used to model delamination/debonding growth at the interfaces in laminated composites. For more details, see *Low-Cycle Fatigue Analysis Using the Direct Cyclic Approach*.
3. In the **Cycle increment size** fields, enter values for the **Minimum** and **Maximum** increment in the number of cycles over which the damage is extrapolated forward. Each value must be greater than 0. For more details, see *Damage Extrapolation Technique in the Bulk Material*.
4. In the **Maximum number of cycles** field, choose one of the following options to specify the total number of cycles allowed in the step:

   - Choose **Default** to use a value that is equal to one plus half of the maximum increment in number of cycles over which the damage is extrapolated.
   - Choose **Value**, and enter a number.

   See *Low-Cycle Fatigue Analysis in Abaqus/Standard* for more details.
5. In the **Damage extrapolation tolerance** field, enter a value or accept the default of 1.0. The maximum extrapolated damage increment will be limited by this value. See *Controlling the Accuracy of Damage*

Extrapolation in the Bulk Material When Using the Continuum Damage Mechanics Approach for more details.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step* or *Editing a step*.)

2. Choose a **Matrix storage** option for the equation solver:

   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

3. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

4. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the direct cyclic step, click **OK** to close the **Edit Step** dialog box.

Configuring a dynamic fully coupled thermal-stress procedure using explicit integration

You must configure a fully coupled temperature-displacement analysis when the stress analysis is dependent on the temperature distribution and the temperature distribution depends on the stress solution. For such cases the thermal and mechanical solutions must be obtained simultaneously rather than sequentially. In Abaqus/Explicit a fully coupled thermal-stress analysis includes inertia effects and models transient thermal response. For more information, see *Fully Coupled Thermal-Stress Analysis in Abaqus/Explicit*.

Create or edit a coupled temperature-displacement procedure using explicit integration

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Dynamic, Temp-disp, Explicit`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, **Mass scaling**, and **Other** tabbed pages, configure settings such as the time period for the step, the increment size, mass scaling definitions, and bulk viscosity parameters as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Time period** field, enter the time period of the step.
4. Select an **Nlgeom** option:
   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Explicit should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:
   - Choose **Automatic** to allow Abaqus/Explicit to determine the time incrementation automatically. For more information, see *Automatic Time Incrementation*.
   - Choose **Fixed** to use a fixed time incrementation scheme. The fixed time increment size is determined either by the initial element stability estimate for the step or by a user-specified time increment. For more information, see *Fixed Time Incrementation*.
3. If you selected **Automatic** time incrementation, perform the following steps:
   a. Choose a **Stable increment estimator** option:
      - Choose **Global** to allow the global estimator to determine the stability limit as the step proceeds. The adaptive, global estimation algorithm determines the maximum frequency of the entire model using the current dilatational wave speed. This algorithm continuously updates the estimate for the maximum frequency. The global estimator will usually allow time increments that exceed the element-by-element values.

- Choose **Element-by-element** to allow Abaqus/Explicit to determine an element-by-element estimate using the current dilatational wave speed in each element.

  The element-by-element estimate is conservative; it will give a smaller stable time increment than the true stability limit that is based upon the maximum frequency of the entire model. In general, constraints such as boundary conditions and kinematic contact have the effect of compressing the eigenvalue spectrum, and the element-by-element estimates do not take this into account.

b. By default, the "improved" method to estimate the element stable time increment for three-dimensional continuum elements and elements with plane stress formulations is used. This method usually results in a larger element stable time increment than a more traditional method. Toggle off **Improved Dt Method** to deactivate the "improved" method.

c. Choose a **Max. time increment** option:

- Choose **Unlimited** if you do not want to impose an upper limit to time incrementation.
- Choose **Value** to enter a value for the maximum time increment allowed. Enter the value in the field provided.

For more information, see *Automatic Time Incrementation*.

4. If you selected **Fixed** time incrementation, choose an option for determining increment size:

- Choose **User-defined time increment** to specify a time increment size directly. Enter that time increment size in the field provided.
- Choose **Use element-by-element time increment estimator** to use time increments the size of the initial element-by-element stability limit throughout the step. The dilatational wave speed in each element at the beginning of the step is used to compute the fixed time increment size.

For more information, see *Fixed Time Incrementation*.

5. If desired, enter a **Time scaling factor** to adjust the stable time increment computed by Abaqus/Explicit. (This option is unavailable if you have specified a **User-defined time increment** for the **Fixed** time incrementation scheme.) For more information, see *Scaling the Time Increment*.

*Configure settings on the **Mass scaling** tabbed page*

1. In the **Edit Step** dialog box, display the **Mass scaling** tabbed page. For background information on mass scaling, see *Mass Scaling*.

(For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose one of the following options for specifying mass scaling:

- Choose **Use scaled mass and "throughout step" definitions from the previous step** if you want mass scaling definitions from the previous step to propagate through the current step. If you choose this option, you can skip the remaining steps in this procedure.
- Choose **Use scaling definitions below** to create one or more new mass scaling definitions for this step. If you choose this option, complete the remaining steps in this procedure.

3. At the bottom of the **Data** table, click **Create**.

An **Edit mass scaling** dialog box appears.

4. Specify which type of mass scaling definition you want to create:

- Choose **Semi-automatic mass scaling** to define mass scaling for any type of analysis except bulk metal rolling.

- Choose **Automatic mass scaling** to define mass scaling for a bulk metal rolling analysis. For more information, see *Automatic Mass Scaling for Analysis of Bulk Metal Rolling*.
- Choose **Reinitialize mass** to reinitialize masses of elements to their original values. This option allows you to prevent the scaled mass from a previous step from being used in the current step. For more information, see *Reverting the Mass Matrix to the Original State*.
- Choose **Disable mass scaling thoughout step** to disable in this step all variable mass scaling definitions from previous steps. For more information, see *Continuous Mass Matrix with No Further Scaling*.

5. If you selected **Semi-automatic mass scaling**, **Automatic mass scaling**, or **Reinitialize mass**, indicate the region to which you want the mass scaling definition applied:
   - Choose **Whole model** to apply the mass scaling definition to all elements in the model.
   - Choose **Set** to apply the mass scaling definition to a particular set of elements. Click the arrow to the right of the **Set** field, and select the set name of interest.

6. If you selected **Semi-automatic mass scaling**, indicate when, during the step, you want Abaqus/Explicit to scale the element masses:
   - Choose **At beginning of step** to perform fixed mass scaling only at the beginning of the step. For more information, see *Fixed Mass Scaling*.
   - Choose **Throughout step** to scale the mass of elements periodically during the step. For more information, see *Variable Mass Scaling*.

7. If you selected **Semi-automatic mass scaling**, indicate how you want Abaqus/Explicit to scale the element masses:
   - Toggle on **Scale by factor** to scale the elements once at the beginning of the step by the value you enter in the field provided. For more information, see *Defining a Scale Factor Directly*.
   - Toggle on **Scale to target time increment of *n*** to enter a desired element stable time increment in the field provided. Click the arrow to the right of the **Scale element mass** field, and select how you want Abaqus/Explicit to apply that target time increment:
     - Select **Uniformly to satisfy target** to scale the masses of the elements equally so that the smallest element stable time increment of the scaled elements equals the target value.
     - Select **If below minimum target** to scale the masses of only the elements whose element stable time increments are less than the target value.
     - Select **Nonuniformly to equal target** to scale the masses of all elements so that they all have the same element stable time increment equal to the target value.

     For more information, see *Defining a Desired Element-by-Element Stable Time Increment*.

   If you toggle on both **Scale by factor** and **Scale to target time increment**, Abaqus/Explicit first scales the masses by the factor value that you enter and then possibly scales them again, depending on the value you enter for target time increment and the option you select for applying that target.

8. If you selected **Automatic mass scaling**, enter the following values:
   a. In the **Feed rate** field, enter the estimated average velocity of the workpiece in the rolling direction at steady-state conditions.
   b. In the **Extruded element length** field, enter the average element length in the rolling direction.
   c. In the **Nodes in cross-section** field, enter the number of nodes in the cross-section of the workpiece. Increasing this value decreases the amount of mass scaling.

For more information, see *Automatic Mass Scaling for Analysis of Bulk Metal Rolling*.

9. If you selected **Semi-automatic mass scaling** throughout the step or **Automatic mass scaling**, specify when, during the step, you want Abaqus/Explicit to perform mass scaling calculations:

   - Choose **Every *n* increments** to specify the frequency, in increments, at which Abaqus/Explicit is to perform mass scaling calculations. Enter the desired frequency in the field provided.

     For example, if you enter a value of 5, Abaqus/Explicit scales the mass at the beginning of the step and at increments 5, 10, 15, etc.

   - Choose **At *n* equal intervals** to specify the number of intervals during the step at which Abaqus/Explicit is to perform mass scaling calculations. Enter the desired value in the field provided.

     For example, if you enter a value of 2, Abaqus/Explicit scales the mass at the beginning of the step, the increment immediately following the half-way point in the step, and the final increment in the step.

10. Click **OK** to close the **Edit mass scaling** dialog box and return to the **Mass scaling** tabbed page of the **Edit Step** dialog box.

    The mass scaling definition that you have just created appears in the **Data** table.

11. If desired, repeat Steps 3 to 10 to create additional mass scaling definitions.
12. Once you have created one or more mass scaling definitions, you can edit or delete them if desired. Select a particular mass scaling definition in the **Data** table, and click **Edit** or **Delete** at the bottom of the **Data** table.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Enter a value for the **Linear bulk viscosity parameter**. Linear bulk viscosity is included by default in Abaqus/Explicit.
3. Enter a value for the **Quadratic bulk viscosity parameter**. This form of bulk viscosity pressure is found only in solid continuum element and is applied only if the volumetric strain rate is compressive.

   For more information, see *Bulk Viscosity*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a geostatic stress field procedure

A geostatic stress field procedure allows you to verify that the initial geostatic stress field is in equilibrium with applied loads and boundary conditions. It also allows you to iterate, if necessary, to obtain equilibrium; or you can allow Abaqus to compute equilibrium automatically for cases in which the initial state is unknown. This type of procedure is usually the first step of a geotechnical analysis, followed by a coupled pore fluid diffusion/stress or static analysis procedure. For more information, see *Geostatic Stress State*.

Create or edit a geostatic stress field procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Geostatic`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, configure settings such as controls to include nonlinear effects of large displacements and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Select an **Nlgeom** option:

   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to use a fixed increment size.

     If you select **Fixed**, no further entries are available on the **Incrementation** tabbed page.

3. If you selected **Automatic** incrementation in Step 2, enter values for the **Increment size** and for the **Max. displacement change**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.
   d. In the **Max. displacement change** field, enter the maximum amount of displacement that is acceptable while Abaqus/Standard calculates the equilibrium state for models in which the initial stress state is unknown or an approximation.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose an **Equation Solver Method** option:

   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.

3. Choose a **Matrix storage** option:

   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

4. Choose a **Solution technique**:

   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

     If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

     For more information, see *Quasi-Newton solution technique*.

5. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a mass diffusion procedure

A mass diffusion analysis models the transient or steady-state diffusion of one material through another, such as the diffusion of hydrogen through a metal. The governing equations for mass diffusion are an extension of Fick's equations: they allow for nonuniform solubility of the diffusing substance in the base material and for mass diffusion driven by gradients of temperature and pressure. For more information, see *Mass Diffusion Analysis*.

Create or edit a mass diffusion procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Mass diffusion`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as steady-state or transient response and automatic or fixed incrementation as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Response** option:

   - Choose **Steady-state** to specify that the analysis provide the steady-state solution directly. The rate of change of concentration with respect to time is omitted from the governing diffusion equation in a steady-state analysis. For more information, see *Steady-State Analysis*.
   - Choose **Transient** to perform time integration with the backward Euler method. This method is unconditionally stable for linear problems. For more information, see *Transient Analysis*.

   Note:

   After you have selected a **Response** option, a message appears informing you that Abaqus/Standard has selected the **Default load variation with time** option (located on the **Other** tabbed page) that corresponds to your **Response** selection. Click **Dismiss** to close the message dialog box.

4. In the **Time period** field, enter the time period of the step.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** incrementation in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.

b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.

c. In the **Maximum** field, enter the maximum time increment allowed.

5. If you selected **Fixed** incrementation in Step 2, enter a value for the constant time increment in the **Increment size** field.

6. If you selected **Automatic** incrementation in Step 2 and **Transient** analysis on the **Basic** tabbed page, do the following:

a. Enter a value in the **End step when normalized concentration change is less than *n*** field. The analysis will end when all nodal normalized concentrations are changing at a rate that is less than the rate that you enter.

b. Enter a value in the **Max. allowable normalized concentration change** field. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node (except nodes with boundary conditions) during any increment of the step.

To configure settings on the **Other** tabbed page:

a. In the **Edit Step** dialog box, display the **Other** tabbed page.

(For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

b. Accept the selection of the **Unsymmetric** matrix storage and solution scheme. This scheme is the only **Matrix storage** option that is valid for mass diffusion analyses. For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

c. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

- Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
- Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
- Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

d. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.

e. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

- Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
- Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
- Select **None** to suppress any extrapolation.

For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring an effective stress analysis for fluid-filled porous media

A coupled pore fluid diffusion/stress analysis allows you to model single phase, partially or fully saturated fluid flow through porous media. For more information, see *Coupled Pore Fluid Diffusion and Stress Analysis*.

Create or edit a coupled pore fluid diffusion/stress procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Soils`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as steady-state or transient pore fluid response and automatic or fixed incrementation as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Pore fluid response** option:

   - Choose **Steady-state** to specify that there are no transient effects in the wetting liquid continuity equation. The steady-state solution corresponds to constant wetting liquid velocities and constant volume of wetting liquid per unit volume in the continuum. For more information, see *Steady-State Analysis*.
   - Choose **Transient consolidation** to use the backward difference operator to integrate the continuity equation. This operator provides unconditional stability so that the only concern with respect to time integration is accuracy. For more information, see *Transient Analysis*.

   Note:

   After you have selected a **Pore fluid response** option, a message appears informing you that Abaqus/Standard has selected the **Default load variation with time** option and the **Matrix storage** option (both located on the **Other** tabbed page) that correspond to your **Pore fluid response** selection. Click **Dismiss** to close the message dialog box.

4. In the **Time period** field, enter the time period of the step.
5. Select an **Nlgeom** option:

   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.

6. Select an automatic stabilization method if you expect the problem to have local instabilities such as surface wrinkling, material instability, or local buckling. Abaqus/Standard can stabilize this class of problems by applying damping throughout the model. For more information, see *Unstable Problems*, and *Automatic Stabilization of Static Problems with a Constant Damping Factor*

   Click the arrow to the right of **Automatic stabilization**, and select a method for defining the damping factor:

   - Select **Specify dissipated energy fraction** to allow Abaqus/Standard to calculate the damping factor from a dissipated energy fraction that you provide. Enter a value for the dissipated energy

fraction in the adjacent field (the default is 2.0×10^{-4}). For more information, see *Calculating the Damping Factor Based on the Dissipated Energy Fraction*.

- Select **Specify damping factor** to enter the damping factor directly. Enter a value for the damping factor in the adjacent field. For more information, see *Directly Specifying the Damping Factor*.
- Select **Use damping factors from previous general step** to use the damping factors at the end of the previous step as the initial factors in the current step's variable damping scheme. These factors override any initial damping factors that are calculated or specified directly in the current step. If there are no damping factors associated with the previous general step (for example, if the previous step does not use any stabilization or the current step is the first step of the analysis), Abaqus uses adaptive stabilization to determine the required damping factors.

7. When using automatic stabilization, Abaqus can use the same damping factor over the course of a step, or it can vary the damping factor spatially and temporally during a step based on the convergence history and the ratio of the energy dissipated by damping to the total strain energy. For more information, see *Adaptive Automatic Stabilization Scheme*. If you selected **Specify dissipated energy fraction**, adaptive stabilization is optional and turned on by default. If you selected **Specify damping factor**, adaptive stabilization is optional and turned off by default. If you selected **Use damping factors from previous general step**, adaptive stabilization is required.

   To use adaptive stabilization, toggle on **Use adaptive stabilization with max. ratio of stabilization to strain energy** (if necessary), and enter a value in the adjacent field for the allowable accuracy tolerance for the ratio of energy dissipated by damping to total strain energy in each increment. The default value of 0.05 should be suitable in most cases.

8. If desired, toggle on **Include creep/swelling/viscoelastic behavior**. If you leave this option toggled off, you indicate that there is no creep or viscoelastic response occurring during this step even if creep or viscoelastic material properties have been defined.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to determine suitable time increment sizes.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

     Note:

     Fixed incrementation is not generally recommended in this case because the time increments in a typical diffusion analysis can increase over several orders of magnitude during the simulation; automatic incrementation is usually a better choice.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, enter values for **Increment size**:
   a. In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
   b. In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
   c. In the **Maximum** field, enter the maximum time increment allowed.

5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. If you selected the **Transient consolidation** response on the **Basic** tabbed page, toggle on **End step when pore pressure change rate is less than *n*** to enter a minimum value for the pore pressure change rate. The analysis will end if all pore pressures are changing at a rate that is less than the rate that you enter.
7. If you selected **Automatic** in Step 2, do the following:
   a. If you selected the **Transient consolidation** response on the **Basic** tabbed page, enter a value for the **Max. pore pressure change per increment**. Abaqus/Standard restricts the time step to ensure that this value is not exceeded at any node (except nodes with boundary conditions) during any increment of the step.
   b. If you toggled on **Include creep/swelling/viscoelastic behavior** on the **Basic** tabbed page, toggle on **Creep/swelling/viscoelastic strain error tolerance** to enter the maximum difference in the creep strain increment calculated from the creep strain rates at the beginning and at the end of the increment. This value controls the accuracy of the creep integration. For more information, see *Specifying the Tolerance for Automatic Incrementation*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose an **Equation Solver Method** option:
   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.
3. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.

     Note:

     The steady-state coupled equations are strongly unsymmetric; therefore, the unsymmetric matrix solution and storage scheme is selected automatically for steady-state analysis steps (see *Defining an Analysis*).
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
4. Choose a **Solution technique**:
   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

For more information, see *Quasi-Newton solution technique*.

5. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:
   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.
6. Abaqus/Standard automatically selects the **Default load variation with time** option that corresponds to your **Pore fluid response** selection on the **Basic** tabbed page. It is recommended that you leave the **Default load variation with time** selection unchanged.
7. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:
   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a transient, static, stress/displacement analysis with time-dependent material response

You can use a quasi-static stress analysis to analyze problems with time-dependent material response (creep, swelling, viscoelasticity, and two-layer viscoplasticity). This type of analysis is valid when inertial effects can be neglected. It can be linear or nonlinear. For more information, see *Quasi-Static Analysis*.

Create or edit a quasi-static stress analysis procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Visco`), or *Editing a step*.
2. On the **Basic**, **Incrementation**, and **Other** tabbed pages, configure settings such as the time period, automatic or fixed incrementation, and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Time period** field, enter the time period of the step.
4. Select an **Nlgeom** option:

   - Toggle **NlgeomOff** to perform a geometrically linear analysis during the current step.
   - Toggle **NlgeomOn** to indicate that Abaqus/Standard should account for geometric nonlinearity during the step. Once you have toggled **Nlgeom** on, it will be active during all subsequent steps in the analysis.

   For more information, see *Linear and nonlinear procedures*.

5. Select an automatic stabilization method if you expect the problem to have local instabilities such as surface wrinkling, material instability, or local buckling. Abaqus/Standard can stabilize this class of problems by applying damping throughout the model. For more information, see *Unstable Problems*, and *Automatic Stabilization of Static Problems with a Constant Damping Factor*

   Click the arrow to the right of **Automatic stabilization**, and select a method for defining the damping factor:

   - Select **Specify dissipated energy fraction** to allow Abaqus/Standard to calculate the damping factor from a dissipated energy fraction that you provide. Enter a value for the dissipated energy fraction in the adjacent field (the default is 2.0×10^{-4}). For more information, see *Calculating the Damping Factor Based on the Dissipated Energy Fraction*.
   - Select **Specify damping factor** to enter the damping factor directly. Enter a value for the damping factor in the adjacent field. For more information, see *Directly Specifying the Damping Factor*.
   - Select **Use damping factors from previous general step** to use the damping factors at the end of the previous step as the initial factors in the current step's variable damping scheme. These factors override any initial damping factors that are calculated or specified directly in the current step. If there are no damping factors associated with the previous general step (for example, if the previous step does not use any stabilization or the current step is the first step of the analysis), Abaqus uses adaptive stabilization to determine the required damping factors.

6. When using automatic stabilization, Abaqus can use the same damping factor over the course of a step, or it can vary the damping factor spatially and temporally during a step based on the convergence history and the ratio of the energy dissipated by damping to the total strain energy. For more information,

see *Adaptive Automatic Stabilization Scheme*. If you selected **Specify dissipated energy fraction**, adaptive stabilization is optional and turned on by default. If you selected **Specify damping factor**, adaptive stabilization is optional and turned off by default. If you selected **Use damping factors from previous general step**, adaptive stabilization is required.

To use adaptive stabilization, toggle on **Use adaptive stabilization with max. ratio of stabilization to strain energy** (if necessary), and enter a value in the adjacent field for the allowable accuracy tolerance for the ratio of energy dissipated by damping to total strain energy in each increment. The default value of 0.05 should be suitable in most cases.

*Configure settings on the **Incrementation** tabbed page*

1. In the **Edit Step** dialog box, display the **Incrementation** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose a **Type** option:

   - Choose **Automatic** if you want Abaqus/Standard to select time increments automatically based on the accuracy of the integration. A **Creep/swelling/viscoelastic strain error tolerance** parameter that you specify limits the maximum inelastic strain rate change allowed over an increment. **Automatic** incrementation is recommended for almost all cases.
   - Choose **Fixed** to specify direct user control of the incrementation. Abaqus/Standard uses an increment size that you specify as the constant increment size throughout the step.

3. In the **Maximum number of increments** field, enter the upper limit to the number of increments in the step. The analysis stops if this maximum is exceeded before Abaqus/Standard arrives at the complete solution for the step.
4. If you selected **Automatic** in Step 2, do the following:
   a. Enter values for **Increment size**:

      - In the **Initial** field, enter the initial time increment. Abaqus/Standard modifies this value as required throughout the step.
      - In the **Minimum** field, enter the minimum time increment allowed. If Abaqus/Standard needs a smaller time increment than this value, it terminates the analysis.
      - In the **Maximum** field, enter the maximum time increment allowed.

   b. In the **Creep/swelling/viscoelastic strain error tolerance** field, enter the maximum difference in the creep strain increment calculated from the creep strain rates at the beginning and at the end of the increment. This value controls the accuracy of the creep integration. For more information, see *Automatic Incrementation*.
5. If you selected **Fixed** in Step 2, enter a value for the constant time increment in the **Increment size** field.
6. Choose a **Creep/swelling/viscoelastic integration** option:

   - Choose **Explicit/Implicit** if you want to allow Abaqus/Standard to invoke the implicit integration scheme. For creep at very low stress levels the unconditional stability of the backward difference operator (implicit method) is desirable.
   - Choose **Explicit** if you want to restrict Abaqus/Standard to using explicit integration. Explicit integration can be less expensive computationally and simplifies implementation of user-defined creep laws in user subroutine *CREEP*

   For more information, see *Selecting Explicit Creep Integration*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Choose an **Equation Solver Method** option:

   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.

3. Choose a **Matrix storage** option:

   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

4. Choose a **Solution technique**:

   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations. For more information, see *Nonlinear solution methods in Abaqus/Standard*.
   - Choose **Quasi-Newton** to use the quasi-Newton technique for solving nonlinear equilibrium equations. This technique can save substantial computational cost in some cases. Generally it is most successful when the system is large and the stiffness matrix is not changing much from iteration to iteration. You can use this technique only for symmetric systems of equations.

     If you choose this technique, enter a value for the **Number of iterations allowed before the kernel matrix is reformed**. The maximum number of iterations allowed is 25. The default number of iterations is 8.

     For more information, see *Quasi-Newton solution technique*.

5. Click the arrow to the right of the **Convert severe discontinuity iterations** field, and select an option for dealing with severe discontinuities during nonlinear analysis:

   - Select **Off** to force a new iteration if severe discontinuities occur during an iteration, regardless of the magnitude of the penetration and force errors. This option also changes some time incrementation parameters and uses different criteria to determine whether to do another iteration or to make a new attempt with a smaller increment size.
   - Select **On** to use local convergence criteria to determine whether a new iteration is needed. Abaqus/Standard will determine the maximum penetration and estimated force errors associated with severe discontinuities and check whether these errors are within the tolerances. Hence, a solution may converge if the severe discontinuities are small.
   - Select **Propagate from previous step** to use the value specified in the previous general analysis step. This value appears in parentheses to the right of the field.

   For more information on severe discontinuities, see *Severe Discontinuities in Abaqus/Standard*.

6. Choose an option for **Default load variation with time**:

   - Choose **Instantaneous** if you want loads to be applied instantaneously at the start of the step and remain constant throughout the step.

- Choose **Ramp linearly over step** if the load magnitude is to vary linearly over the step, from the value at the end of the previous step to the full magnitude of the load.

7. Click the arrow to the right of the **Extrapolation of previous state at start of each increment** field, and select a method for determining the first guess to the incremental solution:

   - Select **Linear** to indicate that the process is essentially monotonic and Abaqus/Standard should use a 100% linear extrapolation, in time, of the previous incremental solution to begin the nonlinear equation solution for the current increment.
   - Select **Parabolic** to indicate that the process should use a quadratic extrapolation, in time, of the previous two incremental solutions to begin the nonlinear equation solution for the current increment.
   - Select **None** to suppress any extrapolation.

   For more information, see *Extrapolation of the Solution*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring an annealing procedure

The anneal procedure is intended to simulate the relaxation of stresses and plastic strains that occurs as metals are heated to high temperatures.
Physically, annealing is the process of heating a metal part to a high temperature to allow the microstructure to recrystallize, removing dislocations caused by cold working of the material.

The anneal procedure can be created following a dynamic, explicit procedure or a dynamic fully coupled thermal-stress procedure using explicit integration (see *Configuring a dynamic, explicit procedure* and *Configuring a dynamic fully coupled thermal-stress procedure using explicit integration*.) During the anneal procedure Abaqus/Explicit sets all appropriate state variables to zero. These variables include stresses, backstresses, plastic strains, and velocities. In the case of metal porous plasticity, the void volume fraction is also set to zero, such that the material becomes fully dense.

There is no time scale in an annealing step; therefore, time does not advance. The annealing process occurs instantaneously. No data are required for the anneal procedure.

For more information, see *Annealing*.

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `General; Anneal`), or *Editing a step*.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Post-anneal reference temperature** option:
   - Choose **Maintain current** to maintain the current temperature at all nodes in the model after the annealing is complete.
   - Choose **Value** to specify a final temperature to which all nodes in the model will be set after the annealing is complete. Enter the value in the field provided.
4. Click **OK** to close the **Edit Step** dialog box.

Configuring linear perturbation analysis procedures

You can configure linear perturbation procedures to analyze linear problems.
Linear perturbation procedures are available only in Abaqus/Standard. The linear perturbation approach allows general application of linear analysis techniques in cases where the linear response depends on preloading or on the nonlinear response history of the model. For more information, see *General and Perturbation Procedures*.

This section provides instructions for using the step editor to configure different types of linear perturbation procedures.

In this section:

- *Configuring a static, linear perturbation procedure*
- *Configuring a frequency procedure*
- *Configuring a buckling procedure*
- *Configuring a complex frequency procedure*
- *Configuring a modal dynamics procedure*
- *Configuring a random response procedure*
- *Configuring a response spectrum procedure*
- *Configuring a direct-solution steady-state dynamic procedure*
- *Configuring a mode-based steady-state dynamic analysis*
- *Configuring a subspace-based steady-state dynamic analysis*
- *Configuring a substructure generation procedure*
- *Configuring a time-harmonic electromagnetic analysis*

Configuring a static, linear perturbation procedure

The response in a linear analysis step is the linear perturbation response about the *base state*. The base state is the current state of the model at the end of the last general analysis step prior to the linear perturbation step. If the first step of an analysis is a perturbation step, the base state is determined from the initial conditions.

For more information, see *Linear Perturbation Analysis Steps*.

Create or edit a static, linear perturbation procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Static, Linear perturbation`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, describe the step and enter equation solver preferences as described in the following procedures.

Configure a static, linear perturbation procedure

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Display the **Other** tabbed page.
4. Choose an **Equation Solver Method** option:
   - Choose **Direct** to use the default direct sparse solver.
   - Choose **Iterative** to use the iterative linear equation solver. The iterative solver is typically most useful for blocky structures with millions of degrees of freedom. For more information, see *Iterative Linear Equation Solver*.
5. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
6. Choose a **Solution technique** option:
   - Choose **Full Newton** to use Newton's method as a numerical technique for solving nonlinear equilibrium equations.
   - Choose **LCP Contact** to activate the linearized contact capability to solve for contact status and contact stresses. This option can be specified only within a static perturbation step with small sliding, frictionless contact.

     If you choose this technique, enter a value for the **Gap distance** to specify a zone within which potential contact constraints are exposed to the LCP solver and **Scale factor** for the initial gap distance specified in the **Gap distance** or the internally computed (contact) pair-wise gap distances. The default is 1.0.
7. When you have finished configuring settings for the static, linear perturbation step, click **OK** to close the **Edit Step** dialog box.

Configuring a frequency procedure

This section provides detailed instructions for configuring a frequency extraction procedure.

In this section:

- *Defining frequency extraction procedures*
- *Using the Lanczos eigensolver for a frequency extraction procedure*
- *Using the AMS eigensolver for a frequency extraction procedure*
- *Using the subspace iteration eigensolver for a frequency extraction procedure*

Defining frequency extraction procedures

When you have configured a step for a frequency procedure, Abaqus/Standard performs an eigenvalue extraction procedure during that step to calculate the natural frequencies and the corresponding mode shapes of a system.

When you configure a frequency step, you must choose one of the following eigenvalue extraction methods:

Lanczos

The Lanczos method is the default method because it has general capabilities. However, this method is usually slower than the AMS method. For more information, see *Lanczos Eigensolver* and *Eigenvalue extraction*.

For detailed instructions on configuring settings for the Lanczos eigenvalue extraction method, see *Using the Lanczos eigensolver for a frequency extraction procedure*.

Automatic multi-level substructuring (AMS)

The AMS method is an add-on analysis capability for Abaqus/Standard. The AMS method is faster than the Lanczos method, particularly when you require a large number of eigenmodes for a system with many degrees of freedom. However, the AMS method has several limitations. For more information, see *Automatic Multilevel Substructuring (AMS) Eigensolver*.

For detailed instructions on configuring settings for the AMS extraction method, see *Using the AMS eigensolver for a frequency extraction procedure*.

Subspace iteration

For the subspace iteration procedure you need only specify the number of eigenvalues required; Abaqus/Standard chooses a suitable number of vectors for the iteration. You can also specify the maximum frequency of interest; Abaqus/Standard extracts eigenvalues until either the requested number of eigenvalues has been extracted or the last frequency extracted exceeds the maximum frequency of interest. For more information, see *Eigenvalue extraction*.

For detailed instructions on configuring settings for the subspace iteration extraction method, see *Using the subspace iteration eigensolver for a frequency extraction procedure*.

For background information on frequency extraction procedures, see *Natural Frequency Extraction*.

Using the Lanczos eigensolver for a frequency extraction procedure

The **Edit Step** dialog box provides **Basic** and **Other** tabbed pages on which you can specify settings for the Lanczos eigensolver.

You can display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:**`Linear perturbation; Frequency`), or *Editing a step*.

Configure settings on the **Basic** tabbed page

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. From the list of **Eigensolver** options, choose **Lanczos**.
4. Choose one of the **Number of eigenvalues requested** options to indicate how many eigenvalues you want calculated:
   - Choose **All in frequency range** if you want Abaqus/Standard to calculate all the eigenvalues within a range determined by a maximum frequency value (and, if desired, a minimum frequency value) that you enter.
   - Choose **Value** if you want a particular number of eigenvalues to be calculated. Enter that value in the field provided.
5. Toggle on **Frequency shift (cycles/time)**2** to specify a positive or negative shifted squared frequency, *S*. If you toggle this option on, enter a value for the frequency shift in the field provided. For more information, see *Frequency Shift*.
6. Toggle on **Minimum frequency of interest (cycles/time)** to specify a lower limit to the frequency range within which Abaqus/Standard will calculate eigenvalues. If you toggle on this option, enter a value for the minimum frequency in the field provided.

   This option is required if you intend to use the Lanczos solver in parallel mode.
7. Toggle on **Maximum frequency of interest (cycles/time)** to specify an upper limit to the frequency range within which Abaqus/Standard will calculate eigenvalues. If you toggle on this option, enter a value for the maximum frequency in the field provided.

   This option is required if you intend to use the Lanczos solver in parallel mode.
8. If the model includes acoustic and structural elements coupled together or ASI-type elements, you can toggle on **Include acoustic-structural coupling where applicable** to include the effect of acoustic-structural coupling during the frequency extraction. (This option is toggled on by default.) For more information, see *Structural-Acoustic Coupling*.
9. Specify your preference for block size:
   - Choose **Default** to use the default block size of 7, which is usually appropriate.
   - Choose **Value** to enter a particular block size. Enter the value in the field provided. In general, the block size for the Lanczos method should be as large as the largest expected multiplicity of eigenvalues.

   For more information, see *Lanczos Eigensolver*.
10. Specify your preference for the **Maximum number of block Lanczos steps**:
    - Choose **Default** to allow Abaqus/Standard to determine the number of block Lanczos steps within each Lanczos run. The default value is usually appropriate.
    - Choose **Value** to enter a limit to the number of Lanczos steps within each Lanczos run. Enter the value in the field provided. In general, if a particular type of eigenproblem converges slowly,

providing more block Lanczos steps will reduce the analysis cost. On the other hand, if you know that a particular type of problem converges quickly, providing fewer block Lanczos steps will reduce the amount of in-core memory used.

11. Toggle on **Use SIM-based linear dynamics procedures** to activate the high performance, mode-based linear dynamics analysis capability. If the eigenmodes that are extracted in this step will be used for subsequent mode-based or subspace-based linear dynamic procedures, the SIM-based capability offers improved efficiency for large models with minimal output requests. For more information, see *Using the SIM Architecture for Modal Superposition Dynamic Analyses*.
12. If you toggled on **Use SIM-based linear dynamics procedures**, Abaqus projects structural and viscous damping operators defined in the model for use in a later, steady-state dynamic step. To deactivate the projection of damping operators, toggle off **Project damping operators**. Deactivating the damping operators may improve the performance of the frequency extraction step; however, structural and viscous material damping will then be ignored in subsequent steady-state dynamic steps. For more information, see *Damping in a Mode-Based Steady-State and Transient Linear Dynamic Analysis Using the SIM Architecture*.
13. Toggle on **Include residual modes** to request that Abaqus/Standard compute residual modes based on the loads specified in the immediately preceding static, linear perturbation step. For more information, see *Residual Modes*.

Configure settings on the **Other** tabbed page

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
2. Accept the default **Matrix storage** setting. Since Abaqus/Standard provides eigenvalue extraction only for symmetric matrices, steps with eigenfrequency extraction or eigenvalue buckling prediction procedures always use the symmetric matrix storage and solution scheme. You cannot change this setting. In such steps Abaqus/Standard will symmetrize all contributions to the stiffness matrix.
3. Choose an option for normalizing eigenvectors:

   - Choose **Displacement** to normalize the eigenvectors so that the largest displacement, rotation, or acoustic pressure entry in each vector is unity.
   - Choose **Mass** to normalize the eigenvectors with respect to the structure's mass matrix (the eigenvectors are scaled so that the generalized mass for each vector is unity).

   If you toggled on **Use SIM-based linear dynamics procedures** on the **Basic** tabbed page, only mass normalization is available. For more information, see *Normalization*.
4. Toggle on **Evaluate dependent properties at frequency** to evaluate frequency-dependent properties for viscoelasticity, springs, and dashpots during the eigenvalue extraction. If you toggle this option on, enter the desired evaluation frequency in the field provided. For more information, see *Evaluating Frequency-Dependent Material Properties*.

When you have finished configuring settings for the frequency step, click **OK** to close the **Edit Step** dialog box.

Using the AMS eigensolver for a frequency extraction procedure

The **Edit Step** dialog box provides **Basic** and **Other** tabbed pages on which you can specify settings for the AMS eigensolver.

You can display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:**`Linear perturbation; Frequency`), or *Editing a step*.

Configure settings on the **Basic** tabbed page

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. From the list of **Eigensolver** options, choose **AMS**.
4. Coupling of structural and acoustic regions does not affect a frequency extraction procedure that uses the AMS eigensolver: the structural and acoustic modes are computed as if uncoupled. By default, Abaqus projects structural-acoustic coupling operators defined in the model for use in a later, steady-state dynamic step. To deactivate the projection, toggle off **Project acoustic-structural coupling where applicable**. Deactivating the coupling operators may improve the analysis speed, but structural-acoustic coupling will then be ignored in subsequent steady-state dynamic steps.
5. For models that include both structural and acoustic elements, Abaqus by default uses the same frequency range when calculating the acoustic eigenvalues as it does when calculating the structural eigenvalues. To specify different frequency ranges for the two regions, enter an **Acoustic range factor**. Abaqus multiplies this factor by the **Maximum frequency of interest** to determine a different maximum frequency for the acoustic region. The minimum frequency remains the same for both regions.
6. Toggle on **Minimum frequency of interest (cycles/time)** to specify a lower limit to the frequency range within which Abaqus/Standard will calculate eigenvalues. If you toggle on this option, enter a value for the minimum frequency in the field provided.
7. In the **Maximum frequency of interest (cycles/time)** field, enter the upper limit to the frequency range within which Abaqus/Standard will extract all the modes.
8. Toggle on **Limit region of saved eigenvectors** if you want to limit eigenvector computation to only the nodes in a particular region. If you toggle on this option, click the arrow to the right of the field provided to select the region of interest.
9. By default, Abaqus projects any structural or viscous damping operators defined in the model for use in a later, steady-state dynamic step. To deactivate the projection of these operators, toggle off **Project damping operators**. Deactivating the damping operators may improve the performance of the frequency extraction step; however, structural and viscous material damping will then be ignored in subsequent steady-state dynamic steps. For more information, see *Damping in a Mode-Based Steady-State and Transient Linear Dynamic Analysis Using the SIM Architecture*.
10. Toggle on **Include residual modes** to request that Abaqus/Standard compute residual modes based on the static response of the model to a nominal (or unit) load.

    If you toggle on this option, specify the residual mode regions and the degrees of freedom (DOF) for which you want residual modes calculated. Abaqus/Standard computes one residual mode for every requested degree of freedom.

    For more information, see *Residual Modes*.

Configure settings on the **Other** tabbed page

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
2. Accept the default **Matrix storage** setting. Since Abaqus/Standard provides eigenvalue extraction only for symmetric matrices, steps with eigenfrequency extraction or eigenvalue buckling prediction procedures always use the symmetric matrix storage and solution scheme. You cannot change this setting. In such steps Abaqus/Standard will symmetrize all contributions to the stiffness matrix.

3. In the **Cutoff multiplier for substructure eigenproblems** field, enter AMS_{cutoff_1}, the cutoff frequency for substructure eigenproblems, defined as a multiplier of the maximum frequency of interest. The default value is 5.
4. In the **First cutoff multiplier for reduced eigenproblems** field, enter AMS_{cutoff_2}, the first cutoff frequency for a reduced eigenproblem, defined as a multiplier of the maximum frequency of interest.

   The default value is 1.6. $AMS_{\text{cutoff}_2} \leq AMS_{\text{cutoff}_1}$.
5. In the **Second cutoff multiplier for reduced eigenproblems** field, enter AMS_{cutoff_3}, the second cutoff frequency for a reduced eigenproblem, defined as a multiplier of the maximum frequency of interest.

   The default value is 1.3. $\mathbf{1.0} \leq AMS_{\text{cutoff}_3} \leq AMS_{\text{cutoff}_2}$.

   For more information, see *Automatic Multilevel Substructuring (AMS) Eigensolver*.

When you have finished configuring settings for the frequency step, click **OK** to close the **Edit Step** dialog box.

Using the subspace iteration eigensolver for a frequency extraction procedure

The **Edit Step** dialog box provides **Basic** and **Other** tabbed pages on which you can specify settings for the subspace iteration eigensolver.

You can display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:**`Linear perturbation; Frequency`), or *Editing a step*.

Configure setting on the **Basic** tabbed page

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. From the list of **Eigensolver** options, choose **Subspace**.
4. In the **Number of eigenvalues requested** field, enter the number of eigenvalues that you want Abaqus/Standard to calculate.
5. Toggle on **Frequency shift (cycles/time)******2** to specify a positive or negative shifted squared frequency, *S*. If you toggle this option on, enter a value for the frequency shift in the field provided. For more information, see *Frequency Shift*.
6. Toggle on **Maximum frequency of interest (cycles/time)** to specify an upper limit to the frequency range within which Abaqus/Standard will calculate eigenvalues. If you toggle on this option, enter a value for the maximum frequency in the field provided.

   Abaqus/Standard will extract eigenvalues until either the requested number of eigenvalues has been extracted or the last frequency extracted exceeds the maximum frequency of interest.
7. Enter a value for the number of **Vectors used per iteration**. In general, the convergence is more rapid with more vectors, but the memory requirement is also larger. Thus, if you know that a particular type of eigenproblem converges slowly, providing more vectors by using this option might reduce the analysis cost.
8. Enter a value for the **Maximum number of iterations**. The default is 30.

Configure settings on the **Other** tabbed page

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
2. Accept the default **Matrix storage** setting. Since Abaqus/Standard provides eigenvalue extraction only for symmetric matrices, steps with eigenfrequency extraction or eigenvalue buckling prediction procedures always use the symmetric matrix storage and solution scheme. You cannot change this setting. In such steps Abaqus/Standard will symmetrize all contributions to the stiffness matrix.
3. Choose an option for normalizing eigenvectors:

   - Choose **Displacement** to normalize the eigenvectors so that the largest displacement, rotation, or acoustic pressure entry in each vector is unity.
   - Choose **Mass** to normalize the eigenvectors with respect to the structure's mass matrix (the eigenvectors are scaled so that the generalized mass for each vector is unity).

   For more information, see *Normalization*.
4. Toggle on **Evaluate dependent properties at frequency** to evaluate frequency-dependent properties for viscoelasticity, springs, and dashpots during the eigenvalue extraction. If you toggle on this option, enter the desired evaluation frequency in the field provided. For more information, see *Evaluating Frequency-Dependent Material Properties*.

When you have finished configuring settings for the frequency step, click **OK** to close the **Edit Step** dialog box.

Configuring a buckling procedure

An eigenvalue buckling analysis is generally used to estimate the critical buckling loads of stiff structures (classical eigenvalue buckling). This type of analysis is a linear perturbation procedure, and buckling loads are calculated relative to the base state of the structure. For more information, see *Eigenvalue Buckling Prediction*.

Create or edit a buckling procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Buckle`), or *Editing a step*.
2. On the **Basic** tabbed page, configure settings such as eigensolver extraction method and maximum number of iterations as described in the following procedures.

   The **Other** tabbed page displays the default matrix storage option. You cannot change this setting because Abaqus/Standard provides eigenvalue extraction for only symmetric matrices. In eigenvalue buckling prediction procedures Abaqus/Standard symmetrizes all contributions to the stiffness matrix.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose either the **Lanczos** eigensolver or the **Subspace** iteration eigensolver.

   The Lanczos method is generally faster when a large number of eigenmodes is required for a system with many degrees of freedom. However, this method does have some limitations (see the warning at the bottom of the **Basic** tabbed page). The subspace iteration method may be faster when only a few (less than 20) eigenmodes are needed. For more information, see *Selecting the Eigenvalue Extraction Method*.
4. In the **Number of eigenvalues requested** field, enter the number of eigenvalues that you want to be estimated. Significant overestimation of the actual number of eigenvalues can create very large files. If you underestimate the actual number of eigenvalues, Abaqus/Standard will issue a corresponding warning message.
5. If you selected the **Lanczos** eigensolver, do the following:
   a. Toggle on **Minimum eigenvalue of interest** to enter a lower limit to the range of eigenvalues Abaqus/Standard will extract. If you toggle on this option, enter the value in the field provided.
   b. Toggle on **Maximum eigenvalue of interest** to enter an upper limit to the range of eigenvalues Abaqus/Standard will extract. If you toggle on this option, enter the value in the field provided.

      If you specify a range of eigenvalues, Abaqus/Standard will extract eigenvalues until either the requested number of eigenvalues has been extracted in the given range or all the eigenvalues in the given range have been extracted.
   c. Choose a **Block size** option:

      - Choose **Default** to use the default block size of 7, which is usually appropriate.
      - Choose **Value** to enter a particular block size in the field provided. In general, the block size for the Lanczos method should be as large as the largest expected multiplicity of eigenvalues.

      For more information, see *Selecting the Eigenvalue Extraction Method*.
   d. Specify your preference for the **Maximum number of block Lanczos steps**:

      - Choose **Default** to allow Abaqus/Standard to determine the number of block Lanczos steps within each Lanczos run. The default value is usually appropriate.

- Choose **Value** to enter a limit to the number of Lanczos steps within each Lanczos run. In general, if a particular type of eigenproblem converges slowly, providing more block Lanczos steps will reduce the analysis cost. On the other hand, if you know that a particular type of problem converges quickly, providing fewer block Lanczos steps will reduce the amount of in-core memory used.

6. If you selected the **Subspace** eigensolver, do the following:
   a. Toggle on **Maximum eigenvalue of interest** to enter an upper limit to the range of eigenvalues Abaqus/Standard will extract. If you toggle on this option, enter the value in the field provided.

      Abaqus/Standard will extract eigenvalues until either the requested number of eigenvalues has been extracted or the last eigenvalue extracted exceeds the maximum eigenvalue of interest.
   b. Enter a value for the number of **Vectors used per iteration**. In general, the convergence is more rapid with more vectors, but the memory requirement is also larger. Thus, if you know that a particular type of eigenproblem converges slowly, providing more vectors by using this option might reduce the analysis cost.
   c. Enter a value for the **Maximum number of iterations**. The default is 30.
7. Click **OK** to save the step and to close the **Edit Step** dialog box.

Configuring a complex frequency procedure

A complex frequency procedure performs eigenvalue extraction to calculate the complex eigenvalues and the corresponding complex mode shapes of a system. It is a linear perturbation procedure and requires that you perform an eigenfrequency extraction procedure (described in *Configuring a frequency procedure*") prior to the complex eigenvalue extraction. For more information, see *Complex Eigenvalue Extraction*.

Create or edit a complex frequency procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Complex frequency`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, configure settings such as the number of eigenvalues requested and matrix solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose an option for specifying the **Number of eigenvalues requested**:
   - Choose **All** if you want Abaqus/Standard to report all the eigenmodes available in the projected subspace, formulated on the basis of all eigenmodes computed in the preceding frequency step.
   - Choose **Value** to enter a specific number of eigenmodes that you want Abaqus/Standard to report.
4. Toggle on **Frequency shift (cycles/time)** to specify a shift point, S, for the complex eigenvalue extraction procedure ($S \geq 0$). Abaqus/Standard reports the complex eigenmodes, μ_n, in order of increasing $|Im(\mu_n) - S|$ so that the modes with the imaginary part closest to a given shift point are reported first. This feature is useful when a particular frequency range is of concern. The default is no shift.
5. Toggle on **Minimum frequency of interest (cycles/time)** to specify a lower limit to the frequency range within which Abaqus/Standard reports eigenmodes.
6. Toggle on **Maximum frequency of interest (cycles/time)** to specify an upper limit to the frequency range within which Abaqus/Standard reports eigenmodes.
7. Toggle on **Include friction-induced damping effects** to include friction-induced contributions to the damping matrix. For more information, see *Contact Conditions with Sliding Friction*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix solver** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.
3. Toggle on **Evaluate dependent properties at frequency** to enter a frequency at which Abaqus/Standard will evaluate frequency-dependent material properties. For more information, see *Evaluating Frequency-Dependent Material Properties*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a modal dynamics procedure

A transient model dynamic analysis gives the response of the model as a function of time based on a given time-dependent loading. The structure's response is based on a subset of the modes of the system, which must first be extracted using an eigenfrequency extraction procedure (described in *Configuring a frequency procedure*). For more information, see *Transient Modal Dynamic Analysis*.

Create or edit a modal dynamics procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:**`Linear perturbation; Modal dynamics`), or *Editing a step*.
2. On the **Basic** and **Damping** tabbed pages, configure settings such as whether or not to carry over initial conditions from the results of the preceding step and damping at particular modes or frequencies as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Indicate whether or not you want to carry over initial conditions from the immediately preceding step:
   - Choose **Use initial conditions (when applicable)** if you want Abaqus/Standard to carry over initial conditions from the immediately preceding step, which must be either another modal dynamic step or a static perturbation step:
     - If the immediately preceding step is a modal dynamic step, both the displacements and velocities are carried over from the end of that step and used as initial conditions for the current step.
     - If the immediately preceding step is a static perturbation step, the displacements are carried over from that step. If initial velocities have been defined (*Initial Conditions*), they will be used; otherwise, the initial velocities will be zero.
   - Choose **Zero initial conditions** If you want the modal dynamic step to begin with zero initial displacements. If you have defined initial velocities Abaqus/Standard will use them; otherwise, the initial velocities will be zero.
4. In the **Time period** field, enter the time period of the step.
5. In the **Time increment** field, enter a value for the desired time increment size.

*Configure settings on the **Damping** tabbed page*

1. In the **Edit Step** dialog box, display the **Damping** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Indicate how you want to provide damping values:
   - Choose **Specify damping over ranges of Modes** to provide damping values for specific mode ranges.
   - Choose **Specify damping over ranges of Frequencies** to provide damping values at specific frequencies. Abaqus/Standard interpolates the damping coefficient for a mode linearly between the specified frequencies

   If you omit damping data on the **Damping** tabbed page, Abaqus/Standard assumes zero damping values. For more information, see *Specifying Modal Damping*.

3. If you selected **Modes** in Step 2, select one or more of the following options for defining damping:

   - Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular mode range, and do the following:

     1. Toggle on **Use direct damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
        - **Critical Damping Fraction**: fraction of critical damping, ξ.

   - Display the **Composite modal** tabbed page to select composite modal damping using the damping coefficients calculated in the preceding frequency step. (The damping calculations performed in the frequency step are performed using damping data provided in the material definition). Do the following:

     1. Toggle on **Use composite damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.

   - Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:

     1. Toggle on **Use Rayleigh damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
        - **Alpha**: mass proportional damping, α_M.
        - **Beta**: stiffness proportional damping, β_M.

   For detailed information on how to enter data, see *Entering tabular data*.

4. If you selected **Frequencies** in Step 2, select one or both of the following options for defining damping:

   - Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular frequency range. Do the following:

     1. Toggle on **Use direct damping data**.
     2. Enter the following in the data table:

        - **Frequency**: frequency value in cycles/time.
        - **Critical Damping Fraction**: fraction of critical damping, ξ.

   - Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:

     1. Toggle on **Use Rayleigh damping data**.
     2. Enter the following in the data table:

- **Frequency**: frequency value in cycles/time.
- **Alpha**: mass proportional damping, α_M.
- **Beta**: stiffness proportional damping, β_M.

For detailed information on how to enter data, see *Entering tabular data*.

5. If desired, repeat Steps 2–4 to create multiple damping definitions.

Configuring a random response procedure

A random response analysis is a linear perturbation procedure that provides the linearized dynamic response of a model to user-defined random excitation. This type of analysis uses the set of modes extracted in a previous eigenfrequency extraction procedure (described in *Configuring a frequency procedure*") to calculate the power spectral densities of response variables and the corresponding root mean square values. For more information, see *Random Response Analysis*.

Create or edit a random response procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Random response`), or *Editing a step*.
2. On the **Basic** and **Damping** tabbed pages, configure settings such as frequency range and damping definitions as described in the following procedures.
3. The power spectral density of the Mises equivalent stress and the root mean square of the Mises equivalent stress are computed in the Visualization module. To obtain these data, you must request specific field and history output.
   a. Display the output request editors as described in *Creating an output request*.
   b. For the field output request of the previous frequency step, select the stress component and invariants (S) output variable.
   c. For the history output request of the random response step, do the following:
      - Select **Set** as the **Domain**, and select the desired set from the list.
      - To obtain the power spectral density of the Mises equivalent stress, select the MISES output variable.
      - To obtain the root mean square of the Mises equivalent stress, select the RMISES output variable.
   d. You can view the MISES and RMISES output variables in the Visualization module.
      - To obtain contour plots of the MISES and RMISES output variables, see *Selecting the primary field output variable*.
      - X–Y data plots can be obtained for the MISES and RMISES output variables at element nodal and unique nodal output variable positions. To obtain these plots, see *Reading X–Y data from output database field output*.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Scale** option to indicate whether you want the frequency range(s) of interest to be divided using a **Logarithmic** or **Linear** scale.
4. Enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time. If you enter zero, Abaqus/Standard assumes that results are required only at the frequency specified in the **Lower Frequency** column.

Number of Points

The number of points between eigenfrequencies at which the response should be calculated, including the end points, from the lower limit of the frequency range to the first eigenfrequency in the range; in each interval from eigenfrequency to eigenfrequency; and from the highest eigenfrequency in the range to the high limit of the frequency range. If the value given is less than two (or omitted), the default value of 20 points is assumed. Accurate RMS values can be obtained only if enough points are used so that Abaqus/Standard can integrate accurately over the frequency range.

Bias

The bias parameter. This parameter is useful only if you request results at 4 or more frequency points. It is used to bias the results points toward the ends of the intervals so that better resolution is obtained there, since the ends of each interval are the eigenfrequencies where the response amplitudes vary most rapidly. The default bias parameter is 3.0. For more information, see *The Bias Parameter*.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Damping** tabbed page*

1. In the **Edit Step** dialog box, display the **Damping** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Indicate how you want to provide damping values:

   - Choose **Specify damping over ranges of Modes** to provide damping values for specific mode ranges.
   - Choose **Specify damping over ranges of Frequencies** to provide damping values at specific frequencies. Abaqus/Standard interpolates the damping coefficient for a mode linearly between the specified frequencies

   If you omit damping data on the **Damping** tabbed page, Abaqus/Standard assumes zero damping values. For more information, see *Specifying Damping*.

3. If you selected **Modes** in Step 2, select one or more of the following options for defining damping:

   - Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular mode range, and do the following:

     1. Toggle on **Use direct damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
        - **Critical Damping Fraction**: fraction of critical damping, ξ.

- Display the **Composite modal** tabbed page to select composite modal damping using the damping coefficients calculated in the preceding frequency step. (The damping calculations performed in the frequency step are performed using damping data provided in the material definition). Do the following:

  1. Toggle on **Use composite damping data**.
  2. Enter the following in the data table:

     - **Start Mode**: the mode number of the lowest mode of a range.
     - **End Mode**: the mode number of the highest mode of a range.

- Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:

  1. Toggle on **Use Rayleigh damping data**.
  2. Enter the following in the data table:

     - **Start Mode**: the mode number of the lowest mode of a range.
     - **End Mode**: the mode number of the highest mode of a range.
     - **Alpha**: mass proportional damping, α_M.
     - **Beta**: stiffness proportional damping, β_M.

- Display the **Structural** tabbed page to define damping that is proportional to the internal forces but opposite in direction to the velocity. Do the following:

  1. Toggle on **Use structural damping data**.
  2. Enter the following in the data table:

     - **Start Mode**: the mode number of the lowest mode of a range.
     - **End Mode**: the mode number of the highest mode of a range.
     - **Damping Constant**: Damping factor, *s*.

For detailed information on how to enter data, see *Entering tabular data*.

4. If you selected **Frequencies** in Step 2, select one or more of the following options for defining damping:

- Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular frequency range. Do the following:

  1. Toggle on **Use direct damping data**.
  2. Enter the following in the data table:

     - **Frequency**: frequency value in cycles/time.
     - **Critical Damping Fraction**: fraction of critical damping, ξ.

- Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:

  1. Toggle on **Use Rayleigh damping data**.
  2. Enter the following in the data table:

     - **Frequency**: frequency value in cycles/time.

- **Alpha**: mass proportional damping, α_M.
- **Beta**: stiffness proportional damping, β_M.

- Display the **Structural** tabbed page to define damping that is proportional to the internal forces but opposite in direction to the velocity. Do the following:
  1. Toggle on **Use structural damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Damping Constant**: Damping factor, *s*.

For detailed information on how to enter data, see *Entering tabular data*.

5. If desired, repeat Steps 2–4 to create multiple damping definitions.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

You can view the root mean square of the Mises equivalent stress in the Visualization module, view the RMISES output variable from the **Field Output** dialog box, or create *X–Y* data from **ODB field output**. For more information on viewing field output, see *Selecting the field output to display*, and *Reading X–Y data from output database field output*.

Configuring a response spectrum procedure

You can use a response spectrum analysis to estimate the peak response (displacement, stress, etc.) of a structure to a particular base motion. The method is only approximate, but it is often a useful, inexpensive method for preliminary design studies. The response spectrum procedure is based on using a subset of the modes of the system, which must first be extracted by using the eigenfrequency extraction procedure (described in *Configuring a frequency procedure*"). For more information, see *Response Spectrum Analysis*.

Create or edit a response spectrum procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Response spectrum`), or *Editing a step*.
2. On the **Basic** and **Damping** tabbed pages, configure settings such as a damping coefficient and the methods for combining multidirectional excitations as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Click the arrow to the right of the **Excitations** field, and select a directional summation method.

   - For the following options, Abaqus/Standard sums the directional excitation components first and then performs the modal summation:

     - Select **Single direction** to sum the directional excitation components for a single direction algebraically.
     - Select **Multiple direction absolute sum** to sum the directional excitation components for multiple directions algebraically.

   - For the following options, Abaqus/Standard performs the modal summation first and then sums the directional excitation components:

     - Select **Multiple direction square root of the sum of squares** to sum the directional excitation components for multiple directions using the square root of the sum of the squares.
     - Select **Multiple direction thirty percent rule** to sum the directional excitation components for multiple directions using the 30% rule.
     - Select **Multiple direction forty percent rule** to sum the directional excitation components for multiple directions using the 40% rule.

   See *Directional Summation Methods*, for more information.
4. Click the arrow to the right of the **Summations** field, and select a modal summation method. For information on each of the methods, see *Modal Summation Methods*.
5. On the **First direction** tabbed page (and on the **Second direction** and **Third direction** tabbed pages if applicable), do the following:
   a. In the **Use response spectrum** field, select the spectrum amplitude to use for calculating the response. Alternatively, you can click  to create a new amplitude. (See *Defining a spectrum*," for more information.)
   b. Enter **Direction cosines X**, **Y**, and **Z**.
   c. In the **Scale factor** field, enter the factor multiplying the magnitudes in the response spectrum.

d. In the **Time duration** field, enter the time duration of the dynamic event from which the spectrum was created. This setting is applicable only when the **Double sum combination** modal summation method is specified.

For multiple direction excitations, you have the option of toggling on **Apply third direction** on the **Third direction** tabbed page to include data for a third direction.

*Configure settings on the **Damping** tabbed page*

1. In the **Edit Step** dialog box, display the **Damping** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. Indicate how you want to provide damping values:

   - Choose **Specify damping over ranges of Modes** to provide damping values for specific mode ranges.
   - Choose **Specify damping over ranges of Frequencies** to provide damping values at specific frequencies. Abaqus/Standard interpolates the damping coefficient for a mode linearly between the specified frequencies

   If you omit damping data on the **Damping** tabbed page, Abaqus/Standard assumes zero damping values. For more information, see *Specifying Damping*.

3. If you selected **Modes** in Step 2, select one or more of the following options for defining damping:

   - Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular mode range. Do the following:

     1. Toggle on **Use direct damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
        - **Critical Damping Fraction**: fraction of critical damping, ξ.

   - Display the **Composite modal** tabbed page to select composite modal damping using the damping coefficients calculated in the preceding frequency step. (The damping calculations performed in the frequency step are performed using damping data provided in the material definition.) Do the following:

     1. Toggle on **Use composite damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.

   - Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:

     1. Toggle on **Use Rayleigh damping data**.
     2. Enter the following in the data table:

        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.

- **Alpha**: mass proportional damping, α_M.
- **Beta**: stiffness proportional damping, β_M.

For detailed information on how to enter data, see *Entering tabular data*.

4. If you selected **Frequencies** in Step 2, select one or both of the following options for defining damping:

- Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular frequency range. Do the following:
  1. Toggle on **Use direct damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Critical Damping Fraction**: fraction of critical damping, ξ.

- Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:
  1. Toggle on **Use Rayleigh damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Alpha**: mass proportional damping, α_M.
     - **Beta**: stiffness proportional damping, β_M.

For detailed information on how to enter data, see *Entering tabular data*.

5. If desired, repeat Steps 2–4 to create multiple damping definitions.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a direct-solution steady-state dynamic procedure

In a direct solution steady-state dynamic procedure, Abaqus/Standard calculates the steady-state harmonic response directly in terms of the physical degrees of freedom of the model, using the mass, damping, and stiffness matrices of the system. For more information, see *Direct-Solution Steady-State Dynamic Analysis*.

Create or edit a direct-solution steady-state dynamic procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Steady-state dynamics, Direct`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, configure settings such as frequency range and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose one of the following options:
   - Choose **Compute real response only** if you want Abaqus/Standard to ignore damping terms. This option can significantly reduce computational time.
   - Choose **Compute complex response** if you want to include damping terms and allow a complex system matrix to be factored.

   For more information, see *Damping*.
4. Choose a **Scale** option to indicate whether you want the frequency range(s) of interest to be divided using a **Logarithmic** or **Linear** scale.
5. Toggle on **Use eigenfrequencies to subdivide each frequency range** if you want the frequency range(s) of interest to be subdivided using the system's eigenfrequencies. This option requires a preceding frequency step. For more information, see *Selecting the Type of Frequency Interval for Which Output Is Requested*.
6. Toggle on **Include friction-induced damping effects** to include friction-induced contributions to the damping matrix. For more information, see *Contact Conditions with Sliding Friction*.
7. Enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time. If you enter zero, Abaqus/Standard assumes that results are required only at the frequency specified in the **Lower Frequency** column.

Number of Points

The number of points in the frequency range at which results should be given.

If you toggled on **Use eigenfrequencies to subdivide each frequency range**, this is the number of points at which results should be given, including the end points, from the lower limit of the frequency range to the first eigenfrequency in the range; in each interval from eigenfrequency

to eigenfrequency; and from the highest eigenfrequency in the range to the high limit of the frequency range.

If you toggled off **Use eigenfrequencies to subdivide each frequency range**, this is the total number of points in the frequency range, including the end points.

Bias

The bias parameter. This parameter is useful only if you request results at 4 or more frequency points. It is used to bias the results points toward the ends of the intervals so that better resolution is obtained there. This option is recommended if you have toggled on **Use eigenfrequencies to subdivide each frequency range**, since the ends of each interval are the eigenfrequencies where the response amplitudes vary most rapidly.

For more information, see *The Bias Parameter*.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a mode-based steady-state dynamic analysis

You can configure a mode-based steady-state dynamic analysis to calculate the steady-state dynamic linearized response of a system to harmonic excitation. Abaqus/Standard calculates the response based on the system's eigenfrequencies and modes, which must first be extracted by using the eigenfrequency extraction procedure (described in *Configuring a frequency procedure*"). This type of procedure is computationally cheaper than direct-solution or subspace-based steady-state analysis, but is less accurate, in particular if significant material damping is present. For more information, see *Mode-Based Steady-State Dynamic Analysis*.

Create or edit a mode-based steady-state dynamic procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Steady-state dynamics, Modal`), or *Editing a step*.
2. On the **Basic** and **Damping** tabbed pages, configure settings such as frequency range and damping definitions as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose a **Scale** option to indicate whether you want the frequency range(s) of interest to be divided using a **Logarithmic** or **Linear** scale.
4. Toggle on **Use eigenfrequencies to subdivide each frequency range** if you want the frequency range(s) of interest to be subdivided using the system's eigenfrequencies. For more information, see *Selecting the Type of Frequency Interval for Which Output Is Requested*.
5. Enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time. If you enter zero, Abaqus/Standard assumes that results are required only at the frequency specified in the **Lower Frequency** column.

Number of Points

The number of points in the frequency range at which results should be given.

If you toggled on **Use eigenfrequencies to subdivide each frequency range**, this is the number of points at which results should be given, including the end points, from the lower limit of the frequency range to the first eigenfrequency in the range; in each interval from eigenfrequency to eigenfrequency; and from the highest eigenfrequency in the range to the high limit of the frequency range.

If you toggled off **Use eigenfrequencies to subdivide each frequency range**, this is the total number of points in the frequency range, including the end points.

Bias

The bias parameter. This parameter is useful only if you request results at 4 or more frequency points. It is used to bias the results points toward the ends of the intervals so that better resolution is obtained there. This option is recommended if you have toggled on **Use eigenfrequencies to subdivide each frequency range**, since the ends of each interval are the eigenfrequencies where the response amplitudes vary most rapidly.

For more information, see *The Bias Parameter*.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Damping** tabbed page*

1. In the **Edit Step** dialog box, display the **Damping** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Indicate how you want to provide damping values:
   - Choose **Specify damping over ranges of Modes** to provide damping values for specific mode ranges.
   - Choose **Specify damping over ranges of Frequencies** to provide damping values at specific frequencies. Abaqus/Standard interpolates the damping coefficient for a mode linearly between the specified frequencies

   If you omit damping data on the **Damping** tabbed page, Abaqus/Standard assumes zero damping values. For more information, see *Specifying Modal Damping*.
3. If you selected **Modes** in Step 2, select one or more of the following options for defining damping:
   - Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular mode range. Do the following:
     1. Toggle on **Use direct damping data**.
     2. Enter the following in the data table:
        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
        - **Critical Damping Fraction**: fraction of critical damping, ξ.
   - Display the **Composite modal** tabbed page to select composite modal damping using the damping coefficients calculated in the preceding frequency step. (The damping calculations performed in the frequency step are performed using damping data provided in the material definition). Do the following:
     1. Toggle on **Use composite damping data**.
     2. Enter the following in the data table:
        - **Start Mode**: the mode number of the lowest mode of a range.
        - **End Mode**: the mode number of the highest mode of a range.
   - Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:
     1. Toggle on **Use Rayleigh damping data**.

2. Enter the following in the data table:
   - **Start Mode**: the mode number of the lowest mode of a range.
   - **End Mode**: the mode number of the highest mode of a range.
   - **Alpha**: mass proportional damping, α_M.
   - **Beta**: stiffness proportional damping, β_M.

- Display the **Structural** tabbed page to define damping that is proportional to the internal forces but opposite in direction to the velocity. Do the following:
  1. Toggle on **Use structural damping data**.
  2. Enter the following in the data table:
     - **Start Mode**: the mode number of the lowest mode of a range.
     - **End Mode**: the mode number of the highest mode of a range.
     - **Damping Constant**: Damping factor, *s*.

For detailed information on how to enter data, see *Entering tabular data*.

4. If you selected **Frequencies** in Step 2, select one or more of the following options for defining damping:

- Display the **Direct modal** tabbed page to specify the fraction of critical damping, ξ, for a particular frequency range. Do the following:
  1. Toggle on **Use direct damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Critical Damping Fraction**: fraction of critical damping, ξ.

- Display the **Rayleigh** tabbed page to define Rayleigh damping, and do the following:
  1. Toggle on **Use Rayleigh damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Alpha**: mass proportional damping, α_M.
     - **Beta**: stiffness proportional damping, β_M.

- Display the **Structural** tabbed page to define damping that is proportional to the internal forces but opposite in direction to the velocity. Do the following:
  1. Toggle on **Use structural damping data**.
  2. Enter the following in the data table:
     - **Frequency**: frequency value in cycles/time.
     - **Damping Constant**: Damping factor, *s*.

For detailed information on how to enter data, see *Entering tabular data*.

5. If desired, repeat Steps 2–4 to create multiple damping definitions.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Configuring a subspace-based steady-state dynamic analysis

You can configure a subspace-based steady-state dynamic analysis to calculate the steady-state dynamic linearized response of a system to harmonic excitation. This type of procedure is based on direct solution of the steady-state dynamic equations projected onto a subspace of modes. You must first extract the modes using the eigenfrequency extraction procedure (described in *Configuring a frequency procedure*"). For more information, see *Subspace-Based Steady-State Dynamic Analysis*.

Create or edit a subspace-based steady-state dynamic procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Steady-state dynamics, Subspace`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, configure settings such as frequency range and matrix solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Choose one of the following options:

   - Choose **Compute real response only** if you want Abaqus/Standard to ignore damping terms. This option can reduce computational time.
   - Choose **Compute complex response** if you want to include damping terms and allow a complex system matrix to be factored.

4. Choose a **Scale** option to indicate whether you want the frequency range(s) of interest to be divided using a **Logarithmic** or **Linear** scale.
5. Toggle on **Include friction-induced damping effects** to include friction-induced contributions to the damping matrix. For more information, see *Contact Conditions with Sliding Friction*.
6. Toggle on **Use eigenfrequencies to subdivide each frequency range** if you want the frequency range(s) of interest to be subdivided using the system's eigenfrequencies. For more information, see *Selecting the Type of Frequency Interval for Which Output Is Requested*.
7. Click the arrow to the right of the **Projection** field, and select an option for controlling the frequency of the subspace projections:

   - Select **Evaluate at each frequency** to project the dynamic equations onto the subspace at each frequency you specify. This method is the most computationally expensive.
   - Select **Constant** to perform only one projection using model properties evaluated at the center frequency of all ranges and individual frequency points that you specify. This method is the least expensive. However, you should choose this method only when the material properties do not depend strongly on frequency.
   - Select **Interpolate at eigenfrequencies** to perform the projections at each extracted eigenfrequency in the requested frequency range and at eigenfrequencies immediately outside the range. The projected mass, stiffness, and damping matrices are then interpolated at each frequency point requested.

- Select **As a function of property changes** to select how often subspace projections onto the modal subspace are performed based on material property changes as a function of frequency. If you select this option, do the following:

  1. In the **Max. damping change** field, enter the maximum relative change in damping material properties before a new projection is to be performed.
  2. In the **Max. stiffness change** field, enter the maximum relative change in stiffness material properties before a new projection is to be performed.

- Select **Interpolate at lower and upper frequency limits** to perform projections at the lower and upper limits of the last frequency range. This method can be used only with the SIM architecture.

8. Enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time. If you enter zero, Abaqus/Standard assumes that results are required only at the frequency specified in the **Lower Frequency** column.

Number of Points

The number of points in the frequency range at which results should be given.

If you toggled on **Use eigenfrequencies to subdivide each frequency range**, this is the number of points at which results should be given, including the end points, from the lower limit of the frequency range to the first eigenfrequency in the range; in each interval from eigenfrequency to eigenfrequency; and from the highest eigenfrequency in the range to the high limit of the frequency range.

If you toggled off **Use eigenfrequencies to subdivide each frequency range**, this is the total number of points in the frequency range, including the end points.

Bias

The bias parameter. This parameter is useful only if you request results at 4 or more frequency points. It is used to bias the results points toward the ends of the intervals so that better resolution is obtained there. This option is recommended if you have toggled on **Use eigenfrequencies to subdivide each frequency range**, since the ends of each interval are the eigenfrequencies where the response amplitudes vary most rapidly.

For more information, see *The Bias Parameter*.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix solver** option:

- Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
- Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
- Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

Configuring a substructure generation procedure

You can configure a substructure generation procedure to control the data that are generated for a substructure.

You cannot have a substructure generation step in a model that contains model instances.

Create or edit a substructure generation procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Substructure generation`), or *Editing a step*.
2. On the **Basic**, **Options**, and **Damping** tabbed pages, configure settings such as substructure generation options and damping specifications.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. In the **Substructure identifier** field, enter an integer between 1 and 9999 that uniquely identifies this substructure in your model. Abaqus prepends a "Z" to the number you specify when it stores the substructure identifier in the model database.
4. If desired, toggle on **Evaluate recovery matrix for** to enable selective recovery in your analysis, and do one the following:

   - Select **Whole model** to enable selective recovery for the entire model.
   - Select **Region**, and select a node set or element set from the list to enable selective recovery for a single set in your model.

*Configure settings on the **Options** tabbed page*

1. In the **Edit Step** dialog box, display the **Options** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. From the **Generation Options**, do any of the following:

   - Toggle on **Compute gravity load vectors** to calculate the substructure's gravity load vectors during substructure generation.
   - Toggle on **Compute reduced mass matrix** to generate a reduced mass matrix for the substructure.
   - Toggle on **Compute reduced structural damping matrix** to generate a reduced structural damping matrix for the substructure.
   - Toggle on **Compute reduced viscous damping matrix** to generate a reduced viscous damping matrix for the substructure.
   - Toggle on **Evaluate frequency-dependent properties at frequency** to evaluate frequency-dependent material properties in substructure generation. If you toggle on this option, you can also specify the custom frequency for the frequency-dependent properties.

3. If you want to specify retained eigenmodes for generation of a coupled acoustic-structural substructure, toggle on **Specify retained eigenmodes by**, then select either **Mode range** or **Frequency range**.
4. If you are specifying retained eigenmodes by mode range, enter the following data in the **Data** table to generate the list of eigenmodes:

Start Mode

The first mode number.

End Mode

The end mode number.

Increment

The increment in mode numbers between the start and end modes.

For detailed information on how to enter data, see *Entering tabular data*.

5. If you are specifying retained eigenmodes by frequency range, enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Damping** tabbed page*

1. In the **Edit Step** dialog box, display the **Damping** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)

2. From the **Global Damping Ratios** options, specify the **Field** to which global damping will apply in this step:

   - Choose **None** to exclude the effects of global damping.
   - Choose **All** to apply global damping to all of the displacement, rotation, and acoustic fields in a model.
   - Choose **Acoustic** to apply global damping only to the acoustic fields in a model.
   - Choose **Mechanical** to apply global damping only to the displacement and rotation fields in a model.

3. If you selected any **Field** setting other than **None**, do any of the following:

   - For the **Alpha** option, specify a value for the first Rayleigh damping ratio.
   - For the **Beta** option, specify a value for the second Rayleigh damping ratio.
   - For the **Structural** option, specify a value for the structural damping ratio.

4. From the **Viscous damping** options, choose one of the following:

   - Choose **None** to exclude the effects of viscous damping altogether at the usage level.
   - Choose **Element** to activate only the generated condensed viscous damping matrix of the substructure.

- Choose **Factor** to activate only the Rayleigh viscous damping.
- Choose **Combined** to activate the combined generated and Rayleigh viscous damping matrix.

5. From the **Structural damping** options, choose one of the following:
   - Choose **None** to exclude the structural damping matrix.
   - Choose **Element** to activate only the generated condensed structural damping matrix of the substructure.
   - Choose **Factor** to activate only the stiffness proportional structural damping matrix.
   - Choose **Combined** to activate the combined generated and stiffness proportional.
6. When you have finished configuring settings for the substructure generation step, click **OK** to close the **Edit Step** dialog box.

Configuring a time-harmonic electromagnetic analysis

A time-harmonic electromagnetic analysis is valid in an electromagnetic model. In a time-harmonic electromagnetic (or eddy current) analysis, you specify one or more excitation frequencies, one or more frequency ranges, or a combination of excitation frequencies and ranges to obtain the time-harmonic solution directly at a given excitation frequency. For more information, see *Eddy Current Analysis*.

Create or edit a time-harmonic electromagnetic procedure

1. Display the **Edit Step** dialog box following the procedure outlined in *Creating a step* (**Procedure type:** `Linear perturbation; Electromagnetic, Time harmonic`), or *Editing a step*.
2. On the **Basic** and **Other** tabbed pages, configure settings such as frequency range and equation solver preferences as described in the following procedures.

*Configure settings on the **Basic** tabbed page*

1. In the **Edit Step** dialog box, display the **Basic** tabbed page.
2. In the **Description** field, enter a short description of the analysis step. Abaqus stores the text that you enter in the output database, and the text is displayed in the state block by the Visualization module.
3. Enter the following data in the **Data** table:

Lower Frequency

The lower limit of the frequency range or a single frequency, in cycles/time.

Upper Frequency

The upper limit of the frequency range, in cycles/time. If you enter zero, Abaqus/Standard assumes that results are required only at the frequency specified in the **Lower Frequency** column.

Number of Points

The number of points in the frequency range at which results should be given. The minimum value is 2. If the value is omitted, the default value of 20 points is assumed.

For detailed information on how to enter data, see *Entering tabular data*.

*Configure settings on the **Other** tabbed page*

1. In the **Edit Step** dialog box, display the **Other** tabbed page.

   (For information on displaying the **Edit Step** dialog box, see *Creating a step*, or *Editing a step*.)
2. Choose a **Matrix storage** option:
   - Choose **Use solver default** to allow Abaqus/Standard to decide whether a symmetric or unsymmetric matrix storage and solution scheme is needed.
   - Choose **Unsymmetric** to restrict Abaqus/Standard to the unsymmetric storage and solution scheme.
   - Choose **Symmetric** to restrict Abaqus/Standard to the symmetric storage and solution scheme.

   For more information on matrix storage, see *Matrix Storage and Solution Scheme in Abaqus/Standard*.

When you have finished configuring settings for the step, click **OK** to close the **Edit Step** dialog box.

Defining output requests

This section explains how to define and edit output requests.

In this section:

- *Creating an output request*
- *Modifying field output requests*
- *Modifying history output requests*

Creating an output request

When you create the first step in a sequence of steps, Abaqus/CAE generates default field and history output requests based on the analysis procedure that you selected for the step. The output requests are propagated to subsequent steps in the analysis. For more information, see *Propagation of output requests*. You can use the output request editors to edit the default output requests in the step in which they were created. You can also use the output request editors to create new output requests. A new output request overrides any propagated requests.

1. From the main menu bar, select **Output->Field Output Requests->Create** or **Output->History Output Requests->Create**.

   Tip: You can also create field and history output requests using the  and  tools, located in the module toolbox. For a diagram of the tools in the Step module toolbox, see *Using the Step module toolbox*.

   Abaqus/CAE displays the **Create Field Output** or **Create History Output** dialog box.

2. In the dialog box, do the following:
   a. Type a name for the output request or accept the default name.
   b. Click the arrow next to the **Step** text field, and select the step from the list that appears. Abaqus/CAE creates the output request during the selected step. After you create an output request, you can use the output requests manager to move it to an earlier or later step.
   c. Click **Continue**.

   Abaqus/CAE displays the **Edit Field Output Request** or the **Edit History Output Request** dialog box.

3. In the editor, enter the data necessary to define the output request. For more information on using the editor, see *Modifying field output requests*, or *Modifying history output requests*.
4. When you have finished configuring your output request, click **OK** to save your data and to exit the editor.

Additional information

- *Understanding output requests*
- *Defining output requests*

Modifying field output requests

You can use the field output editor to modify existing field output requests.
If you modify a field output request during a step into which it was propagated, you can modify only the output variables and the output frequency.

1. From the main menu bar, select **Output->Field Output Request->Manager**.

   Abaqus/CAE displays the **Field Output Requests Manager**. The manager indicates in which step the output request was created and into which steps it was propagated.

2. From the list of field output requests in the manager, select the step in which you want to modify the request.
3. From the buttons on the right side of the **Field Output Requests Manager**, click **Edit**.

   Abaqus/CAE displays the **Edit Field Output Request** editor. Information at the top of the editor indicates the following:

   - The name of the output request.
   - The name of the step in which you are editing the output request.
   - The analysis procedure associated with the step.

4. If you edit the output request in the step in which it was created, you can change the region from which variables will be output. From the top of the editor, click the arrow next to the **Domain** text field and select one of the following:

   - Select **Whole model** to request that Abaqus write field data to the output database for the entire model. Toggle on **Exterior only** to request output only on the exterior nodes and elements; this option is applicable only for three-dimensional models in an Abaqus/Standard or Abaqus/Explicit analysis.
   - Select **Set** to request that Abaqus write field data to the output database for only the named region that you specify. Click the arrow, and select the name from the list of sets that appears.
   - Select **Bolt load** to request that Abaqus write field data to the output database for only the bolt load that you specify. Click the arrow, and select the name from the list of bolt loads that appears.
   - Select **Composite layup** to request that Abaqus write field data to the output database for only the plies in the composite layup that you specify. Click the arrow, and select the name from the list of composite layups that appears.
   - Select **Fastener** to request that Abaqus write field data to the output database for only the fastener that you specify. Click the arrow, and select the name from the list of fasteners that appears.
   - Select **Substructure** to request that Abaqus write field data to the output database for only the substructure sets that you specify. Click  to open the **Select Substructure Sets** dialog box, then select the substructure sets for which you want to write field data. Select multiple sets by using the **[Ctrl]** or **[Shift]** keys while clicking on set names. You can use the name filter to show only substructure sets that match a specified pattern, such as belonging to a particular substructure part instance.
   - Select **Interaction** to request that Abaqus write field data to the output database for only the interaction that you specify. Click the arrow, and select the name from the list of surface-to-surface contact and self-contact interactions that appears.
   - Select **Skin** to request that Abaqus write field data to the output database for only the skin reinforcement that you specify. Click the arrow, and select the name from the list of skins that appears.

- Select **Stringer** to request that Abaqus write field data to the output database for only the stringer reinforcements that you specify. Click the arrow, and select the name from the list of stringers that appears.

5. Specify the desired output frequency:
   - Select **Last increment** to request field output for the last increment only. This output frequency is available only when you choose an Abaqus/Standard analysis procedure.
   - Select **Every n increments** to request that Abaqus write field data to the output database in increments. You can then specify the number of increments in the **n** field that appears. If you specify the frequency in increments, Abaqus also writes output after the last increment of the step. This output frequency is available when you choose an Abaqus/Standard analysis procedure.
   - Select **Every time increment** to request that Abaqus write field data to the output database at every time increment. This output frequency is available only when you choose an Abaqus/Explicit analysis procedure.
   - Select **Evenly spaced time intervals** to request that Abaqus write field data to the output database at a number of evenly spaced time intervals. You can then specify the number of intervals in the **Intervals** field that appears.
   - Select **Every x units of time** to request that Abaqus write field data to the output database every time a particular length of time elapses. You can then specify the length of time in the **x** field that appears.
   - Select **From time points** to request that Abaqus write field data to the output database according to a set of time points. You can then select a set of time points from the **Time Points** list that appears or click  to create a new set of time points. See *Defining time points*, for more information about creating a set of time points. This output frequency is available when you choose an Abaqus/Standard or Abaqus/Explicit analysis procedure.
6. If you requested output at **Evenly spaced time intervals**, **Every x units of time**, or **From time points**, you can also select **Output at exact times** from the **Timing** field to alter the time incrementation size to match the time intervals exactly.
7. Specify the desired element output position:
   - Select **Integration points** to request field output at the integration points where they are calculated (this is the default).
   - Select **Centroidal** to request field output at the centroid of each element.
   - Select **Nodes** to request field output at the node of each element.
   - Select **Averaged at nodes** to request field output extrapolated to the nodes of each element and then averaged with the values from surrounding elements that have the same type of properties.
8. In the **Output Variables** section of the editor, choose one of the following variable selection methods:

Select from list below

Choose this method to select the output variables of interest from the list of variable categories below. Use the following techniques to select particular variables:

- Click the arrow next to the desired variable category. From the list of variables that appears, select the variables of your choice.
- Toggle the desired variable category. This action selects or deselects all variables within that category.

The check box next to a variable category becomes completely filled when all variables within that category are selected. The box becomes half filled if only some of the variables within that category are selected.

Preselected defaults

Choose this method to allow Abaqus/CAE to select a preselected (default) set of output variables appropriate for the step's analysis procedure.

All

Choose this method to automatically select all of the allowable output variables within each variable category in the list.

Edit variables

Choose this method to enter or delete output variables in the text field located above the list of variable categories.



Note:

In addition to the current analysis procedure, other aspects of the model may affect the preselected default output variables. For example, if a selected output variable is valid for the analysis procedure but is not valid for the element type used in the mesh, Abaqus will remove that variable during the analysis.

9. If your domain is set to **Whole model**, **Set**, **Skin**, or **Stringer**, do the following:
   a. If your model contains rebar and you edit the output request in the step in which it was created, you can include output for rebar in the field data that Abaqus writes to the output database. From the bottom of the editor, toggle on **Output for rebar** and choose one of the following options that appears:

Include

Choose **Include** to request that Abaqus write output for rebar in addition to output for the underlying material to the output database.

Only

Choose **Only** to request that Abaqus write only output for rebar to the output database.

If you want to view rebar orientations in the Visualization module, you must toggle on **Output for rebar**.

   b. If you edit the output request in the step in which it was created, you can change the section points from which variables will be output. From the bottom of the editor, choose one of the following:

Use defaults

Choose **Use defaults** to request that Abaqus write field data to the output database from the default section points. Abaqus chooses the default section points based on the section selected in the Property module. (The default section points are usually the outer fibers of the section.) For more information see *The Property module*.

Specify

Choose **Specify** to enter the section points for which Abaqus will write field data to the output database. The specified section points are used only during the selected output request; Abaqus reverts to the default section points for subsequent output requests.

c. Toggle off **Include local coordinate directions when available** to reduce the size of the output database by excluding material orientations from the saved data.

10. If your domain is set to **Bolt load** or **Interaction**, toggle off **Include local coordinate directions when available** to reduce the size of the output database by excluding material orientations from the saved data.



Note:

If you exclude local coordinate directions from the output database, Abaqus/CAE displays all analysis results from the output database in the default coordinate system.

11. If your domain is set to **Composite layup**, specify the section points from which variables will be output. For more information, see *Requesting output from a composite layup*.



Note:

By default, Abaqus/CAE writes field output data from only the top and bottom of a composite layup, and no data from the plies are generated. Therefore, if your model contains a composite layup and you want data from individual plies, you must create a new output request or edit the default output request and specify the section points from which variables will be output.

From the bottom of the editor, choose one of the following:

Selected

Choose **Selected points for each ply** to request that Abaqus write field data to the output database from the top, middle, and/or bottom section point of each ply in the selected composite layup.

All

Choose **All section points in all plies** to request that Abaqus write field data to the output database from all of the section points of all of the plies in the selected composite layup.

Specify

Choose **Specify** to enter the section points for which Abaqus will write field data to the output database. Section points are numbered sequentially from the top of the first ply to the bottom of the last ply. The specified section points are used only during the selected output request; Abaqus reverts to the default section points for subsequent output requests.

12. If you edit the output request for an Abaqus/Explicit analysis procedure in the step in which it was created, you can apply a filter to remove high frequency data from the field output.

From the bottom of the editor, toggle on **Apply filter** and choose the default **Antialiasing** filter or select a named filter that was created using the Filter toolset. For more information, see *The Filter toolset*.

13. When you have finished defining the output request, click **OK** to save your changes.

Additional information

- *Understanding output requests*
- *Defining output requests*
- *The Property module*

Modifying history output requests

You can use the history output editor to modify existing history output requests.
If you modify a history output request during a step into which it was propagated, you can modify only the output variables and the output frequency.

1. From the main menu bar, select **Output->History Output Request->Manager**.

   Abaqus/CAE displays the **History Output Requests Manager**. The manager indicates in which step the output request was created and into which steps it was propagated.

2. From the list of history output requests in the manager, select the step in which you want to modify the request.
3. From the buttons on the right side of the **History Output Requests Manager**, click **Edit**.

   Abaqus/CAE displays the **Edit History Output Request** editor. Information at the top of the editor indicates the following:

   - The name of the output request.
   - The name of the step in which you are editing the output request.
   - The analysis procedure associated with the step.

4. If you edit the output request in the step in which it was created, you can change the region from which variables will be output. From the top of the editor, click the arrow next to the **Domain** text field and select one of the following:

   - Select **Whole model** to request that Abaqus write history data to the output database for the entire model.
   - Select **Set** to request that Abaqus write history data to the output database for only the named region that you specify. Click the arrow, and select the name from the list of sets that appears.
   - Select **Bolt load** to request that Abaqus write history data to the output database for only the bolt load that you specify. Click the arrow, and select the name from the list of bolt loads that appears.
   - Select **Composite layup** to request that Abaqus write history data to the output database for only the plies in the composite layup that you specify. Click the arrow, and select the name from the list of composite layups that appears.
   - Select **Contour integral** to request that Abaqus write history data to the output database for only the contour integral that you specify. Click the arrow, and select the name from the list of contour integrals that appears.

     When you request output from a contour integral, the **History Output Request** editor allows you to select only the frequency of output, the number of contour integrals, and the type of contour integral calculation. For more information, see *Requesting contour integral output*.
   - Select **Fastener** to request that Abaqus write history data to the output database for only the fastener that you specify. Click the arrow, and select the name from the list of fasteners that appears.
   - Select **General contact surface** to request that Abaqus write history data to the output database for only the general contact surface that you specify (this option is available only for Abaqus/Explicit analyses). Click the arrow, and select the surface from the list of surfaces that appears. History output will be written only for surfaces within the general contact domain.
   - Select **Integrated output section** to request that Abaqus write history data to the output database for only the integrated output section that you specify. Click the arrow, and select the name from the list of sections that appears. See *Defining integrated output sections*, for information on creating an integrated output section.

- Select **Interaction** to request that Abaqus write history data to the output database for only the interaction that you specify. Click the arrow, and select the name from the list of surface-to-surface contact and self-contact interactions that appears.
- Select **Skin** to request that Abaqus write history data to the output database for only the skin reinforcement that you specify. Click the arrow, and select the name from the list of skins that appears.
- Select **Spring/Dashpot** to request that Abaqus write history data to the output database for only the springs/dashpots that you specify. Click the arrow, and select the name from the list of springs/dashpots that appears.
- Select **Stringer** to request that Abaqus write history data to the output database for only the stringer reinforcements that you specify. Click the arrow, and select the name from the list of stringers that appears.

5. Specify the desired output frequency:
   - Select **Last increment** to request history output for the last increment only. This output frequency is available only when you choose an Abaqus/Standard analysis procedure.
   - Select **Every n increments** to request that Abaqus write history data to the output database in increments. You can then specify the number of increments in the **n** field that appears. If you specify the frequency in increments, Abaqus also writes output after the last increment of the step. This output frequency is available when you choose an Abaqus/Standard analysis procedure.
   - Select **Every n time increments** to request that Abaqus write history data to the output database in time increments. You can then specify the number of time increments in the **n** field that appears. This output frequency is available only when you choose an Abaqus/Explicit analysis procedure.
   - Select **Evenly spaced time intervals** to request that Abaqus write history data to the output database at a number of evenly spaced time intervals. You can then specify the number of intervals in the **Intervals** field that appears.
   - Select **Every x units of time** to request that Abaqus write history data to the output database every time a particular length of time elapses. You can then specify the length of time in the **x** field that appears.
   - Select **From time points** to request that Abaqus write history data to the output database according to a set of time points. You can then select a set of time points from the **Time Points** list that appears or click  to create a new set of time points. See *Defining time points*, for more information about creating a set of time points. This output frequency is available only when you choose an Abaqus/Standard analysis procedure.
6. If you selected an Abaqus/Standard analysis procedure and requested output at **Evenly spaced time intervals**, **Every x units of time**, or **From time points**, you can also select **Output at exact times** from the **Timing** field to alter the time incrementation size to match the time intervals exactly.
7. From the **Output Variables** field, choose the variables and variable components to output:

Select from list below

Choose this method to select the output variables of interest from the list of variable categories. Use the following techniques to select particular variables:

- In the top half of the editor, click the arrow next to the desired variable category. From the list of variables that appears, select the variables of your choice. If a variable of interest has components, click the arrow next to the variable and select the components of interest. To select or deselect all components of a variable, toggle the variable itself.

- Toggle the desired variable category to select or deselect all variables and variable components within that category.

The check box next to a variable category name becomes completely filled when all variables within that category are selected as well as all of the components of those variables. The box becomes half filled if only some of the variables or variable components within that category are selected. Likewise, the check box next to a variable name becomes completely filled when all components of that variable are selected. The box becomes half filled if only some of the components of that variable are selected.

Preselected defaults

Choose this method to allow Abaqus/CAE to select a preselected (default) set of output variables and components appropriate for the step's analysis procedure.

All

Choose this method to automatically select all of the allowable output variables and variable components within each variable category in the list.

Edit variables

Choose this method to enter or delete output variables and components in the text field located above the list of variable categories.



Note:

In addition to the current analysis procedure, other aspects of the model may affect the preselected default output variables. For example, if an output variable is valid for the analysis procedure but is not valid for the element type used in the mesh, Abaqus will remove that variable during the analysis.

8. If you want to include sensor output, toggle on **Include sensor when available**. This option is available when the domain **Set** is selected.
9. Toggle **Use global directions for vector-valued output**. This option is available when the domain **Set** is selected.

   When **Use global directions for vector-valued output** is on, nodal history output is requested in the global directions.

   When **Use global directions for vector-valued output** is off, nodal history output is requested in the local directions defined by nodal transformations.
10. If your domain is set to **Whole model**, **Set**, **Skin**, or **Stringer** do the following:
    a. If your model contains rebar and you edit the output request in the step in which it was created, you can include output for rebar in the field data that Abaqus writes to the output database. From the bottom of the editor, toggle on **Output for rebar** and choose one of the following options that appears:

Include

Choose **Include** to request that Abaqus write output for rebar in addition to output for the underlying material to the output database.

Only

Choose **Only** to request that Abaqus write only output for rebar to the output database.

If you want to view rebar orientations in the Visualization module, you must toggle on **Output for rebar**.

b. If you edit the output request in the step in which it was created, you can change the section points from which variables will be output. From the bottom of the editor, choose one of the following:

Use defaults

Choose **Use defaults** to request that Abaqus write field data to the output database from the default section points. Abaqus chooses the default section points based on the section selected in the Property module. (The default section points are usually the outer fibers of the section.) For more information see *The Property module*.

Specify

Choose **Specify** to enter the section points for which Abaqus will write field data to the output database. The specified section points are used only during the selected output request; Abaqus reverts to the default section points for subsequent output requests.

11. If your domain is set to **Composite layup**, specify the section points from which variables will be output.



Note:

By default, Abaqus/CAE writes field output data from only the top and bottom of a composite layup, and no data from the plies are generated. Therefore, if your model contains a composite layup and you want data from individual plies, you must create a new output request or edit the default output request and specify the section points from which variables will be output.

From the bottom of the editor, choose one of the following:

Selected

Choose **Selected points for each ply** to request that Abaqus write field data to the output database from the top, middle, and/or bottom section point of each ply.

All

Choose **All section points in all plies** to request that Abaqus write field data to the output database from all of the section points of all of the plies.

Specify

Choose **Specify** to enter the section points for which Abaqus will write field data to the output database. Section points are numbered sequentially from the top of the first ply to the bottom of the last ply. For example, if you have three plies, each with three section points, and you want output from the middle section point of each ply, you would enter `2,5,8`. The specified section points are used only during the selected output request; Abaqus reverts to the default section points for subsequent output requests.

12. If you edit the output request for an Abaqus/Explicit analysis procedure in the step in which it was created, you can apply a filter to remove high frequency data from the history output.

    From the bottom of the editor, toggle on **Apply filter** and choose the default **Antialiasing** filter or select a named filter that was created using the Filter toolset. For more information, see *The Filter toolset*.

13. When you have finished defining the output request, click **OK** to save your changes.

Additional information

- *Understanding output requests*
- *Defining output requests*
- *The Property module*

Requesting specialized output

This section describes how you request and configure specialized output.

In this section:

- *Defining integrated output sections*
- *Configuring restart output requests*
- *Configuring diagnostic printing*
- *Configuring monitor requests*
- *Defining time points*

Defining integrated output sections

Select **Output->Integrated Output Sections** from the main menu bar to create an integrated output section. For more information, see *Integrated Output Section Definition*.

1. From the main menu bar, select **Output->Integrated Output Sections->Create**.

   The **Create Integrated Output Section** dialog box appears.
2. In the dialog box, type a name for the output section or accept the default name.
3. Select the surface on which to define the integrated output section using either of the following methods:

   - Select a surface in the viewport. When you have finished selecting, click mouse button 2.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** to define the section on geometry.
     - Click **Mesh** to define the section on native or orphan mesh faces.

   - To select from a list of existing surfaces, do the following:

     1. Click **Surfaces** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available surfaces.
     2. Select the surface of interest, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   The integrated output section editor appears. The surface on which you are defining the section is highlighted in the viewport.
4. By default, the integrated output section is anchored at the global origin. To define a reference point at which the section will be anchored, select **Anchor at reference point**, and click **Edit** .
5. If the reference point is not physically connected to the model, you can adjust its initial position and specify that it should follow the average motion of the surface associated with the integrated output section. By default, the reference point does not track the average motion of the surface.

   - To reposition the reference point at the center of the surface in the initial configuration, toggle on **Move point to center of surface**.
   - To specify that the reference point should both translate and rotate with the average motion of the surface, select **Average translation and rotation** for the **Point motion**.
   - To specify that the reference point should translate with the average translation of the surface without any rotation, select **Average translation** for the **Point motion**.
6. By default, the coordinate system for the integrated output section is initialized to the global coordinate system. To choose a local coordinate system instead, click  next to the **CSYS** field.
7. If desired, toggle on **Project orientation onto surface** to project the coordinate system (either global or local) onto the section surface.

8. Click **OK** to close the integrated output section editor.

Additional information

• *Integrated output requests*

Configuring restart output requests

Select **Output->Restart Requests** from the main menu bar to configure the information written to the restart files. For more information, see *Restarting an analysis* and *Restarting an Analysis*.

1. From the main menu bar, select **Output->Restart Requests**.

   The **Edit Restart Requests** dialog box appears with a list of the steps in the current model.

2. From the dialog box, select the step and specify the frequency at which Abaqus writes restart information. Do the following:

Abaqus/Standard

For an Abaqus/Standard analysis step, enter the frequency in increments or in time intervals:

- To write restart information after every regular number of increments, enter that number in the **Frequency** column.
- To write restart information at a specific number of equally spaced time intervals during a step, enter that number in the **Intervals** column.

  Restart information is written at the increment ending immediately after the time specified by the time interval. In an Abaqus/Standard analysis, you can toggle on the **Time Marks** column to write restart information at the exact time specified by the time interval.

By default, no restart information is written for an Abaqus/Standard analysis step (i.e., the frequency is set to zero).

Abaqus/Explicit

For an Abaqus/Explicit analysis step, enter the number of equally spaced time **Intervals** at which restart information is written. By default, Abaqus/Explicit writes restart information only at the beginning and end of each step (i.e., **Intervals** is set to one). You cannot avoid writing data to the restart files for an Abaqus/Explicit analysis step; the number of **Intervals** must be set to one or greater.

By default, Abaqus/Explicit writes restart information at the increment ending immediately after the time specified by the time interval. Toggle on the **Time Marks** column to write restart information at the exact time specified by the time interval.

3. For an Abaqus/Standard or Abaqus/Explicit analysis, you can request that information from only one increment per step or from every specified increment be retained.

   - Toggle on the **Overlay** column to request that only information from one increment per step be retained in the restart files, thus minimizing the size of the files. When a step completes, the last increment is retained.
   - Toggle off the **Overlay** column to request that information from every specified increment or time interval be retained in the restart files.

   You can use the **Overlay** column only when the frequency is greater than zero.

4. Repeat Steps 2 and 3 for each step of the analysis.

5. When you have finished configuring your restart output requests, click **OK** to close the **Edit Restart Requests** dialog box.

Additional information

• *Restart output requests*

Configuring diagnostic printing

Select **Output->Diagnostic Print** to configure the diagnostic messages that Abaqus writes to the message file (for Abaqus/Standard) or to the status file (for Abaqus/Explicit) and reports to the Job module during an analysis. If the analysis of your model fails or produces unexpected results, you can examine these messages and look at a description of the analysis. (For more information, see *Diagnostic printing*.)

Abaqus/Standard also writes most diagnostic information to the output database for use in the Visualization module. However, the diagnostic information written to the output database is not affected by the diagnostic print settings. (For more information, see *Viewing diagnostic output*.)

1. From the main menu bar, select **Output->Diagnostic Print**.

   The **Edit Diagnostic Print** dialog box appears with a list of all the steps in the current model.
2. In the **Frequency** column, specify the frequency at which you want diagnostic printing data written for each step. (This option is not available for Abaqus/Explicit analyses.)
3. Click in the other columns to request specific diagnostic information during a particular step.

   A check mark appears indicating that type of diagnostic printing has been enabled during the step. Click in the column again to turn off diagnostic printing.
4. When you have finished configuring your diagnostic printing, click **OK** to exit the **Edit Diagnostic Print** dialog box.

Additional information

- *Diagnostic printing*
- *Requesting specialized output*
- *Viewing diagnostic output*

Configuring monitor requests

Use the **DOF Monitor** dialog box to monitor a degree of freedom at a point during the course of an analysis. Abaqus reports the degree of freedom values for that point to the Job module, to the status file (`jobname.sta`), and, for Abaqus/Standard analyses, to the message file (`jobname.msg`). For more information, see *Degree of freedom monitor requests*.

1. From the main menu bar, select **Output->DOF Monitor**.

   The **DOF Monitor** dialog box appears.

2. In the dialog box, toggle **Monitor a degree of freedom throughout the analysis**.

   If **Monitor a degree of freedom throughout the analysis** is toggled on, the monitoring options become available in the dialog box.

3. Click  next to the **Region** field, and use one of the following methods to select the point to monitor:

   - Use an existing set to define the point. On the right side of the prompt area, click **Points**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**. (Abaqus/CAE displays an error message when you click **OK** in the **DOF Monitor** dialog box if you select a set containing more than one vertex or node.)

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Points** on the right side of the prompt area.

   - Use the mouse to select a point in the viewport. (For more information, see *Selecting objects within the current viewport*.)

   The point that you select becomes highlighted in red in the viewport.

4. In the **Degree of freedom** field, enter the degree of freedom that you want to monitor.
5. If you are performing an Abaqus/Standard analysis, in the **Print to the message file every *n* increments** field, enter how often you want Abaqus to write the degree of freedom values to the message file.

   (There is no option to control the frequency at which values are written to the status file, since this frequency is determined by Abaqus. For more information, see *About Output*.)

6. Click **OK** to save your request and to exit the dialog box.

Additional information

- *Degree of freedom monitor requests*
- *Requesting specialized output*
- *Monitoring the progress of an analysis job*

Defining time points

You can define named sets of time points that describe all of the moments during an analysis when you want to request that Abaqus write field output or history output data. By requesting output at specified time points, you can create an irregular sampling that focuses on gathering data during the critical points in your analysis.

The **Create Time Points** dialog box provides two methods for defining sets of time points. You can specify each time point by entering every value individually on its own line in the table portion of the dialog box. Alternatively, you can select **Specify using delimiters** to define several ranges of time with a different increment that defines the sampling rate in each range. For example, you can create a time point every 0.1 seconds for the time range between 0 and 1, then change the increment to 0.05 for the time range between 1 and 2. The individual and time range definition methods are mutually exclusive; you cannot, for example, add individual time points to a series of time points specified by the time ranges method.



Note:

You can also import a set of time points either individually or using time ranges into a model database from an ASCII file. Once the data are imported, you can edit those time points.

The **Create Time Points** dialog box presents a set of time points data in a tabular format. You can modify these sets by adding, removing, and clearing rows of data or by editing individual data points or time ranges. Once you create a set of time points, you can edit its contents or copy, rename, or delete the set. If you rename or delete a set of time points for which you have requested field or history output, Abaqus/CAE issues an error message.

1. Open the **Create Time Points** dialog box.

   From the main menu bar, select **Output->Time Points->Create**.

2. Enter a name for this set of time points, and click **OK**.

   The **Edit Time Points** dialog box appears.

3. To read time point data from an ASCII file, perform the following steps:

   1. Click mouse button 3 in the table portion of the dialog box; then choose **Read from File** from the list that appears.

      The **Read Data from ASCII File** dialog box appears.

   2. Click , and use the **ASCII File Selection** dialog box to navigate to the file you want to import. Click **OK** when you have the file selected.
   3. If desired, you can modify the starting location in the time points table for the data import. By default, the starting row and column values for the import are both equal to 1; modify either value if you want to customize the imported data with additional rows and columns of values.
   4. Click **OK** to import the values into the time points table.

4. Select one of the two time point definition methods:

   - To define time points individually, enter a time point in row 1, and then add a new row and time point for every additional time point in the set. Individually entered time points must be entered in sequential order.
   - To specify time points using delimiters, select **Specify using delimiters**, and then enter the starting time, ending time, and increment for each series of time points that you want to create. See *Entering tabular data*, for more information about tabular data entry.

5. Click **OK** to save your set of time points.

Additional information

- *Defining output requests*
- *Requesting specialized output*

Customizing ALE adaptive meshing

This section explains how to configure ALE adaptive meshing in particular analysis steps.

In this section:

- *Defining an ALE adaptive mesh region*
- *Specifying ALE adaptive mesh constraints*
- *Specifying controls for ALE adaptive meshing*

Defining an ALE adaptive mesh region

You can define regions of the model where you want ALE adaptive meshing and you can specify the frequency and intensity of adaptive meshing for that domain. For more information, see the following sections:

- *About ALE Adaptive Meshing*
- *Defining ALE Adaptive Mesh Domains in Abaqus/Explicit*
- *Defining ALE Adaptive Mesh Domains in Abaqus/Standard*



Note:

You can define only one adaptive mesh domain for any particular step.

1. From the main menu bar, select **Other->ALE Adaptive Mesh Domain->Manager**.

   Abaqus/CAE displays the **ALE Adaptive Mesh Domain Manager** showing the steps defined in the model and the adaptive meshing controls associated with each step.

2. In the **ALE Adaptive Mesh Domain Manager**, select the step during which you will configure the adaptive meshing, and click **Edit**.

   Abaqus/CAE displays the ALE adaptive mesh domain editor.

3. At the top of the editor, choose **Use the ALE adaptive mesh domain below**.

4. Click  next to the **Region** field, and use one of the following methods to select the region you want to mesh:

   - Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.
   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate you have finished selecting.

   Abaqus/CAE highlights the selected regions in red in the viewport.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. If you want to specify nondefault adaptive mesh controls, toggle on **ALE Adaptive mesh controls**, and select the name of the adaptive mesh controls to associate with the region.

   If you have not yet created mesh controls, you can click  to the right of the **ALE Adaptive mesh controls** field to define the controls that you want to assign to the adaptive mesh domain. See *Specifying controls for ALE adaptive meshing*.

6. Enter the **Frequency**, in increments, at which Abaqus will mesh the region.
7. Enter the number of **Remeshing sweeps per increment**. An increment in this case is each adaptive mesh increment, as defined by the **Frequency** setting above.
8. For adaptive meshing in Abaqus/Explicit steps, type the number of **Initial remeshing sweeps** that Abaqus/Explicit will apply at the beginning of the step.
9. Click **OK** to save the settings and to close the ALE adaptive mesh domain editor.

The **ALE Adaptive Mesh Domain Manager** displays the adaptive mesh controls associated with the step.

10. Click **Dismiss** to close the **ALE Adaptive Mesh Domain Manager**.

Additional information

- *ALE Adaptive Meshing*
- *The Set and Surface toolsets*

Specifying ALE adaptive mesh constraints

You can define ALE adaptive mesh constraints on the motion of the mesh for an adaptive mesh domain. The ALE adaptive mesh constraint can either prescribe independent mesh motion for nodes in an adaptive mesh domain or specify nodes that must follow the material. For more information, see the following sections:

- *About ALE Adaptive Meshing*
- *Defining ALE Adaptive Mesh Domains in Abaqus/Explicit*
- *Defining ALE Adaptive Mesh Domains in Abaqus/Standard*

1. From the main menu bar, select **Other->ALE Adaptive Mesh Constraint->Create**.

   Abaqus/CAE displays the ALE adaptive mesh constraint editor.
2. Enter a name for the ALE adaptive mesh constraint.
3. Select the step in which you want to create the constraint.
4. Choose the type of mesh motion, **Displacement/Rotation** or **Velocity/Angular velocity**, and click **Continue**.
5. Use one of the following methods to select the region that you want to constrain:

   - Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.
   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate you have finished making selections.

   Abaqus/CAE highlights the selected regions in red in the viewport.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit ALE Adaptive Mesh Constraint** dialog box appears.
6. If you are editing an existing mesh constraint you can edit the region to which the constraint is applied by clicking  in the top part of the editor. Use the region selection techniques described in Step 5.
7. If you want to change the coordinate system (**CSYS**) in which to apply the constraint, click  and use one of the following methods:

   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.

     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.

   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the constraint is created.
8. Click the arrow to the right of the **Motion** field, and specify whether you want to prescribe mesh motions that are independent of the underlying material or define nodes that must follow the underlying material.

If you are performing an Abaqus/Standard analysis, you can also choose to define mesh motion in user subroutine *UMESHMOTION*.

9. If you selected **Independent of underlying material** in Step 8, toggle on the degrees of freedom that you want to constrain. The text field becomes available in which you can specify a value for the degree of freedom. Toggle off a degree of freedom to leave the degree of freedom unconstrained.
10. If you selected **Independent of underlying material** in Step 8, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click **Create** to create a new amplitude. (See *The Amplitude toolset* for more information.)
11. Click **OK** to save the named constraint and to close the editor.

Additional information

• *ALE Adaptive Meshing*

Specifying controls for ALE adaptive meshing

You can specify controls for the adaptive meshing and advection algorithms applied to an adaptive mesh domain. For more information, see *ALE Adaptive Meshing and Remapping in Abaqus/Explicit* and *ALE Adaptive Meshing and Remapping in Abaqus/Standard*.

In this section:

- *Specifying ALE adaptive mesh controls for an Abaqus/Standard analysis*
- *Specifying ALE adaptive mesh controls for an Abaqus/Explicit analysis*

Specifying ALE adaptive mesh controls for an Abaqus/Standard analysis

You use the **Edit ALE Adaptive Mesh Controls** dialog box to customize various aspects of the adaptive meshing and advection algorithms applied to an adaptive mesh domain. For more information, see *ALE Adaptive Meshing and Remapping in Abaqus/Standard*.

1. From the main menu bar, select **Other->ALE Adaptive Mesh Controls->Create**.

   A **Create ALE Adaptive Mesh Controls** dialog box appears.

2. In the dialog box, type a name for the ALE adaptive mesh controls, and click **Continue**.

   The **Edit ALE Adaptive Mesh Controls** dialog box appears.

3. Click the arrow to the right of the **Smoothing algorithm** field, and select a method for calculating adaptive mesh smoothing:

   - Select **Determined by analysis product** to accept the default for the analysis product you are using (in this case, Abaqus/Standard).
   - Select **Enhanced algorithm based on evolving geometry** to use geometrically enhanced forms of the basic smoothing algorithms as a technique to mitigate distortion. These forms are heuristic and based on nodal locations only. Due to their heuristic nature, geometric enhancements may not always improve the mesh smoothing.
   - Select **Conventional smoothing** to use the conventional forms of the smoothing algorithms.

   For more information on adaptive mesh smoothing, see *The ALE Adaptive Mesh Sweeping Algorithm*.

4. Calculation of the new mesh in Abaqus/Standard is based on some combination of two basic smoothing methods: original configuration projection smoothing and volume smoothing. You must specify the weighting factor for each method in the **Volumetric** and **Original configuration projection** fields. For more information, see *Combining Smoothing Methods*.

5. In the **Initial feature angle** field, enter the initial geometric feature angle, θ_I, in degrees ($0° \le \theta_I \le 180°$). This angle is used to detect geometric edges and corners. The default value is $\theta_I = 30°$. Setting $\theta_I = 180°$ will ensure that no geometric edges or corners are detected or enforced.

   For more information, see *Controlling the Detection of Geometric Edges and Corners*.

6. In the **Transition feature angle** field, enter the transition geometric feature angle, θ_T, in degrees ($0° \le \theta_T \le 180°$). This angle is used to determine when geometric edges and corners should be deactivated to allow remeshing across them. The default value is $\theta_T = 30°$. Setting $\theta_T = 0°$ will ensure that no geometric edges or corners are deactivated.

   For more information, see *Controlling the Activation and Deactivation of Geometric Edges and Corners*.

7. Click **OK** to save the named set of controls and to close the editor.

Specifying ALE adaptive mesh controls for an Abaqus/Explicit analysis

You use the **Edit ALE Adaptive Mesh Controls** dialog box to customize various aspects of the adaptive meshing and advection algorithms applied to an adaptive mesh domain. For more information, see *ALE Adaptive Meshing and Remapping in Abaqus/Explicit*.

1. From the main menu bar, select **Other->ALE Adaptive Mesh Controls->Create**.

   A **Create ALE Adaptive Mesh Controls** dialog box appears.

2. In the dialog box, type a name for the ALE adaptive mesh controls, and click **Continue**.

   The **Edit ALE Adaptive Mesh Controls** dialog box appears.

3. Choose a mesh smoothing **Priority**:

   - Choose **Improve aspect ratio** to perform adaptive meshing that minimizes element distortion and improves element aspect ratios at the expense of diffusing initial mesh gradation. This objective is recommended for problems with moderate to large overall deformation. For more information, see *Specifying a Uniform Mesh Smoothing Objective*.
   - Choose **Preserve initial mesh grading** to perform adaptive meshing that attempts to preserve initial mesh gradation while reducing distortions as the analysis evolves. This objective is recommended only for adaptive mesh domains with reasonably structured graded meshes undergoing low to moderate overall deformation. For more information, see *Specifying a Graded Mesh Smoothing Objective*.

4. Click the arrow to the right of the **Smoothing algorithm** field, and select a method for calculating adaptive mesh smoothing:

   - Select **Determined by analysis product** to accept the default for the analysis product you are using (in this case, Abaqus/Explicit).
   - Select **Enhanced algorithm based on evolving geometry** to use geometrically enhanced forms of the basic smoothing algorithms as a technique to mitigate distortion.
   - Select **Conventional smoothing** to use the conventional forms of the smoothing algorithms.

   For more information on adaptive mesh smoothing, see *Mesh Smoothing Methods*.

5. Choose a **Meshing predictor** option:

   - Choose **Current deformed position** to perform adaptive meshing based on current nodal positions. This method is recommended for all Langrangian-like problems and for problems with very large distortions. For more information, see *Positioning Nodes in Lagrangian Domains*.
   - Choose **Position from previous ALE adaptive mesh increment** to perform adaptive meshing based on the positions of the nodes at the end of the previous adaptive mesh increment. This technique is recommended for Eulerian-like problems where material flow is significant compared to the overall deformation. For more information, see *Positioning Nodes in Eulerian Domains*.

6. In the **Curvature refinement** field, enter a value for the curvature refinement weight, α_C. An appropriate value allows you to ensure that there is sufficient mesh refinement near highly curved boundaries. The default, $\alpha_C = 1$, works well on a wide variety of problems. For more information, see *Solution-Dependent Meshing Based on Concave Boundary Curvature*.
7. Calculation of the new mesh in Abaqus/Explicit is based on some combination of three basic smoothing methods: volume smoothing, Laplacian smoothing, and equipotential smoothing. You must specify the weighting factor for each method in the **Volumetric**, **Laplacian**, and **Equipotential** fields. For more information, see *Mesh Smoothing Methods*.

8. In the **Initial feature angle** field, enter the initial geometric feature angle, θ_I, in degrees ($0° \le \theta_I \le 180°$). This angle is used to detect geometric edges and corners. The default value is $\theta_I = 30°$. Setting $\theta_I = 180°$ will ensure that no geometric edges or corners are detected or enforced.
9. In the **Transition feature angle** field, enter the transition geometric feature angle, θ_T, in degrees ($0° \le \theta_T \le 180°$). This angle is used to determine when geometric edges and corners should be deactivated to allow remeshing across them. The default value is $\theta_T = 30°$. Setting $\theta_T = 0°$ will ensure that no geometric edges or corners are deactivated.
10. In the **Mesh constraint angle field**, enter the mesh constraint angle, θ_C, in degrees ($5° \le \theta_C \le 85°$). The default value is $\theta_C = 60°$.

    When adaptive mesh constraints are applied to nodes on Lagrangian or sliding boundary regions, the analysis will terminate if the angle between the normal to the boundary region and the direction of the prescribed constraint becomes less than θ_C. When adaptive mesh constraints are applied to nodes that are part of a Lagrangian or active geometric edge, the analysis will terminate if the angle between the prescribed constraint and the plane perpendicular to the edge becomes less than θ_C.
11. Choose an algorithm for remapping solution variables after adaptive meshing has been performed:
    - Choose **First order** to use a first order method based on donor cell differencing.
    - Choose **Second order** to use a second order method based on the work of Van Leer.

    For more information, see *Advection Methods for Element Variables*.
12. Choose a method for advecting momentum:
    - Choose **Element center projection** for the least expensive method.
    - Choose **Half-index shift** for a method that is more expensive computationally but which may demonstrate better dispersion properties.

    For more information, see *Momentum Advection*.
13. Click **OK** to save the named set of controls and to close the editor.

Customizing the Abaqus analysis controls

This section explains how to customize the Abaqus analysis controls.

In this section:

- *Customizing general solution controls*
- *Customizing solver controls*

Customizing general solution controls

The general solution controls editor allows you to modify the algorithms used for convergence and for time integration accuracy.
For more information, see *Analysis Convergence Controls*.



Note:

This option is available only for general Abaqus/Standard analysis steps.

1. From the main menu bar, select **Other->General Solution Controls->Manager**.
   Abaqus/CAE displays the **General Solution Controls Manager** showing the steps defined in the model and the general solution controls associated with each step.
2. In the **General Solution Controls Manager**, select the analysis step of interest, and click **Edit**.
   The **General Solution Controls Editor** appears.
3. Choose an option for specifying solution controls:
   - Choose **Propagate from previous step** if you want the solution controls settings from the previous step to remain active in the current step.
   - Choose **Reset all parameters to their system-defined defaults** if you want all of the parameters on all of the tabbed pages in the editor to revert to their default values.
   - Choose **Specify** if you want to enter new values for particular parameters for this step.
4. If desired, display the **Field Equations** tabbed page to define tolerances for field equations:
   a. If you want to select a particular type of equation for which the solution control parameters are being defined, choose **Specify individual fields**. Otherwise, choose **Apply to all applicable fields** to apply the values that you enter to all fields.
   b. Enter new values for one or more of the following parameters:
      - R_n^α, convergence criterion for the ratio of the largest residual to the corresponding average flux norm for convergence. Default $R_n^\alpha = 5 \times 10^{-3}$.
      - C_n^α, convergence criterion for the ratio of the largest solution correction to the largest corresponding incremental solution value. Default $C_n^\alpha = 10^{-2}$.
      - $\tilde{q}_0^\alpha$, initial value of the time average flux for this step. The default is the time average flux from previous steps or 10^{-2} if this is Step 1.
      - $\tilde{q}_u^\alpha$, user-defined average flux. When this value is defined, $\tilde{q}^\alpha(t) = \tilde{q}_u^\alpha$ for all t.

      The remaining items rarely need to be reset from their default values:

      - R_P^α, alternative residual convergence criterion to be used after I_P^α iterations. Default $R_P^\alpha = 2 \times 10^{-2}$.
      - ϵ^α, criterion for zero flux compared to $\tilde{q}^\alpha$. Default $\epsilon^\alpha = 10^{-5}$.
      - C_ϵ^α, convergence criterion for the ratio of the largest solution correction to the largest corresponding incremental solution value when there is zero flux in the model. Default $C_\epsilon^\alpha = 10^{-3}$.

- R_l^α, convergence criterion for the ratio of the largest residual to the corresponding average flux norm for convergence to be accepted in one iteration (that is, for a linear case). Default $R_l^\alpha = 10^{-8}$.
- C_f, field conversion ratio used in scaling the relationship between two active fields when one is of negligible magnitude. Default $C_f = 1.0$.
- ϵ_l^α, criterion for zero flux compared to the time averaged value of the largest flux $\tilde{q}_{\max}^\alpha$ in the model during the current step. Default $\epsilon_l^\alpha = 10^{-5}$.
- ϵ_d^α, criterion for zero displacement increment (and/or zero penetration if the **Convert severe discontinuity iterations** setting for the step is **On**) compared to the characteristic element length in the model. This item is used only for the displacement field and warping degree of freedom equilibrium equations. Default $\epsilon_d^\alpha = 10^{-8}$.

At any time you can click **System-defined Defaults for this Page** to return all parameters on this tabbed page to their system-defined default values. For more information, see *Defining Tolerances for Field Equations*.

5. If desired, display the **Time Incrementation** tabbed page to set time incrementation control parameters:
   a. Toggle on **Discontinuous analysis** to set $I_0 = 8$ and $I_R = 10$. These settings can help avoid premature cutbacks of the time increment.
   b. Enter new values for one or more of the following parameters:
      - I_0, number of equilibrium iterations (without severe discontinuities) after which the check is made whether the residuals are increasing in two consecutive iterations. Minimum value is $I_0 = 3$. Default $I_0 = 4$. If you toggled on **Discontinuous analysis**, $I_0 = 8$ and cannot be changed.
      - I_R, number of consecutive equilibrium iterations (without severe discontinuities) at which logarithmic rate of convergence check begins. Default $I_R = 8$. If you toggled on **Discontinuous analysis**, $I_R = 10$ and cannot be changed. The logarithmic rate of convergence is not checked if fixed time incrementation is used.

      The remaining items rarely need to be reset from their default values:

      - I_P, number of consecutive equilibrium iterations (without severe discontinuities) after which the residual tolerance R_p is used instead of R_n. Default $I_P = 9$.
      - I_C, upper limit on the number of consecutive equilibrium iterations (without severe discontinuities), based on prediction of the logarithmic rate of convergence. Default $I_C = 16$.
      - I_L, number of consecutive equilibrium iterations (without severe discontinuities) above which the size of the next increment will be reduced. Default $I_L = 10$.
      - I_G, maximum number of consecutive equilibrium iterations (without severe discontinuities) allowed in consecutive increments for the time increment to be increased. Default $I_G = 4$.

- I_S, maximum number of severe discontinuity iterations allowed in an increment if the **Convert severe discontinuity iterations** setting for the step is **Off**. Default $I_S = 12$. This parameter is not used if the **Convert severe discontinuity iterations** setting for the step is **On**.
- I_A, maximum number of cutbacks allowed for an increment. Default $I_A = 5$.
- I_J, maximum number of severe discontinuity iterations allowed in two consecutive increments for the time increment to be increased if the **Convert severe discontinuity iterations** setting for the step is **Off**. Default $I_J = 5$. This parameter is not used if the **Convert severe discontinuity iterations** setting for the step is **On**.
- I_T, minimum number of consecutive increments in which the time integration accuracy measure must be satisfied without any cutbacks to allow a time increment increase. Default $I_T = 3$. Maximum allowed $I_T = 10$.
- I_S^c, maximum number of severe discontinuity iterations allowed in an increment if the **Convert severe discontinuity iterations** setting for the step is **On**. Default $I_S^c = 50$. This parameter serves only as a protection against failure of the default convergence criteria and should rarely need to be changed. This parameter is not used if the **Convert severe discontinuity iterations** setting for the step is **Off**.
- D_f, cutback factor used when the solution appears to be diverging. Default $D_f = 0.25$.
- D_C, cutback factor used when the logarithmic rate of convergence predicts that too many equilibrium iterations will be needed. Default $D_C = 0.5$.
- D_B, cutback factor for the next increment when too many equilibrium iterations (I_L) are used in the current increment. Default $D_B = 0.75$.
- D_A, cutback factor used when the time integration accuracy tolerance is exceeded. Default $D_A = 0.85$.
- D_S, cutback factor used when too many iterations (I_S) arise because of severe discontinuities. Default $D_S = 0.25$.
- D_H, cutback factor used when element calculations have problems such as excessive distortion in large-displacement problems. Default $D_H = 0.25$.
- D_D, increase factor when two consecutive increments converge in a small number of equilibrium iterations (I_G). Default $D_D = 1.5$.
- W_G, ratio of average time integration accuracy measure over I_T increments to the corresponding tolerance for the next allowable time increment to be increased. Default $W_G = 0.75$.
- D_G, increase factor for the next time increment, as a ratio of the average integration accuracy measure over I_T increments to the corresponding tolerance, when the time integration accuracy measure is less than W_G of the tolerance during I_T consecutive increments. Default $D_G = 0.8$.
- D_M, maximum time increment increase factor for all cases except dynamic stress analysis and diffusion-dominated processes. Default $D_M = 1.5$.

- D_M, dyn, maximum time increment increase factor for dynamic stress analysis. Default $D_M = 1.25$.
- D_M, diff, maximum time increment increase factor for diffusion-dominated processes (creep, transient heat transfer, soils consolidation, transient mass diffusion). Default $D_M = 2.0$.
- D_L, minimum ratio of proposed next time increment to D_M times the current time increment for the proposed time increment to be used in a linear transient problem. This parameter is intended to avoid excessive decomposition of the system matrix and should be less than 1.0. Default $D_L = 0.95$.
- D_E, minimum ratio of proposed next time increment to the last successful time increment for extrapolation of the solution vector to take place. Default $D_E = 0.1$.
- D_R, maximum allowable ratio of time increment to stability limit for conditionally stable time integration procedures. Default is 1.0.
- D_F, fraction of stability limit used as current time increment when the time increment exceeds the above factor times the stability limit. This value cannot exceed 1.0. Default 0.95.
- D_T, increase factor for the time increment directly before a time point or end time of a step is reached. This parameter is used to avoid the small time increment that is sometimes necessary to hit a time point or to complete a step and must be greater than or equal to 1.0. If output or restart data are requested at exact times in a step, the default $D_T = 1.25$; otherwise, the default $D_T = 1.0$.

At any time you can click **System-defined Defaults for this Page** to return all parameters on this tabbed page to their system-defined default values. For more information, see *Controlling the Time Incrementation Scheme*.

6. If desired, display the **Constraint Equations** tabbed page, and enter new values for one or more of the following values:
   - T^{vol}, volumetric strain compatibility tolerance for hybrid solid elements. Default $T^{vol} = 10^{-5}$.
   - T^{axial}, axial strain compatibility tolerance for hybrid beam elements. Default $T^{axial} = 10^{-5}$.
   - T^{tshear}, transverse shear strain compatibility tolerance for hybrid beam elements. Default $T^{tshear} = 10^{-5}$.
   - T^{cont}, contact and slip compatibility tolerance. If the **Convert severe discontinuity iterations** setting for the step is **On**, the ratio of the maximum error in the contact or slip constraints to the maximum displacement increment must be less than this tolerance.

     If the **Convert severe discontinuity iterations** setting for the step is **Off**, this item is used only with softened contact specified in the contact interaction property. The ratio of the error in the soft contact constraint clearance to the user-specified clearance at which the contact pressure is zero must lie below this tolerance for $p > p^0$, where p^0 is the pressure value at zero clearance. Default $T^{cont} = 5 \times 10^{-3}$.
   - T^{soft}, soft contact compatibility tolerance for low pressure. This tolerance, which is used only if the **Convert severe discontinuity iterations** setting for the step is **Off**, is similar to T^{cont} for

softened contact, except that it represents the tolerance when $p = 0.0$. The actual tolerance will be interpolated linearly between T^{cont} and T^{soft} for $0 \le p \le p^0$. Default $T^{soft} = 0.1$.

- T^{disp}, displacement compatibility tolerance for distributing coupling elements. The ratio of the error in the distributing coupling displacement compatibility to a measure of the characteristic length of the coupling arrangement must lie below this tolerance. This characteristic length is twice the average of the coupling node arrangement principal radii of gyration. Default $T^{disp} = 10^{-5}$.
- T^{rot}, rotation compatibility tolerance for distributing coupling elements. Default $T^{rot} = 10^{-5}$.

At any time you can click **System-defined Defaults for this Page** to return all parameters on this tabbed page to their system-defined default values. For more information, see *Defining Tolerances for Constraint Equations*.

7. If desired, display the **Line Search Control** tabbed page, and enter new values for one or more of the following values:

   - N^{ls}, maximum number of line search iterations. Default $N^{ls} = 0$ for steps that use the Newton method and $N^{ls} = 5$ for steps that use the quasi-Newton method. A suggested value for activation of the line search algorithm is $N^{ls} = 5$. Specify $N^{ls} = 0$ to forcibly deactivate the method.
   - s^{ls}_{max}, maximum correction scale factor. Default $s^{ls}_{max} = 1.0$.
   - s^{ls}_{min}, minimum correction scale factor. Default $s^{ls}_{min} = 0.0001$.
   - f^{ls}_{s}, residual reduction factor at which line searching terminates. Default $f^{ls}_{s} = 0.25$.
   - η^{ls}, ratio of new to old correction scale factors below which line searching terminates. Default $\eta^{ls} = 0.10$.

   At any time you can click **System-defined Defaults for this Page** to return all parameters on this tabbed page to their system-defined default values. For more information, see *Activating the "Line Search" Algorithm*.

8. If desired, display the **VCCT Linear Scaling** tabbed page, and enter a value for β. The default is $\beta = 0.9$.

   For most crack propagation simulations using VCCT or the enhanced VCCT criterion, the deformation can be nearly linear up to the point of the onset of crack growth; past this point the analysis becomes very nonlinear. In this case linear scaling can be used to reduce the solution time effectively to reach the onset of crack growth. For more information, see *Linear Scaling to Accelerate Convergence for VCCT in Abaqus/Standard*.

9. If desired, display the **Direct Cyclic** tabbed page, and enter new values for one or more of the following values:

   - I_{PI}, iteration number at which the periodicity condition is first imposed. Default $I_{PI} = 1$.
   - CR^{α}_{n}, stabilized state detection criterion for the ratio of the largest residual coefficient on any terms in the Fourier series to the corresponding average flux norm. Default $CR^{\alpha}_{n} = 5 \times 10^{-3}$.
   - CU^{α}_{n}, stabilized state detection criterion for the ratio of the largest correction to the displacement coefficient on any terms in the Fourier series to the largest displacement coefficient. Default $CU^{\alpha}_{n} = 5 \times 10^{-3}$.

- CR_0^α, plastic ratcheting detection criterion for the ratio of the largest residual coefficient on the constant term in the Fourier series to the corresponding average flux norm. Default $CR_0^\alpha = 5 \times 10^{-3}$.
- CU_0^α, plastic ratcheting detection criterion for the ratio of the largest correction to the displacement coefficient on the constant term in the Fourier series to the largest displacement coefficient. Default $CU_0^\alpha = 5 \times 10^{-3}$.

At any time you can click **System-defined Defaults for this Page** to return all parameters on this tabbed page to their system-defined default values. For more information, see *Controlling the Solution Accuracy in Direct Cyclic Analysis*.

10. Click **OK** to save your customized settings and to close the editor.

Additional information

• *General solution controls*

Customizing solver controls

The solver controls editor allows you to modify the parameters that control the iterative linear equation solver. For more information, see *Iterative Linear Equation Solver*.



Note:

Solver controls are available only for **Static, General**; **Static, Linear perturbation**; **Visco**; **Heat transfer**; **Geostatic**; and **Soils** analysis steps.

1. From the main menu bar, select **Other->Solver Controls->Manager**.

   Abaqus/CAE displays the **Solver Controls Manager** showing the steps defined in the model and the solver controls associated with each step.
2. In the **Solver Controls Manager**, select the analysis step of interest, and click **Edit**.

   The **Edit Solver Controls** dialog box appears.
3. Choose how you want to specify the solver controls:
   - Choose **Propagate from previous step** if you want the solver controls settings from the previous step to remain active in the current step.
   - Choose **Reset all parameters to their system-defined defaults** if you want all of the parameters in the editor to revert to their default values.
   - Choose **Specify** if you want to enter new values for particular parameters in this step.
4. Enter the data necessary to specify the customized solver controls:
   - In the **Relative tolerance** field, enter a value for the relative tolerance of convergence.
   - In the **Max. number of iterations** field, enter a value for the maximum number of linear solver iterations.
   - In the **ILU factorization fill-in level** field, enter a value for the incomplete LU factorization fill-in level (0 *level* 3). This setting is applicable only for **Geostatic** and **Soils** analysis steps.

   For more information on these parameters, see *Setting Controls for the Iterative Linear Solver*.
5. Click **OK** to save your customized settings and to close the editor.

Additional information

- *Solver controls*
- *Configuring analysis procedure settings*

The Interaction module

You can use the Interaction module to define and manage the following objects:

- Mechanical and thermal interactions between regions of a model or between a region of a model and its surroundings.
- The interface region and coupling schemes for an Abaqus/Standard to Abaqus/Explicit co-simulation.
- Analysis constraints between regions of a model.
- Assembly-level wire features, connector sections, and connector section assignments to model connectors.
- Inertia (point mass, rotary inertia, and heat capacitance) on regions of the model.
- Cracks on regions of the model.
- Springs and dashpots between two points of a model or between a point of a model and ground.

In this section:

- *Understanding the role of the Interaction module*
- *Entering and exiting the Interaction module*
- *Understanding interactions*
- *Understanding interaction properties*
- *Understanding constraints*
- *Understanding contact and constraint detection*
- *Understanding connectors*
- *Understanding connector sections and functions*
- *Understanding Interaction module managers and editors*
- *Understanding symbols that represent interactions, constraints, and connectors*
- *Using the Interaction module toolbox*
- *Using the Interaction module*
- *Using the interaction editors*
- *Using the interaction property editors*
- *Using the constraint editors*
- *Using contact and constraint detection*
- *Using the connector section editors*
- *Using the Query toolset to obtain connector assignment information*

Understanding the role of the Interaction module

You can use the Interaction module to define interactions.

You can define the following:

- Contact interactions.
- Elastic foundations.
- Cavity radiation.
- Thermal film conditions.
- Radiation to and from the ambient environment.
- Abaqus/Standard to Abaqus/Explicit co-simulation interaction.
- Pressure penetration.
- Incident waves.
- Acoustic impedance.
- Cyclic symmetry.
- A user-defined actuator/sensor interaction.
- Model change interactions.
- Tie constraints.
- Rigid body constraints.
- Display body constraints.
- Coupling constraints.
- Adjust points constraints.
- MPC constraints.
- Shell-to-solid coupling constraints.
- Embedded region constraints.
- Equation constraints.
- Connector section assignments.
- Inertia.
- Cracks.
- Springs and dashpots.

Interactions are step-dependent objects, which means that when you define them, you must indicate in which steps of the analysis they are active. (For more information about step-dependent objects, see *Understanding the status of an object in a step*.) For example, you can define film and radiation conditions on a surface only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step. Similarly, you can define an interaction with a user-defined actuator/sensor only during the initial step.

The Set and Surface toolsets in the Interaction module allow you to define and name regions of your model to which you would like interactions and constraints applied. You can use the Amplitude toolset to define variations in some interaction attributes over the course of the analysis. The Analytical Field toolset allows you to create analytical fields that you can use to define spatially varying parameters for selected interactions. The Reference Point toolset allows you to define reference points that are used in constraints and creating assembly-level wire features.

Abaqus/CAE does not recognize mechanical contact between part instances or regions of an assembly unless that contact is specified in the Interaction module; the mere physical proximity of two surfaces in an assembly is not enough to indicate any type of interaction between the surfaces.

For information on defining cracks to study their initiation and propagation, see *Fracture mechanics*. For information on defining inertia, see *Inertia*. For information on defining springs and dashpots, see *Springs and dashpots*.

Additional information

- *About Contact Interactions*
- *The Amplitude toolset*
- *The Analytical Field toolset*
- *The Reference Point toolset*
- *The Set and Surface toolsets*

Entering and exiting the Interaction module

You can enter the Interaction module at any time during an Abaqus/CAE session by clicking **Interaction** in the **Module** list located in the context bar. **Interaction**, **Constraint**, **Connector**, **Special**, **Feature**, and **Tools** menus appear on the main menu bar; and a **Step** list appears under the context bar.

To exit the Interaction module, click another module in the **Module** list. You need not take any specific action to save objects created in the Interaction module before exiting the module; they are saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

Additional information

- *Using the Special menu in the Interaction module*

Understanding interactions

You can use the Interaction module to define several types of interactions.

You can define the following types of interactions:

General contact

General contact interactions allow you to define contact between many or all regions of the model with a single interaction. General contact is also used to define contact between Lagrangian bodies and Eulerian materials in a coupled Eulerian-Lagrangian analysis (see *Defining contact in Eulerian-Lagrangian models*). Typically, general contact interactions are defined for an all-inclusive surface that contains all exterior faces; feature edges; and—in Abaqus/Explicit—analytical rigid surfaces, edges based on beams and trusses, and Eulerian material boundaries. To refine the contact domain, you can include or exclude specific surface pairs. Surfaces used in general contact interactions can span many disconnected regions of the model. Attributes, such as contact properties, surface properties, and contact formulation, are assigned as part of the contact interaction definition but independently of the contact domain definition, which allows you to use one set of surfaces for the domain definition and another set of surfaces for the attribute assignments. For detailed instructions on creating this type of interaction, see *Defining general contact*.

General contact interactions and surface-to-surface or self-contact interactions can be used together in the same analysis. Only one general contact interaction can be active in a step during an analysis.

For more information, see *About Contact Interactions*, *About General Contact in Abaqus/Standard*, *About General Contact in Abaqus/Explicit*, and *Eulerian Analysis*. The assignment of a penalty stiffness scale factor is not supported in Abaqus/CAE. In addition, node-based surfaces cannot be used in a general contact interaction in Abaqus/CAE.

Surface-to-surface contact, self-contact, and pressure penetration

Surface-to-surface contact interactions describe contact between two deformable surfaces or between a deformable surface and a rigid surface. Self-contact interactions describe contact between different areas on a single surface. For detailed instructions on creating these types of interactions, see *Defining surface-to-surface contact*, *Defining self-contact*, and *Using contact and constraint detection*. For more information, see *About Contact Pairs in Abaqus/Standard* and *About Contact Pairs in Abaqus/Explicit*.

If your model includes complex geometries and numerous contact interactions, you may want to customize the variables that control the contact algorithms to obtain cost-effective solutions. These controls are intended for advanced users and should be used with great care. For more information, see *Contact controls editors*.

A pressure penetration interaction allows you to simulate the pressure of a fluid penetrating between two surfaces involved in surface-to-surface contact. The fluid pressure is applied normal to the surfaces. You must create a surface-to-surface contact interaction to specify the main and secondary surfaces for the pressure penetration. The bodies forming the joint can both be deformable, as is the case with threaded connectors; or one can be rigid, as occurs when a soft gasket is used as a seal between stiffer structures. A pressure penetration interaction can be used only in an Abaqus/Standard analysis. For detailed instructions on creating pressure penetration interactions, see *Defining pressure penetration*. For more information, see *Fluid Pressure Penetration Loads*.

Fluid cavity

A fluid cavity interaction allows you to select and assign properties to a liquid- or gas-filled fluid cavity in the model. Fluid cavity selection includes a reference point and the surface that encloses the cavity. The properties are defined in a fluid cavity interaction property (for more information, see *Understanding interaction properties*). You can define fluid cavity interactions in the initial step of an Abaqus/Standard or an Abaqus/Explicit analysis. The fluid cavity interaction remains constant throughout all steps of an analysis; you cannot modify or deactivate

it after the initial step. For detailed instructions on creating fluid cavity interactions, see *Defining a fluid cavity interaction*.

Fluid exchange

A fluid exchange interaction allows you to define movement of fluid between a cavity and the environment or between two cavities. To create a fluid exchange interaction, you must first select an existing fluid cavity interaction for each cavity (one for exchange to environment or two for exchange between cavities). Then you can select or create a fluid exchange interaction property (for more information, see *Understanding interaction properties*) and set the effective exchange area. For detailed instructions on creating fluid exchange interactions, see *Defining a fluid exchange interaction*.

Fluid inflator

A fluid inflator interaction allows you to inflate a fluid cavity to model the flow characteristics of inflators used for airbag systems. To create a fluid inflator interaction, you must first select an existing fluid cavity interaction. Then you can select or create a fluid inflator interaction property (for more information, see *Understanding interaction properties*). For detailed instructions on creating fluid inflator interactions, see *Defining a fluid inflator interaction*.

XFEM crack growth

An XFEM crack growth interaction allows you to activate or deactivate growth of a crack created using the extended finite element method. For detailed instructions on creating this type of interaction, see *Deactivating and activating an XFEM crack growth*.

Model change

A model change interaction allows you to remove and reactivate elements during an analysis. You can use model change interactions in all Abaqus/Standard analysis procedures except for the static, Riks procedure and linear perturbation procedures. For detailed instructions on creating this type of interaction, see *Defining a model change interaction*. For more information on removing and reactivating elements, see *Element and Contact Pair Removal and Reactivation*.

Cyclic symmetry

Cyclic symmetry enables you to model an entire 360° structure at considerably reduced computational expense by analyzing only a single repetitive sector of a model. You can create cyclic symmetry interactions only in the initial step. Once a cyclic symmetry interaction is created, cyclic symmetry applies to the entire analysis history. If you deactivate a cyclic symmetry interaction in a frequency step, Abaqus/CAE evaluates all possible nodal diameters being evaluated for that step. For detailed instructions on creating this type of interaction, see *Defining cyclic symmetry*. For more information about cyclic symmetry in Abaqus, see *Analysis of Models that Exhibit Cyclic Symmetry*.

Elastic foundation (Abaqus/Standard only)

Elastic foundations allow you to model the stiffness effects of a distributed support on a surface without actually modeling the details of the support. You can create elastic foundation interactions only in the initial step. Once an elastic foundation is activated, you cannot deactivate it in later analysis steps. For detailed instructions on creating this type of interaction, see *Defining foundations*. For more information, see *Element Foundations*.

Cavity radiation (Abaqus/Standard only)

Cavity radiation interactions describe heat transfer due to radiation in enclosures. Two cavity radiation models are available in Abaqus/CAE: a fully implicit definition and an approximation. The full version can be used for heat transfer without deformation in two-dimensional, three-dimensional, and axisymmetric models. It can include open or closed cavities and accounts for symmetries and surface blocking, but it does not support surface motion within cavities. For detailed instructions on creating this type of interaction, see *Defining a cavity radiation interaction*.

The cavity radiation approximation is defined using a surface radiation interaction. You can approximate cavity radiation in any heat transfer analysis, with or without deformation. However, approximate cavity radiation can be used only for closed cavities in three-dimensional models. The approximation treats the cavity as a black body enclosure with a temperature equal to the average temperature of the entire surface. Under these limited conditions, approximate cavity radiation can save considerable computational expense. For detailed instructions on creating this type of interaction, see *Defining a surface radiative interaction*.

For more information on both types of cavity radiation, see *Cavity Radiation in Abaqus/Standard*.

Thermal film conditions

Film condition interactions define heating or cooling due to convection by surrounding fluids. Two types of film condition interaction are available in Abaqus/CAE: surface film conditions define convection from model surfaces, and concentrated film conditions define convection from nodes or vertices. You can define film condition interactions only during a heat transfer, fully coupled thermal-stress, or coupled thermal-electrical step. For detailed instructions on defining these types of interactions, see *Defining a surface film condition interaction*, and *Defining a concentrated film condition interaction*, respectively. For more information, see *Thermal Loads*.

Radiation to and from the ambient environment

Radiation interactions describe heat transfer to a nonreflecting environment due to radiation. Two types of radiation interactions are available in Abaqus/CAE: surface radiation interactions describe heat transfer with a nonconcave surface, and concentrated radiation interactions describe radiation from nodes or vertices. You can define radiation interactions only during a heat transfer, fully coupled thermal-stress, or coupled thermal-electrical step. For detailed instructions on creating these types of interactions, see *Defining a surface radiative interaction*, and *Defining a concentrated radiative interaction*, respectively. For more information, see *Thermal Loads*.

Abaqus/Standard to Abaqus/Explicit co-simulation

For an Abaqus/Standard to Abaqus/Explicit co-simulation, you must specify the interface region (region for exchanging data) and coupling schemes (time incrementation process and frequency of data exchange) for the co-simulation. In each model, you create a Standard-Explicit co-simulation interaction to define the co-simulation behavior; only one Standard-Explicit co-simulation interaction can be active in a model. The settings in each co-simulation interaction must be the same in the Abaqus/Standard model and the Abaqus/Explicit model.

A Standard-Explicit co-simulation interaction can be created only in a general static, implicit dynamic, or explicit dynamic step. The interaction is valid only in the step in which it is created and is not propagated to subsequent steps. For detailed instructions on creating this type of interaction, see *Defining a Standard-Explicit co-simulation interaction*. For more information, see *Structural-to-Structural Co-Simulation*.

Incident waves

Incident wave interactions model incident wave loading due to external acoustic wave sources. For detailed instructions on creating this type of interaction, see *Defining incident waves*. For more information, see *Acoustic and Shock Loads*.

Acoustic impedance

An acoustic impedance specifies the relationship between the pressure of an acoustic medium and the normal motion at an acoustic-structural interface. For detailed instructions on creating this type of interaction, see *Defining acoustic impedance*. For more information, see *Acoustic and Shock Loads*.

Actuator/sensor (Abaqus/Standard only)

An actuator/sensor interaction models a combination of sensors and actuators and, therefore, allows for modeling control system components. Currently, this type of interaction allows sensing and actuation at just one point. For detailed instructions on creating this type of interaction, see *Defining an actuator/sensor interaction*.

The interaction definition and its optional associated property are used to define the basic aspects of the interaction, but the user must provide user subroutine *UEL* to supply the specific formulae for how actuation depends on sensor readings. You specify the name of the file containing the user subroutine when you create the analysis job in the Job module.



Warning:

This feature is intended for advanced users only. Its use in all but the simplest test examples will require considerable coding by the user/developer. *User-Defined Elements*, should be read before proceeding.

Actuator/sensor interactions are available only for Abaqus/Standard analyses. For more information, see *About User Subroutines and Utilities*.

Additional information

• *Defining Contact Interactions*

Understanding interaction properties

You can define a set of data that is referred to by an interaction but is independent of the interaction; for example, the coefficients that define friction during contact. This set of data is called an interaction property.
One interaction property can be referred to by many different interactions.

You can create the following types of interaction properties:

Contact

A contact interaction property can define tangential behavior (friction and elastic slip) and normal behavior (hard, soft, or damped contact and separation). In addition, a contact property can contain information about damping, thermal conductance, thermal radiation, and heat generation due to friction. A contact interaction property can be referred to by a general contact, surface-to-surface contact, or self-contact interaction. For detailed instructions on defining this type of interaction property, see *Defining a contact interaction property*.

Film condition

A film condition interaction property defines a film coefficient as a function of temperature and field variables. A film condition interaction property can be referred to only by a film condition interaction. For detailed instructions on defining this type of interaction property, see *Defining a film condition interaction property*.

Cavity radiation

A cavity radiation interaction property defines emissivity for a cavity as a function of temperature and field variables. A cavity radiation interaction property can be referred to only by a cavity radiation interaction. For detailed instructions on defining this type of interaction property, see *Defining a cavity radiation interaction property*.

Fluid cavity

A fluid cavity interaction property defines the type of fluid occupying the cavity and the fluid properties. You can choose either a hydraulic fluid or a pneumatic fluid. Hydraulic fluids must include a fluid density; and they might include a fluid bulk modulus, thermal expansion coefficients, and other temperature-dependent data. Pneumatic fluids must include an ideal gas molecular weight, and they might include a molar heat capacity (Abaqus/Explicit only). For detailed instructions on defining this type of interaction property, see *Defining a fluid cavity interaction property*.

Fluid exchange

A fluid exchange interaction property defines the fluid flow between a cavity and the environment or from one cavity to another. You can define a fluid exchange based on bulk viscosity, mass flux, mass rate leakage, volume flux, or volume rate leakage. For detailed instructions on defining this type of interaction property, see *Defining a fluid exchange interaction property*.

Fluid inflator

A fluid inflator interaction property defines the mass flow rate and temperature as a function of inflation time either directly or by entering tank test data. It also defines the mixture of gases entering the fluid cavity. For detailed instructions on defining this type of interaction property, see *Defining a fluid inflator interaction property*.

Acoustic impedance

An acoustic impedance interaction property defines surface impedance or the proportionality factors between the pressure and the normal components of surface displacement and velocity in an acoustic analysis. An acoustic impedance interaction property can be referred to only by an acoustic impedance interaction. For detailed instructions on defining this type of interaction property, see *Defining an acoustic impedance interaction property*.

Incident wave

An incident wave interaction property defines the speed of the incident wave and other characteristics of the wave loading. An incident wave interaction property can be referred to only by an incident wave interaction. For detailed instructions on defining this type of interaction property, see *Defining an incident wave interaction property*.

Actuator/sensor

An actuator/sensor interaction property provides the `PROPS`, `JPROPS`, `NPROPS`, and `NJPROPS` variables that are passed into a *UEL* user subroutine used with an actuator/sensor interaction. For detailed instructions on defining this type of interaction property, see *Defining an actuator/sensor interaction property*.

Wear

A wear interaction property defines the contact wear properties based on Archard's wear rate model (see *Contact Wear*). For detailed instructions on defining this type of interaction property, see *Defining a wear interaction property*.

Understanding constraints

Constraints defined in the Interaction module define constraints on the analysis degrees of freedom, whereas constraints defined in the Assembly module define constraints only on the initial positions of instances. In the Interaction module you can constrain the degrees of freedom between regions of a model, and you can suppress and resume constraints to vary the analysis model. Currently, you can create the following types of constraints:

Tie

A tie constraint allows you to fuse together two regions even though the meshes created on the surfaces of the regions may be dissimilar. For detailed instructions on creating this type of constraint, see *Defining tie constraints*, and *Using contact and constraint detection*. For more information, see *Mesh Tie Constraints*.

Rigid body

A rigid body constraint allows you to constrain the motion of regions of the assembly to the motion of a reference point. The relative positions of the regions that are part of the rigid body remain constant throughout the analysis. For detailed instructions on creating this type of constraint, see *Defining rigid body constraints*. For more information on reference points, see *The Reference Point toolset*. For more information, see *Rigid Body Definition*.

Display body

A display body constraint allows you to select a part instance that will be used for display only. You do not have to mesh the part instance, and it is not included in the analysis; however, when you view the results of the analysis, the Visualization module displays the selected part instance. You can constrain the part instance to be fixed in space, or you can constrain it to follow selected nodes. You can apply a display body constraint to an instance of an Abaqus native part or to an instance of an orphan mesh part. For detailed instructions on creating this type of constraint, see *Defining display body constraints*. You can customize the appearance of display bodies in the Visualization module; for more information, see *Customizing the appearance of display bodies*.

A display body constraint is especially useful for mechanism or multibody dynamic problems where rigid parts interact with each other via connectors. In such cases you can create a simple rigid part, such as a point part, and a display body that is more representative of the physical part. For an example of a model that includes a display body constraint combined with connectors, see *Display bodies*. You can also use display bodies to model stationary objects that are not involved in the analysis but that help you to visualize the results.

For more information, see *Display Body Definition*.

Coupling

A coupling constraint allows you to constrain the motion of a surface to the motion of a single point. For detailed instructions on creating this type of constraint, see *Defining coupling constraints*. For more information, see *Coupling Constraints*.

Adjust points

An adjust points constraint allows you to move a point or points onto a specified surface. For detailed instructions on creating this type of constraint, see *Defining adjust points constraints*. For more information, see *Adjusting Nodal Coordinates*. This adjustment may be useful in assembled fasteners and other applications; see *About assembled fasteners*, and *Creating assembled fasteners*.

MPC constraint

An MPC constraint allows you to constrain the motion of the secondary nodes of a region to the motion of a single point. For detailed instructions on creating this type of constraint, see *Defining MPC constraints*. A multi-point constraint between two points is defined using connectors. For detailed instructions, see *Connectors*. For more information, see *General Multi-Point Constraints*.

Shell-to-solid coupling

A shell-to-solid coupling constraint allows you to couple the motion of a shell edge to the motion of an adjacent solid face. For detailed instructions on creating this type of constraint, see *Defining shell-to-solid coupling constraints*. For more information, see *Shell-to-Solid Coupling*.

Embedded region

An embedded region constraint allows you to embed a region of the model within a “host” region of the model or within the whole model. For detailed instructions on creating this type of constraint, see *Defining embedded region constraints*. For more information, see *Embedded Elements*.

Equation

Equations are linear, multi-point equation constraints that allow you to describe linear constraints between individual degrees of freedom. For detailed instructions on creating this type of constraint, see *Defining equation constraints*. For more information, see *Linear Constraint Equations*.

Understanding contact and constraint detection

The contact detection tool in Abaqus/CAE provides a fast and easy way to define contact interactions and tie constraints in a three-dimensional model.
Instead of individually selecting surfaces and defining the interactions between them, you can instruct Abaqus/CAE to automatically locate all surfaces in a model that are likely to interact based on initial proximity. You can tune the proximity settings and specify a variety of options that control the active search domain, the definitions of surfaces, and the default interaction or constraint settings. The search works for both geometry and meshed models.

Each detected interaction or constraint involves two identified surfaces, also known as a contact pair candidate. The contact detection dialog box lists each contact pair candidate and its default parameters in a tabular format. The default contact pair candidate parameters are slightly different than the default parameters used in the traditional Abaqus/CAE interaction editors; in particular, the contact detection tool initially assigns surface-to-surface discretization to each contact pair candidate instead of node-to-surface discretization.

Using the tabular interface, you can review the contact pair candidates to ensure that the surface definitions are comprehensive, the main and secondary assignments are appropriate, and the parameters are correct. If necessary, you can modify parameters or surface assignments in the table, and you can create new contact pairs where appropriate. Once the contact pair candidates are configured to your specifications, Abaqus/CAE defines all of the contact interactions and tie constraints simultaneously.

For step-by-step instructions on using the contact detection dialog box, see *Using contact and constraint detection*.

In this section:

- *The contact detection dialog box*
- *Use of the contact detection algorithm*
- *Default interaction and constraint parameters*
- *Tips for using the contact detection tool*

The contact detection dialog box

To use contact detection, select **Interaction->Find contact pairs** or **Constraint->Find contact pairs** from the main menu bar. The contact detection dialog box appears as shown in *Figure 1*. Initially there are no identified contact pairs.

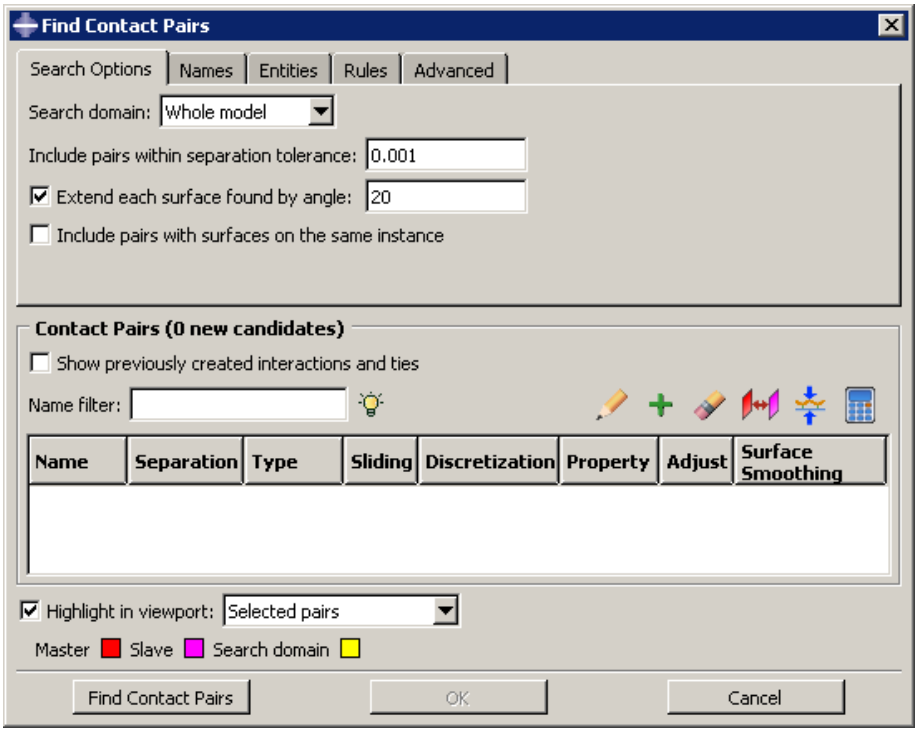


Figure 1: Initial view of the contact detection tool.

Using the contact detection tool is a two-step process: first Abaqus/CAE searches for surfaces in the model that are likely to interact; then you have a chance to review the identified surfaces and modify the default contact pair parameters before creating interactions and constraints. You provide some basic criteria to guide the search. These criteria include the search domain and the distance between surfaces that will likely be in contact.

After entering the necessary search criteria, click **Find Contact Pairs** to begin the search. Abaqus/CAE updates the contact pair candidates table, as illustrated in *Figure 2*.

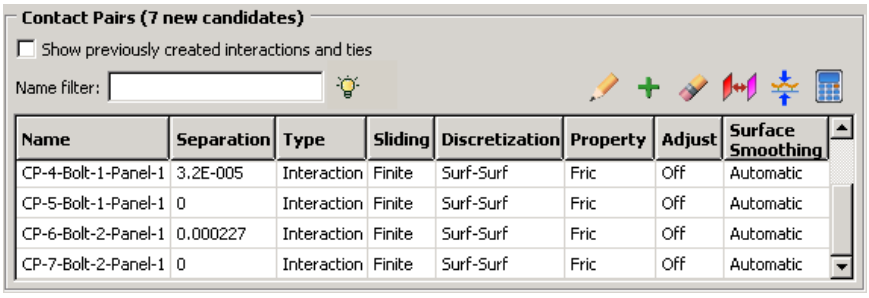


Figure 2: The contact pair candidates table.

You can create either a contact interaction or a tie constraint for each contact pair candidate in the table. You can also modify the parameters of the interaction or constraint definition by clicking on the appropriate table cell (see *Figure 3*).

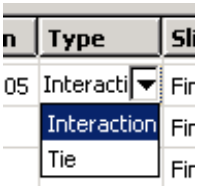


Figure 3: Changing cell values in the contact pair candidates table.

Clicking mouse button 3 on the table displays a menu of extended options and allows you to manually add contact pairs to the table. When you toggle on **Show previously created interactions and ties**, any pre-existing surface-to-surface interactions and tie constraints are added to the contact pair candidates table; you can modify existing contact pairs in the same manner as newly detected contact pair candidates.

The interactions and constraints shown in the contact pair candidates table do not become part of the model until you click **OK**. When you have finished setting parameters for the contact pairs, click **OK**. Abaqus/CAE simultaneously creates contact interactions and tie constraints for every contact pair in the table according to the specified parameters. The created interactions and constraints are added to the Model Tree and the **Interaction Manager**; you can review, modify, suppress, and delete the created interactions using either of these interfaces.

For detailed instructions on using the automatic contact detection tool, see *Using contact and constraint detection*.

Additional information

- *Understanding interactions*
- *Understanding constraints*
- *Managing objects in the Interaction module*
- *Understanding contact and constraint detection*

Use of the contact detection algorithm

This section discusses the use of the contact detection algorithm.

In this section:

- *The contact detection algorithm*
- *Additional criteria for defining contact pairs*
- *Contact detection for geometry*
- *Contact detection for meshed models*
- *Detection of overclosed surfaces*
- *Defining contact within the same instance and self-contact*
- *Considerations for shells*

The contact detection algorithm

Surfaces must meet two requirements to be identified by the automatic contact detection tool:

- The surfaces must be separated by a distance less than or equal to the specified separation tolerance.
- The surfaces must be intuitively opposed, as defined below.

Abaqus/CAE defines the separation between two surfaces as the distance between the points of closest approach on the surfaces. This distance is reported in the **Separation** column of the contact pair candidates table. The separation tolerance is the primary input used during the contact detection search. You should specify a separation tolerance that encompasses the separation distances between all of the potentially contacting surfaces in your model. For more information, see *Choosing a separation tolerance and extension angle* below.

Note:

The value reported in the **Separation** column may not correspond exactly to the separation used by Abaqus/Standard during an analysis. Certain automatic surface enhancements applied during the analysis to improve contact robustness (such as main surface smoothing and surface extensions) can lead to slight discrepancies between the separations calculated in the Abaqus/CAE preprocessor and the Abaqus/Standard analysis. For more details on automatic surface enhancements and contact formulations, refer to *Defining Contact Pairs in Abaqus/Standard*.

Two surfaces are considered intuitively opposed if the two surface normals constructed at the points of closest approach lie between 135° and 225° of each other (see *Figure 1*). In other words, the surfaces must be offset from each other by less than 45° at the points of closest approach. It is not possible to adjust or ignore the surface orientation requirement.

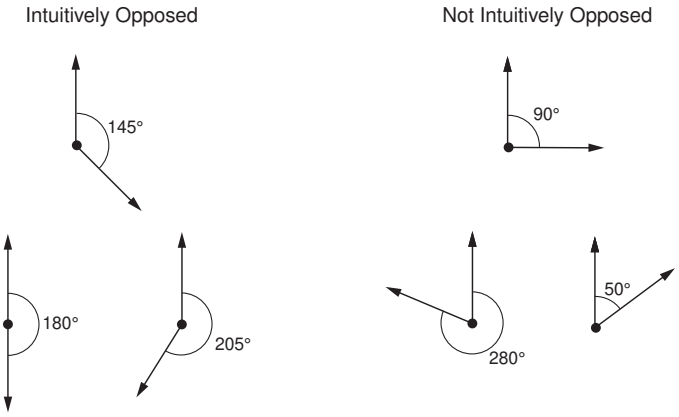


Figure 1: The relative orientation of the normals determines whether or not the surfaces are intuitively opposed.

Figure 2 illustrates a simple example of the contact pair requirements.

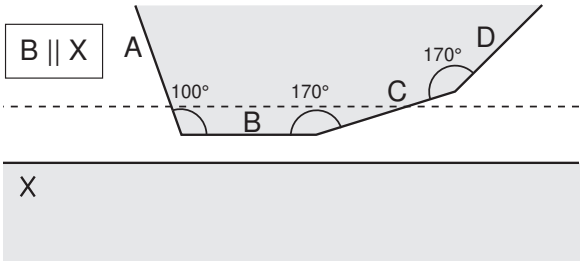


Figure 2: Two bodies involved in potential contact. The bodies are rendered in two dimensions for simplicity.

The dashed line represents the separation tolerance as calculated from surface *X*. Surface *B*, which is parallel to surface *X*, is identified as part of a contact pair because it is both within the separation tolerance and intuitively opposed to surface *X*. Similarly, surface *C* meets both of these criteria. Surface *D*, although it is intuitively opposed to surface *X*, does not lie within the separation tolerance at any point; surface *D* is not considered for inclusion in a contact pair. Surface *A*, although it is within the separation tolerance, is not intuitively opposed to surface *X*; therefore, surface *A* is also excluded from any contact pair definition. The connected surfaces (*A*, *B*, *C*, and *D*) do not form contact pairs with each other. By default, Abaqus/CAE only searches for surfaces on separate part instances. However, even if you were to enable searching within the same instance (see *Defining contact within the same instance and self-contact* below), these surfaces would not meet the orientation requirements.

Additional criteria for defining contact pairs

After using the separation and orientation checks to compile a list of potential contact pairs, the contact detection tool can perform a series of additional checks that adjust the surface definitions to make them more useful and realistic. All three of these additional checks are optional, but they are enabled by default.

Extending surfaces

By default, any surface identified by the contact detection tool is extended to include adjacent model faces within 20°, even if the adjacent faces do not meet the separation and orientation requirements. The 20° angle is measured as the offset between the normals of the detected surface and the adjacent face at the common edge. You can modify the extension angle using the **Extend each surface found by angle** option. As faces are added to the surface definition, Abaqus/CAE also checks any faces adjacent to the newly added faces. Abaqus/CAE eliminates any redundant definitions if an extended surface incorporates a face from a separately defined contact pair. For example, consider extending surfaces within 20° for the model in *Figure 2*. Abaqus/CAE creates a single contact pair: one surface consists of face *X*, and the other surface consists of faces *B*, *C*, and *D*. Face *D* is within 20° of face *C*, which is within 20° of face *B*; the redundant contact pair consisting of face *C* and face *X* is eliminated, since it is incorporated by the larger contact pair.

Merging contact pairs within a specified angle

You can use the **Merge pairs when surfaces are within angle** option to combine multiple contact pairs into a single definition. The faces involved in the contact pairs must be adjacent and they must lie within the specified angle (as described above). The merge option does not extend faces; it only combines positively identified contact pairs. By default, contact pairs with surfaces within 20° are merged by the contact detection tool. The merge option is typically used as an alternative to surface extension to merge contact pair candidates automatically without extending surface definitions beyond the separation tolerance. For example, merging pairs within 20° without extending surfaces for the model in *Figure 2* results in a single contact pair: one surface consists of face *X*, and the other surface consists of faces *B* and *C*.

Checking for surface overlap

By default, the contact detection tool eliminates any contact pairs whose surfaces do not "overlap"; two surfaces do not overlap if a normal from any point on one of the surfaces does not pass through the opposing surface. For example, the surfaces in *Figure 1* do not overlap, even though they may pass the separation and orientation checks.

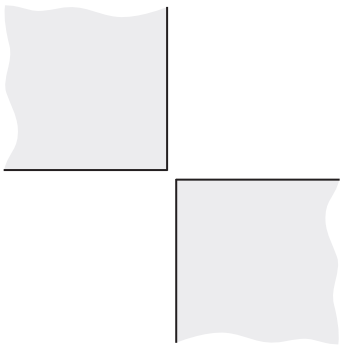


Figure 1: Non-overlapping surfaces. The bodies are rendered in two dimensions for simplicity.

You can suppress the check for surface overlap and allow the creation of contact pairs for non-overlapping surfaces by using the **Include opposing surfaces that do not overlap** option.

Contact detection for geometry

Abaqus/CAE begins searching a model comprised of geometry by dividing the model into individual faces. A face consists of the area enclosed in connected geometric edges or partitions. Once all of the faces are identified, Abaqus/CAE compares the faces to determine if they meet the separation and orientation requirements, then defines surfaces from the faces by applying extension, merging, and overlap checks (see *Additional criteria for defining contact pairs* above). Any two surfaces that meet all of the requirements are flagged as a contact pair candidate.

Abaqus/CAE automatically assigns the main and secondary designations to surfaces in a detected contact pair. Analytical rigid or discrete rigid surfaces are always assigned the main role; if the contact pair involves two rigid surfaces, the assignment of main and secondary roles is arbitrary. For contact pairs involving two deformable surfaces, Abaqus/CAE first determines if the surface geometry has been meshed and assigns the main role to the surface with the coarser mesh. If mesh information is unavailable, the surface with the larger area becomes the main surface. The algorithm that assigns main and secondary roles does not account for dissimilar underlying stiffness or element assignments; if these factors play a significant role in your contact interactions, you should review the main and secondary assignments before creating an interaction. For further discussion of main and secondary assignments, see *Selecting Surfaces Used in Contact Pairs*.

Contact detection for meshed models

Contact detection also works with mesh models. The search algorithm for meshed models works in much the same way as with geometry, but it uses element faces instead of geometric faces. By default, Abaqus/CAE only searches for contact pairs between separate part instances. Mesh models that are imported into Abaqus/CAE often consist of only a single part instance; therefore, you should enable searching within the same instance before using contact detection on these models (see *Defining contact within the same instance and self-contact* below for more details).

Warning:

Unlike geometry-based searches, the reported separation between surfaces for mesh-based surfaces is not necessarily the distance between the exact points of closest approach, but rather a close approximation. If the specified search tolerance is very large compared to the characteristic element size, the accuracy of this approximation is greatly reduced.

Before defining surfaces on element faces, Abaqus/CAE applies the same extension, merging, and overlap checks as with geometry faces (see *Additional criteria for defining contact pairs* above). Because element faces are typically much smaller than geometric faces, you should always allow some extension of the surfaces to get ample coverage from a surface definition; *Figure 1* compares the created surfaces for geometry and meshed geometry when no surface extension is allowed.

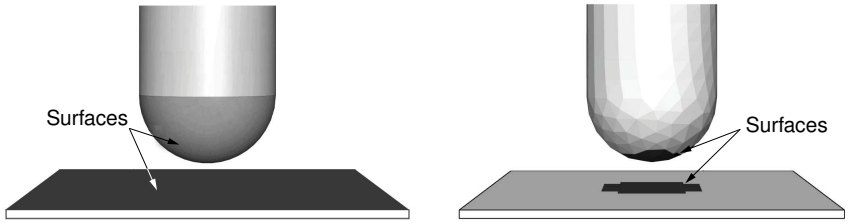


Figure 1: Discrepancies between created surfaces when no surface extension is allowed for geometry (left) and meshed geometry (right).

If you remesh your model, any surfaces defined on elements faces may become invalid. By extension, the interactions and constraints based on these faces also become invalid.

When assigning main and secondary designations to the mesh surfaces, rigid surfaces always become the main; if the contact pair involves two rigid surfaces, the assignment of main and secondary roles is arbitrary. For contact pairs involving two deformable surfaces, Abaqus/CAE considers the mesh densities on each surface; the surface with the coarser mesh becomes the main surface. If the mesh densities on the two surfaces are equivalent, the assignment of main and secondary roles is arbitrary. The algorithm that assigns main and secondary roles does not account for dissimilar underlying stiffness or element types; if these factors play a significant role in your contact interactions, you should review the main and secondary assignments before creating an interaction. For further discussion of main and secondary assignments, see *Selecting Surfaces Used in Contact Pairs*.

The contact detection tool does not detect contact between geometry and orphan elements or analytical surfaces and orphan elements. If your model includes part instances that have been meshed from geometry, you can use the options on the **Advanced** tabbed page of the contact detection dialog box to indicate whether these instances should be treated as geometry (the default) or an element mesh during the search. If your model contains instances of both geometry and orphan mesh elements, you should first mesh all of the geometries, then perform a mesh-based search to capture all possible contact pairs.

In most cases the geometry is a more faithful representation of the object being modeled than the meshed geometry. In addition, geometry-based interactions and constraints are not affected by remeshing. However, the mesh is the geometry used in the analysis. Mesh discretization can lead to slight disparities in separation distances between the

two representations, which may become important in precise analyses. After searching, you can check individual contact pairs for disparities between the native and meshed geometry by using the **Recalculate Separation** option.

Detection of overclosed surfaces

If two faces in an assembly intersect at any point, the contact detection tool reports those faces as an overclosed contact pair. Overclosed contact pairs that appear in the contact pair candidates table must still meet the surface orientation requirements. A red zero in the **Separation** column indicates that the two surfaces in the contact pair are intersecting.

Note: A black zero in the **Separation** column implies that the two surfaces are exactly touching at their closest points. There is no overclosure or intersection in this situation.

If you extend or merge an overclosed surface to include faces that are not overclosed, Abaqus/CAE reports the entire contact pair as overclosed.

You should visually inspect all overclosed surfaces before creating contact interactions. Models with severe overclosures should be adjusted to remove the overclosures (or at least lessen their severity). Minor overclosures can be addressed by using the contact adjustment options (available in the contact pair candidates table) or the interference fit options (available in the contact interaction editor).

Faces must intersect to be reported as overclosed. If a face is enclosed entirely within another part instance, the automatic contact detection tool does not report that face as being overclosed. Such a face may still meet the separation and orientation requirements with respect to an external face on the enclosing instance. By default, Abaqus/CAE eliminates enclosed faces from the contact pair candidates table because the surfaces do not "overlap" (see *Additional criteria for defining contact pairs*). If you disable the overlap checks, Abaqus/CAE reports a contact pair candidate for enclosed faces, but the contact pair candidates table does not provide any indication that the surfaces are overclosed or penetrating. Because the contact detection tool does not recognize these faces as overclosed, the adjustment options that are applied to overclosed surfaces by default (see *Default interaction and constraint parameters*) are not applied to this contact pair. If an enclosed face is embedded deeper than the separation tolerance from any external face, the automatic contact detection tool does not identify those faces as a contact pair candidate.

As an example, consider the model in *Figure 1*.

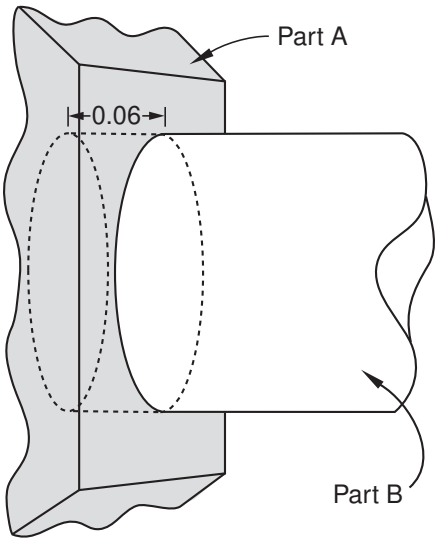


Figure 1: In this model, one end of the cylindrical part instance is entirely enclosed within another part instance.

The separation tolerance specified for this search is 0.1. The circular face at the end of part instance B is within the separation tolerance and is intuitively opposed to the rectangular face on part instance A, but there is no intersection. The contact pair candidates table lists a normal contact pair consisting of the circular face and the rectangular face separated by a distance of 0.06. The cylindrical side face of part instance B is listed as overclosed because it intersects the rectangular face of part instance A.

Although the contact detection tool does not recognize completely enclosed surfaces as being overclosed, such surfaces are still treated as overclosures during an analysis. Severe overclosures commonly lead to convergence difficulties. When reviewing overclosed contact pairs in the contact pair candidates table, check adjoining surfaces for fully enclosed faces.

Defining contact within the same instance and self-contact

You can use the contact detection tool to define contact between different areas of the same part instance or model instance. This capability is particularly useful for complicated models that are imported into Abaqus/CAE as a single part instance. If you enable the **Include pairs with surfaces on the same instance** option, Abaqus/CAE checks different geometry or element faces on the same part instance or model instance to determine whether or not they meet the separation and orientation requirements. Surfaces and contact pairs are defined on any faces that meet the requirements.

In some situations, surface extension options cause the main and secondary surfaces to overlap. If the main surface and the secondary surface consist of the same faces, Abaqus/CAE automatically adjusts the contact pair to create a self-contact interaction, in which a surface contacts itself during deformation. A single surface is created in this situation.

Considerations for shells

If a section definition has been assigned to a shell part, the contact detection tool accounts for the thickness of the shell in separation calculations. The reported separation has been adjusted according to the thickness and offset specified in the shell section assignment. You can use the **Account for shell thickness and offset** option to ignore shell section properties during a contact detection search. Varying thickness distributions are never considered in the separation calculations.

The contact detection tool automatically selects a shell side on which to create a surface (see *Specifying a particular side or end of a region*, for more information). The side is selected such that the surface normals at the points of closest approach are intuitively opposed.

When working with a model that includes orphan shell elements, make sure that the element normal orientations are consistent between elements; that is, the positive element faces (SPOS) should all be located on the same side of the shell structure (see *About Shell Elements*, for more information). If the element normal orientations are inconsistent, Abaqus/CAE misinterprets the angles between the element faces, and the surface extension and merge operations do not function appropriately.

For certain spline-based shells or faces, the same surface may interact with both sides of a shell (see *Figure 1*, for example).

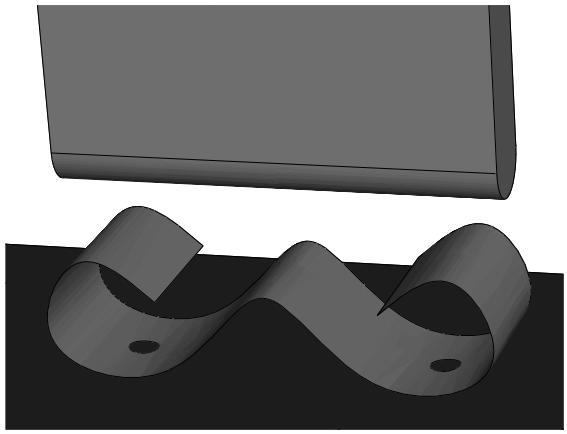


Figure 1: A spline-based shell.

Normally you would define a separate contact pair involving each side of the shell. The contact detection tool, however, cannot create multiple contact pairs that involve the same two faces; it will define a single contact pair and select the shell side according to the orientation at the point of closest approach. You must manually define another contact pair for the other side of the shell.

The contact detection tool does not create any double-sided surfaces. If appropriate, you can edit the definition of a created surface in the Model Tree to make it double-sided (see *Editing sets and surfaces*).

Default interaction and constraint parameters

After completing a search for potential contact pairs, Abaqus/CAE populates the contact pair candidates table with all of the parameters necessary to create interactions.

Names are provided for the contact pair and any created surfaces. *Table 1* outlines the naming algorithm.

Table 1: Algorithms used to create names in the contact detection tool.

Contact pairs	*Prefix*-*Contact_pair_number*-*Main_instance*-*Secondary_instance*
Main surfaces	*Prefix*-*Contact_pair_number*-*Main_instance*
Secondary surfaces	*Prefix*-*Contact_pair_number*-*Secondary_instance*
Merged main surface	*Prefix*-`All-m`
Merged secondary surface	*Prefix*-`All-s`
Merged “all” surface	*Prefix*-`All`

Use the **Names** tabbed page before searching to modify the naming prefix and control the creation of surfaces. For details, see *Specifying naming options for contact detection*.

The default parameters supplied to contact pairs by the contact detection tool are slightly different than the defaults used in the traditional interaction or constraint editor. Most notably, the contact detection tool initially assigns surface-to-surface discretization to each contact pair instead of node-to-surface discretization. See *Mesh Tie Constraints* and *Contact Formulations in Abaqus/Standard* for a discussion of surface discretization and the associated constraint enforcement methods.

The default surface adjustment options depend on the separation between the surfaces in a contact pair. You can use the **Rules** tabbed page before searching to control the default adjustment options that are assigned to detected contact pairs. You can also use this page to specify a separation tolerance within which all contact pairs default to tie constraints. For more information about the **Rules** page, see *Defining default contact pair parameters*.

Table 2 lists the default contact pair parameters supplied by Abaqus/CAE. You can edit each parameter individually before creating interactions and constraints. For detailed instructions on editing parameters and defaults, see *Reviewing and modifying detected contact pairs*.

Table 2: Default contact pair parameters for the contact detection tool.

Parameter	Default Value
Active/Suppressed	Active
Type[1]	Interaction
Sliding	Finite sliding
Discretization	Surface-to-surface
Interaction property	The first contact interaction property listed in the **Interaction Property Manager**[2]; if no interaction properties have been created, this parameter is blank
Contact controls	This parameter is blank; contact controls are unavailable in the initial step
Adjust[1]	**Off** for contact interactions between nonintersecting surfaces; **0** for contact interactions between intersecting surfaces; **On** for tie constraints
Creation step	Initial
Surface smoothing	Automatic
[1]Defaults for the Type and Adjust parameters are controlled by the **Rules** options.	
[2]The **Interaction Property Manager** lists all created interaction properties alphabetically by name.	



Note:

Some of the parameters discussed above are not visible in the contact pair candidates table by default. Click mouse button 3 anywhere in the table, and select **Edit Visible Columns** to control which parameters appear in the table.

For more information about interaction and constraint parameters, see *Mesh Tie Constraints*, *About Contact Pairs in Abaqus/Standard*, and *About Contact Pairs in Abaqus/Explicit*.

Tips for using the contact detection tool

The contact detection tool is available for use in any three-dimensional model requiring the creation of contact interactions and tie constraints. It quickly and thoroughly identifies and creates interactions and ties based on minimal specifications.

The tool greatly simplifies the contact definition process in models for which a general contact definition is not applicable. Some basic guidelines ensure the most effective and efficient use of the tool.

In this section:

- *Choosing a separation tolerance and extension angle*
- *Reviewing contact pair candidates*
- *Saving the search parameters*
- *Features that may cause difficulties for the contact detection tool*
- *Limitations of the contact detection tool*

Choosing a separation tolerance and extension angle

The specified separation tolerance is the primary driver of the contact pair search algorithm. Abaqus/CAE supplies a default separation tolerance based on the relative size of the faces in your model. You may need to modify this value depending on the expected response of your model during an analysis. To effectively capture all significant contact pairs, the specified separation tolerance should be on the same order as or greater than the expected displacements or deflections in your model.

Specifying a very large separation tolerance usually captures more contact pairs then are necessary in an analysis. While extra contact pairs do not necessarily reduce the quality of a model, the extraneous definitions are difficult to manage and can degrade performance.

When selecting an angle to control the extension of surfaces, you should consider the topology and surface characteristics of the areas that are likely to come into contact. Surfaces should extend slightly beyond the area of potential contact, so set the extension angle to capture any chamfers or soft corners along the edges of a face. Indentations, grooves, or embossments can sometimes break up the definition of a surface; the angle that these features make with the main face should dictate the extension angle.

For meshed models, you can preview the extension of surfaces before searching for contact pairs by displaying only the feature edges on a model (see *Defining mesh feature edges*). If the extension angle is equal to the feature angle, the surface definition in a particular area extends as far as the nearest visible feature edge. Adjust the feature angle until the visible edges enclose the area you want to capture, then set the extension angle accordingly.

Reviewing contact pair candidates

You should always review contact pair candidates before creating interactions and constraints. Look for any discontinuities in surface definitions.

Discontinuities are often caused by small connecting faces that are not intuitively opposed to the logical contacting surface in a contact pair (see *Figure 1*).

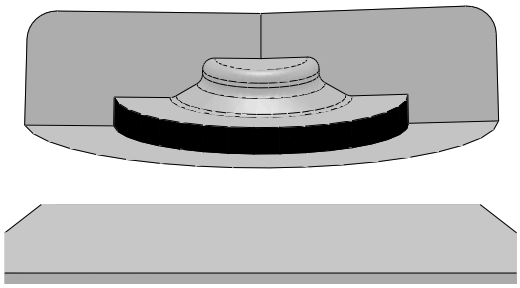


Figure 1: The automatic contact detection tool will not identify the highlighted perpendicular face.

You may want to rerun the search using revised extension and merge options to incorporate the discontinuities into larger surfaces. If necessary, add a contact pair manually using the **Add** option. You can also combine discontinuous surfaces using the **Merge** option.

You should investigate any intersecting surfaces to verify that they match your modeling intent. A contact pair with only a single overclosed node will be reported as intersecting, so slight discrepancies can cause overclosures. Overclosed contact pairs without appropriate adjustment or interference fit options can lead to convergence difficulties in an analysis. You should also check any faces or surfaces adjacent to overclosed contact pairs to ensure they are not enclosed faces. See *Detection of overclosed surfaces*, for more information.

Saving the search parameters

By default, the search parameters that you specify in the **Find Contact Pairs** dialog box persist only as long as the dialog box is open; if you close the dialog box, default search parameters are provided the next time you access the contact detection tool.

If you click  on the **Advanced** tabbed page, Abaqus/CAE sets the currently specified search parameters as the default search parameters. These parameters are supplied as defaults in all future sessions of Abaqus/CAE. The only parameter that is not saved is the search domain, which always uses a default of **Whole model**.

When you save the current search parameters, Abaqus/CAE asks if you want to save the current separation tolerance as a default. Normally Abaqus/CAE recalculates the default separation tolerance based on the current model; if you opt to save the separation tolerance, this calculation is skipped and the same value is always provided as the default separation tolerance.

The default search parameters for the contact detection tool are saved in the `abaqus_2025.gpr` file; see *Understanding Abaqus/CAE GUI settings*, for more information. To return the default search parameters to their original settings, click  on the **Advanced** tabbed page.

Features that may cause difficulties for the contact detection tool

You may encounter difficulties using the contact detection tool with certain model features and designs. These situations do not cause performance or stability problems, but the search results most often will not match your modeling intent.

Stacked shells and thin layers

Models with layers of shells or thin plates stacked closely in parallel can lead to the definition of extraneous contact pairs. The automatic contact detection tool can find contact pairs involving surfaces separated by an intermediate layer, as long as these surfaces are intuitively opposed and within the separation tolerance. In addition, if searching within the same instance is enabled and the overlapping surface check is disabled, the contact detection tool may detect potential contact between the top side and bottom side of a thin continuum plate. Abaqus/CAE creates contact pair candidates for all of these surfaces, even though they will never be in contact. This problem is most common when the layers or plates are a local feature of the model, since a larger separation tolerance is required to capture surfaces in other areas of the model. To overcome this problem, limit the search domain to a particular area of the model and use a separation tolerance that is appropriate for that area. You may also be able to use the **Entities** tabbed page of the contact detection dialog box to eliminate certain geometry or element types (shells, for example) from your search domain. Otherwise, you should delete the extraneous contact pair candidates before creating interactions.

Concave surfaces

While the contact search algorithm effectively accounts for most appropriate surfaces, it can misinterpret the relationship between a concave surface and a flat surface. Concave surfaces create difficulties because their surface normal orientation can vary widely across the span of a single surface, and the points of closest approach between surfaces is sometimes a poor reference. Consider, for example, the situations in *Figure 1*.

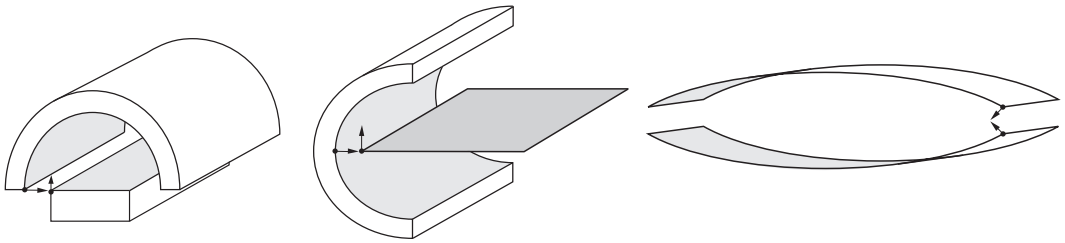


Figure 1: The normals of the shaded surfaces are not intuitively opposed at the points of closest approach.

Even if the points of closest approach in these models are within the separation tolerance, the surface normals at these points do not pass the orientation test. The contact detection tool will not report these surfaces as contact pair candidates, and adjusting the separation tolerance has no effect on this behavior. You can sometimes modify the extension angle to capture the concave surface within another surface definition. Otherwise, you must manually define the contact pair using the **Add** option.

Mechanisms involving large rotations

When modeling mechanisms that undergo large rotations, the contact detection tool often will not effectively capture your modeling intent. In such mechanisms the intended contact surfaces initially may be positioned far away from each other, while nearby surfaces never actually come into contact. The Geneva mechanism depicted in *Figure 2* is a typical example.

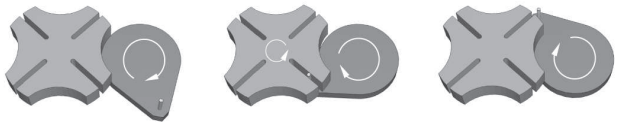


Figure 2: Motion of a Geneva mechanism.

The important contact surfaces in this model are the pin on the right-hand body and the slots on the left-hand body. In the initial configuration, the pin is relatively distant from any of the slots. The neighboring surfaces, on the other hand, are insignificant to the contact conditions of the model. Contact for such models is best defined manually using the interaction editor (see *Defining surface-to-surface contact*).

Limitations of the contact detection tool

Although helpful for simplifying the contact definition process, several limitations exist for the contact detection tool.

Contact detection cannot create contact pairs involving the following features:

- Two-dimensional models
- Axisymmetric models
- Beams and trusses
- Face-to-edge contact
- Edge-to-edge contact
- Contact between orphan mesh elements and analytical rigid surfaces
- Hybrid models containing both orphan mesh and unmeshed geometry

The minimum allowable separation tolerance is 1×10^{-5}. The maximum allowable separation tolerance is 1×10^{5}. Abaqus/CAE cannot accurately calculate separations outside of this range. If your model requires the use of a separation tolerance that does not meet these requirements, you should scale the dimensions of the entire model so that they fall within the functional range.

Understanding connectors

Connectors allow you to model a connection between two points in an assembly or between a point in an assembly and ground. To model a connector in Abaqus/CAE, you must create an assembly-level wire feature, a connector section, and a connector section assignment that associates the connector section with selected wires.

The wire feature contains one or more wires that define the underlying connector geometry. The connector section specifies the type of connection, connector behaviors, and section data. Similar to the manner in which you assign a section to a region of the model in the Property module, you create a connector section assignment to assign a connector section to a region of the model; specifically, you assign a connector section to wires. You also specify local orientations for the endpoints of the wires in the connector section assignment definition.

For more information on connectors in Abaqus/CAE, including an overview and an example of connector modeling, see *Connectors*.

Additional information

- *About Connectors*
- *Creating or modifying wire features for multiple connectors*
- *Creating connector sections*
- *Creating and modifying connector section assignments*

Understanding connector sections and functions

The connector section defines the connection type and may include connector behavior and section data. For some complex coupled connector behaviors, additional functions describing the nature of the coupling effects (connector derived components and connector potential) must be defined. A connector section can be referred to by one or more different connector section assignments.

In this section:

- *Connection types*
- *Connector behaviors*
- *What types of friction models are available?*
- *Connector derived components and connector potentials*

Connection types

Table 1 summarizes the connection types available when creating connector sections. You can define basic, assembled, complex, and MPC connection types.

Table 1: Connection types.

Basic Types		Assembled/Complex Types	MPC Types
Translational	Rotational		
ACCELEROMETER	*ALIGN*	*BEAM*	*Beam*
AXIAL	*CARDAN*	*BUSHING*	*Elbow*
CARTESIAN	*CONSTANT VELOCITY*	*CVJOINT*	*Link*
JOIN	*EULER*	*CYLINDRICAL*	*Pin*
LINK	*FLEXION-TORSION*	*HINGE*	*Tie*
PROJECTION CARTESIAN	*FLOW-CONVERTER*	*PLANAR*	*User-defined*
RADIAL-THRUST	*PROJECTION FLEXION-TORSION*	*RETRACTOR*	
SLIDE-PLANE	*REVOLUTE*	*SLIPRING*	
SLOT	*ROTATION*	*TRANSLATOR*	
	ROTATION-ACCELEROMETER	*UJOINT*	
	UNIVERSAL	*WELD*	

Basic types

Basic connection types include translational types and rotational types. Translational types affect translational degrees of freedom at both endpoints of the wires to which the connector section is assigned and may affect rotational degrees of freedom at the first points of the wires. Rotational types affect only rotational degrees of freedom at both endpoints of the wires. You can use a single basic connection type (translational or rotational) or one translational and one rotational type.

Assembled types

Assembled connection types are predefined combinations of basic connection types.

Complex types

Complex connection types affect a combination of degrees of freedom in the connection and cannot be combined with other connection types. They typically model highly coupled physical connections.

MPC types

MPC connection types are used to define multi-point constraints between two points.

For a description of each connection type and the equivalent basic connection types that define the kinematic constraints of assembled type connections, see *Connection Types* and *General Multi-Point Constraints*.

Connector behaviors

You can apply connector behaviors to connection types that have available components of relative motion. Available components of relative motion are displacements and rotations that are not kinematically constrained. Multiple connector behaviors can be defined in a connector section. You can specify the following connector behaviors:

- ***Elasticity***: Define spring-like elastic behavior.
- ***Damping***: Define dashpot-like damping behavior.
- ***Friction***: Define Coulomb-like and hysteretic friction using predefined or user-defined friction models.
- ***Plasticity***: Define plastic behavior.
- ***Damage***: Define damage initiation and evolution behavior.
- ***Stop***: Define limit values of the admissible range of positions.
- ***Lock***: Specify a user-defined locking criterion.
- ***Failure***: Define limit values for force, moment, or position.
- ***Reference Length***: Define the translational or angular positions at which constitutive forces and moments are zero.
- ***Integration***: Specify implicit or explicit time integration for elasticity, damping, and friction (Abaqus/Explicit analyses only).

For detailed instructions on defining connector behaviors, see *Using the connector section editors*. For more information on connector behaviors, see *Connector Behavior*.

What types of friction models are available?

You can model predefined or user-defined friction behavior. In general, for predefined friction you specify a set of geometric quantities that are characteristic of the connection type for which friction is modeled. In addition, you can define internal contact force contributions, such as prestress from the connection. Abaqus automatically defines the contact force contributions and the local tangent directions along which friction occurs.

You can model predefined friction for the following connection types:

Assembled/Complex types

- Cylindrical (Slot + Revolute)
- Hinge (Join + Revolute)
- Planar (Slide-Plane + Revolute)
- Slip Ring (complex)
- Translator (Slot + Align)
- U Joint (Join + Universal)

Basic types

- Slide-Plane
- Slot

Predefined friction is also available if you define a combined translational and rotational connection type that is equivalent to one of these assembled types. You can define only one friction behavior for a given connection type if you are modeling predefined friction.

If a predefined friction model is not available or does not adequately describe the mechanism being analyzed, you can specify a user-defined friction model (except in the case of the **Slip Ring** connection type, which does not allow a user-defined friction model). You must specify slip direction information, the friction-producing normal force or normal moment, and the friction law. You may use several connector friction behaviors to represent the frictional effects in the connector.

For detailed instructions on defining friction, see *Defining friction*. For more information, see *Connector Friction Behavior*.

Connector derived components and connector potentials

You can define complex coupled behavior for connectors using connector derived components and connector potentials. Connector derived components are user-specified component definitions based on a function of intrinsic connector components of relative motion. You can create derived components to specify the friction-generating normal force in connectors as a complex combination of connector forces and moments or to use as an intermediate result in a connector potential function.

Connector potentials are user-defined mathematical functions of intrinsic components of relative motion or derived components. These functions can be quadratic, elliptical, or maximum norms. You use connector potentials to define coupled friction, plasticity, and damage connector behaviors.

For detailed instructions on defining derived components and potentials, see *Specifying connector derived components*, and *Specifying potential terms*. For more information on connector functions, see *Connector Functions for Coupled Behavior*.

Understanding Interaction module managers and editors

You can create and manage objects in the Interaction module using managers and editors.

In this section:

- *Managing objects in the Interaction module*
- *Interaction editors*
- *Interaction property editors*
- *Contact controls editors*
- *Contact initialization editor*
- *Constraint editors*
- *Connector section editors*
- *Connector section assignment editors*

Managing objects in the Interaction module

The Interaction module provides the following managers that you can use to organize and manipulate objects associated with a given model:

- The **Interaction Manager** allows you to create and manage interactions.
- The **Interaction Property Manager** allows you to create and manage interaction properties.
- The **Contact Controls Manager** allows you to create and manage contact controls for surface-to-surface contact and self-contact interactions.
- The **Contact Initialization Manager** allows you to create and manage contact initialization rules for general contact interactions in Abaqus/Standard.
- The **Constraint Manager** allows you to create and manage constraints.
- The **Connector Section Manager** allows you to create and manage connector sections.
- The **Connector Section Assignment Manager** allows you to create and manage connector section assignments.

For example, a list of interaction properties appears in the **Interaction Property Manager** shown in *Figure 1*.

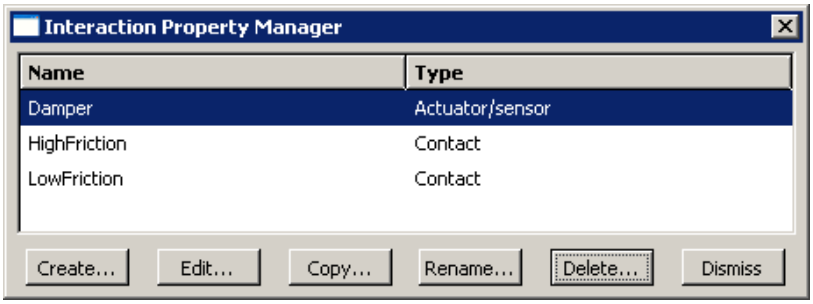


Figure 1: The Interaction Property Manager.

The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the managers allow you to create new objects or to edit, copy, rename, and delete existing ones. In the **Connector Section Assignment Manager**, you can only create, edit, or delete connector section assignments. You can also initiate these procedures using the **Interaction**, **Interaction->Property**, **Interaction->Contact Controls**, **Interaction->Contact Initialization**, **Constraint**, **Connector->Section**, and **Connector->Assignment** menus from the main menu bar. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

You can use the **Copy** button in the **Interaction Manager**, the corresponding menu command, or the Model Tree to copy an interaction. You can copy an interaction from any step to any valid step, with some restrictions. For more details, see *Copying step-dependent objects using manager dialog boxes*.

The **Interaction Manager** is a step-dependent manager, which means that it contains additional information on the history of each interaction through the analysis. The **Interaction Manager** is shown in *Figure 2*.

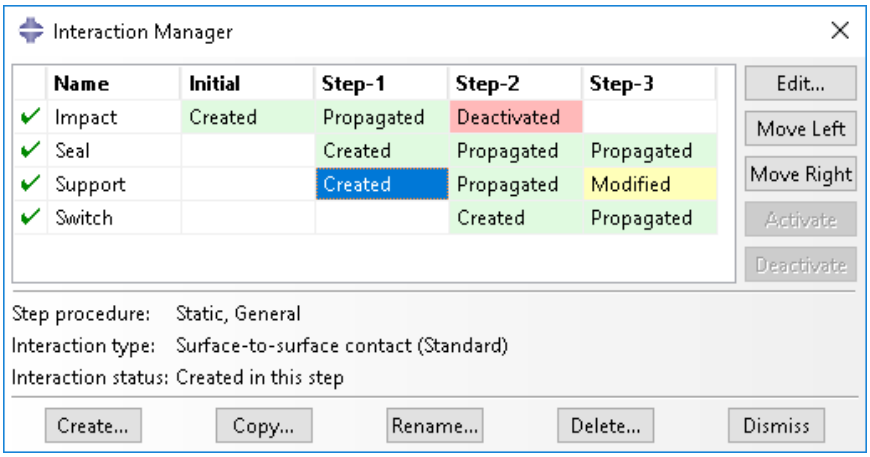


Figure 2: The Interaction Manager.

The **Move Left**, **Move Right**, **Activate**, and **Deactivate** buttons allow you to manipulate the stepwise history of interactions. For more information, see *Modifying the history of a step-dependent object*.

You can suppress and resume previously defined interactions, constraints, and connector section assignments from the managers. You can use the icons in the column along the left side of the manager to suppress these attributes or to resume previously suppressed attributes for an analysis. The suppress and resume procedures are also available from the **Interaction**, **Constraint**, and **Connector** menus in the main menu bar. For more information, see *Suppressing and resuming objects*.

For detailed instructions on creating interactions, interaction properties, constraints, connector sections, and connector section assignments, see *Using the Interaction module*.

Additional information

- *What are basic managers?*
- *What are step-dependent managers?*
- *Changing the status of an object in a step*
- *Understanding Interaction module managers and editors*

Interaction editors

To create interactions, select **Interaction->Create** from the main menu bar. A **Create Interaction** dialog box appears in which you can provide a name for the interaction, select the step in which the interaction will be created, and choose the type of the interaction.

When you click **Continue** in the **Create Interaction** dialog box after selecting any interaction type except general contact, you are prompted to select the regions to which to apply the interaction. Once you have selected the region or regions, an interaction editor appears in which you can specify additional information about the interaction, such as the interaction property that you want to associate with the interaction. For general contact interactions, the interaction editor appears when you click **Continue** in the **Create Interaction** dialog box. For example, the general contact editor for Abaqus/Explicit analyses is shown in *Figure 1*.

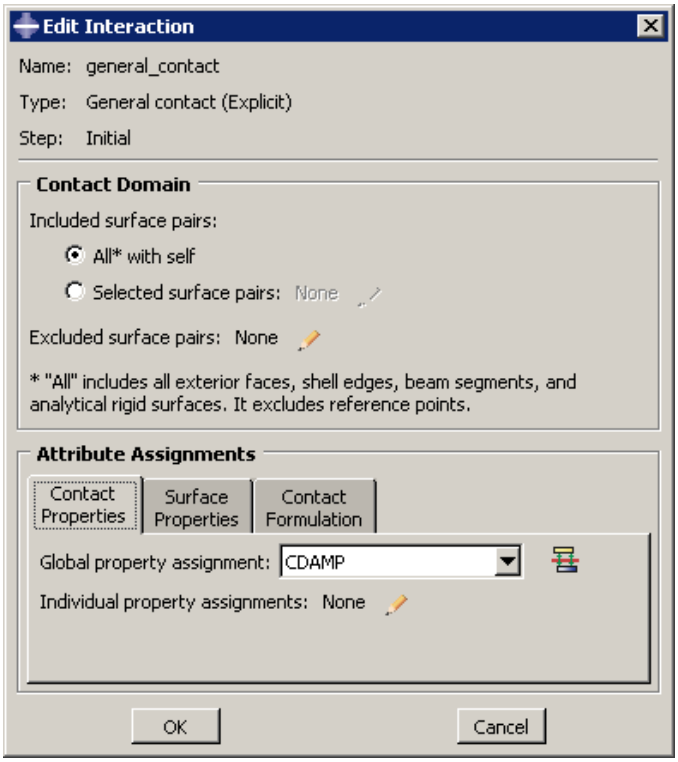


Figure 1: The general contact editor.

Each interaction editor displays the current step and the name and type of the interaction that you are defining in the top panel of the dialog box. The format of the rest of the editor varies depending on the type of interaction you are defining.

Once you have created an interaction, you can modify the interaction in the following ways:

- You can modify some or all of the data that you entered in the editor when you created the interaction.
- You can use the **Interaction Manager** to modify the stepwise history of the interaction. (For more information, see *What are step-dependent managers?*.)

You can display information on a particular editor feature by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest.

Additional information

• *Understanding modified step-dependent objects*

- *Understanding Interaction module managers and editors*

Interaction property editors

To create interaction properties, select **Interaction->Property->Create** from the main menu bar. A **Create Interaction Property** dialog box appears in which you can specify a name for the interaction property and the type of interaction property that you want to create. Once you have specified this information, click **Continue** in the **Create Interaction Property** dialog box to display the interaction property editor.

The format of the interaction property editor depends on the type of interaction property you are defining. For example, the film condition and actuator/sensor property editors display data fields in which you can enter all of the information necessary to define the property. The film condition property editor is shown in *Figure 1*.

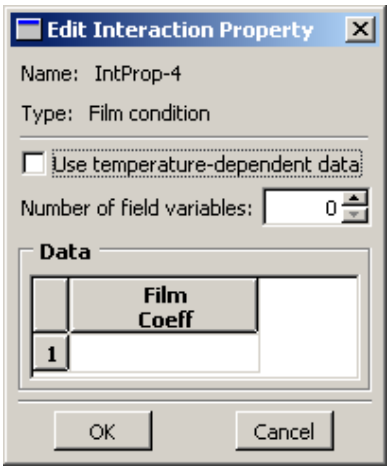


Figure 1: The film condition property editor.

The format of the contact property editor, on the other hand, is identical to the material editor in the Property module (see *Creating materials*, for more information). Like the material editor, the contact property editor contains menus from which you select options to include in the property definition, as shown in *Figure 2*.

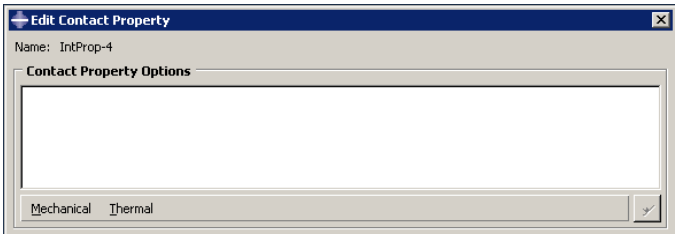


Figure 2: The contact property editor contains Mechanical and Thermal option menus.

When you select an option from a menu, the name of the option appears in the **Contact Property Options** list at the top of the editor, and the option becomes part of your interaction property definition. In addition, the option definition area in the lower half of the editor changes to provide fields in which you can specify information for the currently selected option.

For example, the **Contact Property Options** list in *Figure 3* reflects that the **Tangential Behavior** and **Normal Behavior** options (located in the **Mechanical** menu) have been included in the property definition. **Tangential Behavior** is currently selected, and the related parameters appear in the lower half of the editor. If you want to remove an option from a contact property definition, you can select that option from the **Contact Property Options** list and then click


.

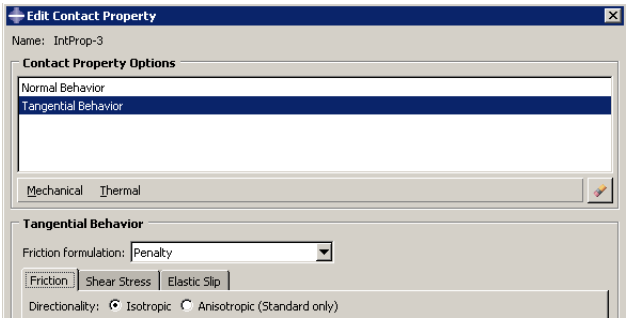


Figure 3: A mechanical contact property definition that includes the Tangential Behavior and Normal Behavior options.

You can display help on a particular feature of the editor by selecting **Help->On Context** from the main menu bar and then clicking the feature of interest. For detailed instructions on creating properties, see *Creating interaction properties*, and *Using the interaction property editors*.

Additional information

- *Managing objects*
- *Understanding modified step-dependent objects*

Contact controls editors

To create contact controls for surface-to-surface contact and self-contact interactions, select **Interaction->Contact Controls->Create** from the main menu bar. A **Create Contact Controls** dialog box appears in which you can specify a name for the contact controls and the type of contact controls that you want to create. Once you have specified this information, click **Continue** to display the contact controls editor.



Warning:

Contact controls are intended for advanced users. The default settings of these controls are appropriate for most analyses. Using nondefault values of these controls may greatly increase the computational time of the analysis or produce inaccurate results. Changing these settings in an Abaqus/Standard analysis may also cause convergence problems.

Each contact controls editor displays the name and type of the contact controls that you are defining in the top panel of the dialog box. The format of the rest of the editor varies depending on whether you are defining controls for an Abaqus/Standard or an Abaqus/Explicit analysis.

You can display help on a particular feature of the editor by selecting **Help->On Context** from the main menu bar and then clicking the feature of interest. For more information, see *Specifying contact controls in an Abaqus/Standard analysis*, and *Specifying contact controls in an Abaqus/Explicit analysis*.

Additional information

• *Customizing contact controls*

Contact initialization editor

To create contact initialization rules for a general contact interaction in Abaqus/Standard, select **Interaction->Contact Initialization->Create** from the main menu bar. The contact initialization editor appears in which you can specify a name for the initialization definition and the rules associated with that definition.

You can display help on the contact initialization editor by selecting **Help->On Context** from the main menu bar and then clicking the editor. For more information, see *Creating contact initializations*.

Additional information

• *Creating contact initializations*

Constraint editors

To create constraints, select **Constraint->Create** from the main menu bar. A **Create Constraint** dialog box appears in which you can specify the name and type of the constraint. Click **Continue** to specify the regions to which to apply the constraint (if applicable) and to display the editor in which you can enter the data necessary to define the constraint.

Each constraint editor displays the name and type of the constraint you are defining in the top panel of the dialog box. The format of the rest of the editor varies depending on the type of constraint you are defining. For example, the tie constraint editor is shown in *Figure 1*.

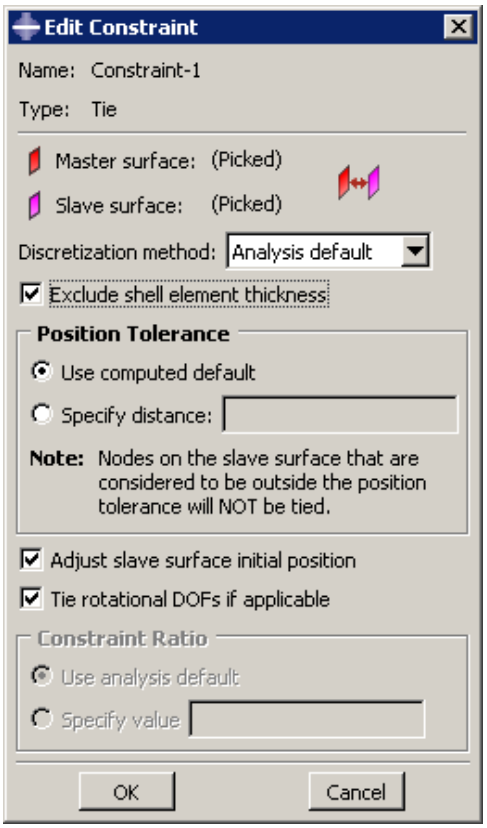


Figure 1: The tie constraint editor.

You can display information on a particular editor feature by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest. For detailed instructions on creating constraints, see *Using the constraint editors*.

Additional information

- *Creating constraints*

Connector section editors

The connector section editor allows you to create connector sections and to add the connector behaviors available in Abaqus/Standard and Abaqus/Explicit.

To create connector sections, select **Connector->Section->Create** from the main menu bar. A **Create Connector Section** dialog box appears in which you can specify a name, category, and type for the connector section that you want to create. When you select a connection type from the **Assembled/Complex** or **Basic** category, the available and constrained components of relative motion (CORM) for that connection type are displayed in the dialog box. In addition, you can click  to see a schematic drawing of the connection type along with the Abaqus idealization of the connection. Once you have specified the name, category, and type, click **Continue** in the **Create Connector Section** dialog box to display the connector section editor. For a connection type in the **MPC** category, select the type and enter data, if required. Click **OK** to finish creating the **MPC** section and to close the **Create Connector Section** dialog box.

The connector section editor allows you to add the connector behaviors available in Abaqus/Standard and Abaqus/Explicit. When you click **Add** on the **Behavior Options** tabbed page, a list of behaviors appears. After you select a behavior, the name of the behavior appears in the **Behavior Options** list at the top of the editor, and the behavior becomes part of your connector section definition. The option definition area in the lower half of the editor changes to provide fields in which you can specify information for the currently selected behavior. If you want to remove a behavior from a connector section definition, you can select it from the **Behavior Options** list and then click **Delete**.



Note:

Abaqus/CAE does not perform any checks for dependencies on other behaviors; you should ensure that all required behaviors are defined. For example, if you define a plasticity behavior, you must also define an elasticity behavior.

You can define multiple behaviors of the same type, such as elasticity. Only forces or moments that are consistent with the available components of relative motion for the chosen connection type can be selected to define behaviors. For example, the **Behavior Options** list in *Figure 1* reflects that two **Elasticity** behaviors and a **Reference Length** behavior have been included in the connector section definition. The highlighted **Elasticity** behavior defines elastic behavior in the same direction as the selected moment.

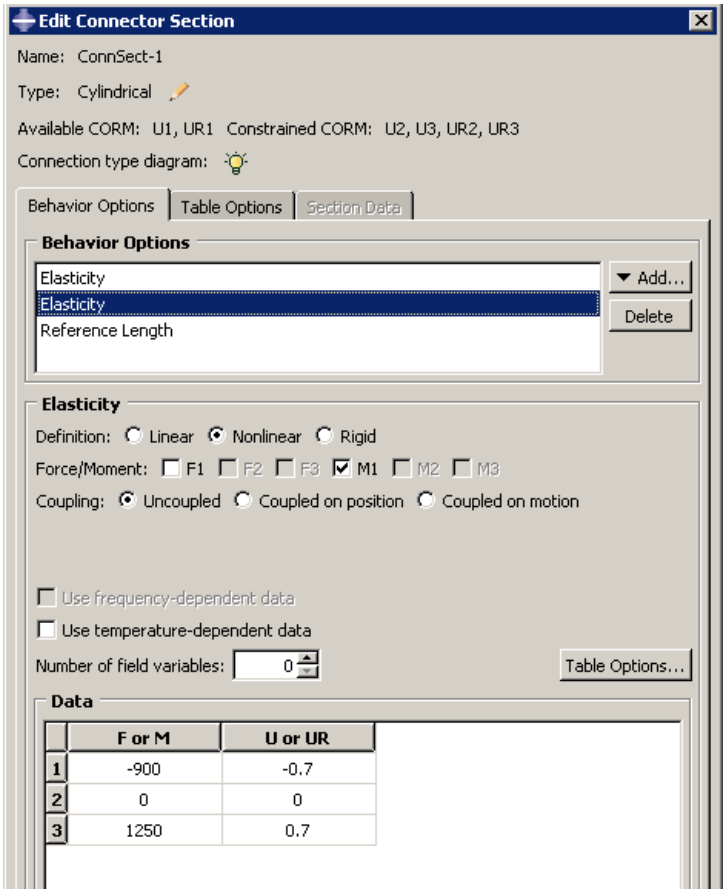


Figure 1: The connector section editor.

On the **Table Options** tabbed page, you can specify behavior settings for the regularization (Abaqus/Explicit analyses only) and the extrapolation of tabular data for all of the behavior options in a connector section. Alternatively, you can specify behavior settings for individual behavior options by clicking the **Table Options** button in the option definition area in the lower half of the editor. Behavior settings for individual behavior options take precedence over the connector section behavior settings.

When section data are applicable for the specified connection type, you can enter the data on the **Section Data** tabbed page.

You can display help on a particular feature of the editor by selecting **Help->On Context** from the main menu bar and then clicking the feature of interest. For detailed instructions on creating connector sections and defining behaviors and section data, see *Creating connector sections*, and *Using the connector section editors*.

Additional information

- *Managing objects*
- *Creating connector sections*
- *Using the connector section editors*
- *Connectors*

Connector section assignment editors

To create connector section assignments, select **Connector->Assignment->Create** from the main menu bar. Select the wires to which you want to assign the connector section.

The connector section assignment editor contains three tabbed pages that allow you to specify the connector section that you want to assign to the wires and the orientations of the endpoints of the wires. For example, the connector section assignment editor that assigns the connector section `Shock_Absorber` to selected wires is shown in *Figure 1*.

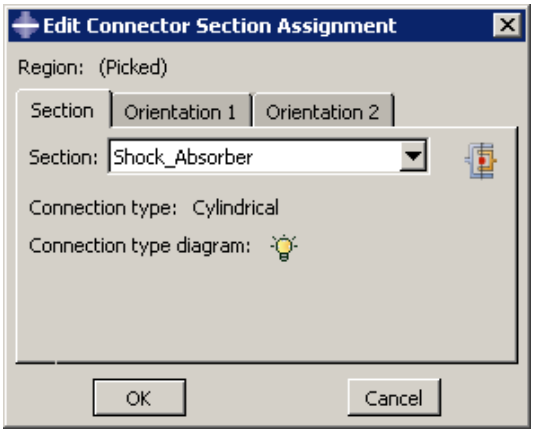


Figure 1: The connector section assignment editor.

You can display information on a particular editor feature by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest. For detailed instructions on creating connector section assignments, see *Creating and modifying connector section assignments*.

Additional information

- *Creating or modifying wire features for multiple connectors*
- *Creating connector sections*
- *Creating and modifying connector section assignments*
- *Connectors*

Understanding symbols that represent interactions, constraints, and connectors

When you apply interactions, constraints, and connectors to regions of the model, you can choose to display symbols in the viewport that indicate where you have applied the interactions, constraints, and connectors. For information about graphical symbol types, see *Symbols used to represent interactions, constraints, and connectors*.

You can apply interactions, constraints, and connectors to geometry or to orphan nodes and elements.

Interactions and constraints

If you apply an interaction or constraint to geometry, symbols appear approximately equally spaced over the surface or surfaces to which the interaction or constraint is applied. If the interaction or constraint definition involves a node region rather than a surface, the symbols appear equally spaced on the edges of the node region and at any vertices in the node region. If the interaction or constraint is applied to a single vertex, a symbol appears at that vertex.

If you apply the interaction or constraint to orphan nodes or elements, symbols appear at the center of each element face for surface-based regions or at the nodes for node-based regions.

For interactions (and prescribed conditions) that use analytical field distributions, the symbols are scaled based on the analytical field value. In addition, a plus sign (+) or a minus sign (−) is displayed inside each symbol to indicate whether the magnitude of the interaction is positive or negative at that location. Abaqus/CAE displays scaled-down symbols for interactions when an analytical field evaluates to zero for a portion of its region. These scaled-down symbols are noticeably smaller than the default symbol size. For examples of these symbols, see *Understanding prescribed condition symbol type, color, and size*. For more information, see *Using analytical expression fields*.

Connectors

When you apply a connector to wires, squares appear at the first points of the wires and triangles appear at the second points of the wires. If you specify an orientation other than the global coordinate system for orientation 1, an orientation triad appears at the first points of the wires. For orientation 2, an orientation triad appears at the second points of the wires only if you specify a coordinate system by name; if you toggle on the option to **Use orientation 1**, the orientation triad does not appear at the second points. The connection type label appears midway along a line between the endpoints of the selected wires. You can also display the tag associated with the connector section assignment, though this display is turned off by default.

For information about controlling the visibility of these symbols, see *Controlling the display of attributes*.

Using the Interaction module toolbox

You can access all the Interaction module tools through either the main menu bar or the Interaction module toolbox. *Figure 1* shows the icons for all the tools in the Interaction module toolbox.

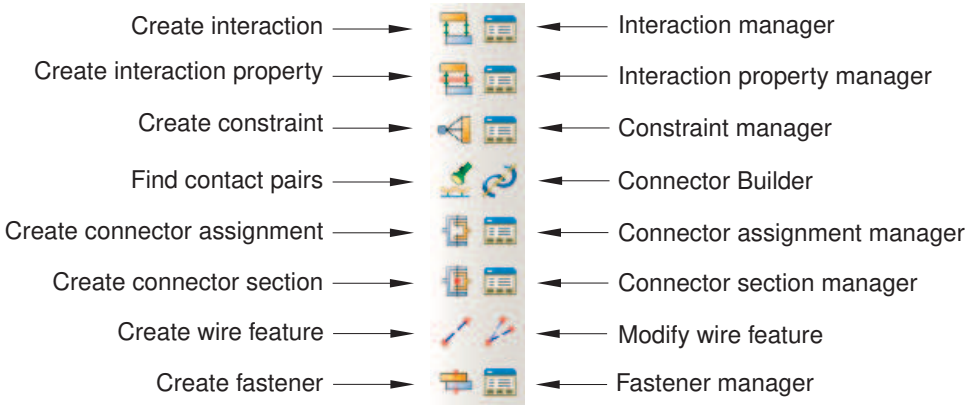


Figure 1: The Interaction module toolbox.

Using the Interaction module

This section describes in detail how to use different features of the Interaction module.

See also *What are step-dependent managers?* for information on managing interactions.

In this section:

- *Creating interactions*
- *Creating interaction properties*
- *Customizing contact controls*
- *Creating contact initializations*
- *Creating contact stabilization definitions*
- *Creating constraints*
- *Selecting a process for defining connector geometry*
- *Creating a single connector*
- *Creating or modifying wire features for multiple connectors*
- *Creating coincident point connectors*
- *Creating connector sections*
- *Creating and modifying connector section assignments*
- *Editing the region to which an interaction or constraint is applied*
- *Using the Special menu in the Interaction module*

Creating interactions

When you create an interaction, you must specify the name of the interaction, the step in which to activate the interaction, the type of interaction, and, if required, the region of the assembly to which you want to apply the interaction.

The available types of interactions depend on the procedure selected for the step. For example, you can define heat flux on a surface only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step. Similarly, you can define interactions with a user-defined actuator/sensor only during the initial step.

If you are creating surface-to-surface contact interactions, you can automate many of the steps in the below procedure by using the contact detection tool. For more information, see *Using contact and constraint detection*.

1. From the main menu bar, select **Interaction->Create**.

   A **Create Interaction** dialog box appears with a default name displayed in the **Name** text field.

   Tip: You can also create an interaction using the  tool in the Interaction module toolbox.

2. Type a name for the interaction. For more information about naming objects, see *Using basic dialog box components*.
3. Select the step in which to activate the interaction. Click the arrow next to the **Step** text field, and select from the list that appears.

   The **Types for Selected Steps** list changes to a list of all of the available interaction types.
4. From the **Types for Selected Steps** list, select the interaction type, and click **Continue**.
5. If required, select the region to which you want to apply the interaction using one of the following methods:

   - Select a region in the viewport. You can use the angle method to select a group of faces or edges from geometry or a group of element faces. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.

     Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** to apply the interaction to a geometry region or to a reference point.
     - Click **Mesh** to apply the interaction to a native or orphan mesh selection.

     By default, for most interactions a set or surface is created that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.
   - To select from a list of existing sets or surfaces, do the following:

     1. Click **Sets** or **Surfaces** on the right side of the prompt area. (The name of the button depends on the type of object you are creating. For example, if you are creating a surface-to-surface contact interaction, a **Surfaces** button appears.)

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets or surfaces.
     2. Select the set or surface of interest, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

The interaction editor appears. The region to which you are applying the interaction is highlighted in the viewport.

6. Enter all of the data necessary to define the interaction, and click **OK**. For detailed information on a particular feature of the editor, select **Help->On Context** from the main menu bar and then click the feature of interest or see *Using the interaction editors*.

   Symbols appear in the viewport that represent the interaction that you just created. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

- *Understanding and using toolboxes and toolbars*
- *What are step-dependent managers?*
- *Selecting objects within the viewport*
- *Using the interaction editors*
- *The Set and Surface toolsets*

Creating interaction properties

You can create an interaction property by entering data in an interaction property editor. The format of the editor varies according to the type of property you are defining; when you create a property, you must first specify the property type so that the appropriate editor appears.

1. From the main menu bar, select **Interaction->Property->Create**.

   Tip: You can also create an interaction property using the  tool in the Interaction module toolbox.

   A **Create Interaction Property** dialog box appears.

2. Enter a property name. For more information about naming objects, see *Using basic dialog box components*.
3. Select the property type, and click **Continue**.

   The editor for the property type you have specified appears.

4. In the editor, enter all of the data necessary to define the interaction property.

   Note:

   You can display help on a particular editor feature by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest.

5. Click **OK** to save the data and to exit the editor.

Additional information

- *Understanding interaction properties*
- *Interaction property editors*
- *Using the interaction property editors*

Customizing contact controls

The contact controls editors allow you to modify the algorithms used to enforce contact conditions. The default contact controls are usually sufficient, but customizing these controls may result in a more cost-effective solution.



Warning:

Contact controls are intended for advanced users. The default settings of these controls are appropriate for most analyses. Using nondefault values of these controls may greatly increase the computational time of the analysis or produce inaccurate results. Changing these settings in an Abaqus/Standard analysis may also cause convergence problems.

1. From the main menu bar, select **Interaction->Contact Controls->Create**.
2. In the **Create Contact Controls** dialog box that appears, do the following:
   - Name the contact controls.
   - Select **Abaqus/Standard contact controls** or **Abaqus/Explicit contact controls**.
3. Click **Continue**.

   The editor for the type of contact controls that you specified appears.
4. In the editor, enter the data necessary to customize the contact controls. For detailed instructions, see the following sections:
   - *Specifying contact controls in an Abaqus/Standard analysis*
   - *Specifying contact controls in an Abaqus/Explicit analysis*
5. If you want to reset the values in the editor to the default values, click **Defaults** at the bottom of the editor.
6. Click **OK** to save your customized settings and to exit the editor.

Creating contact initializations

By default, Abaqus adjusts the position of slightly overclosed surfaces in a general contact domain as discussed in *Contact Initialization for General Contact in Abaqus/Standard* and *Contact Initialization for General Contact in Abaqus/Explicit*. Contact initializations are used to modify the default behavior for contact surface adjustments. Each contact initialization definition contains a set of adjustment rules; the general contact definition specifies the surfaces to which each initialization definition is applied (see *Specifying and modifying contact initialization assignments for general contact*).

Contact initializations are intended to correct small gaps or overclosures between surfaces. Specifying large initialization adjustments can lead to mesh distortion and increased computational cost for an analysis.

1. From the main menu bar, select **Interaction->Contact Initialization->Create**.

   The **Create Contact Initialization** dialog box appears.

2. Enter a **Name** for the contact initialization definition, and select a **Type** for the contact initialization.

   The **Edit Contact Initializations** dialog box appears.

3. You can specify the following techniques for treating **Initial Overclosures** between two surfaces:

   - Select **Resolve with strain-free adjustments** to adjust certain surfaces to be exactly touching at the beginning of the analysis without creating strain in the model. Only portions of surfaces that lie within the specified distance range are adjusted.
   - Select **Treat as interference fits** to resolve surface overclosures gradually over the course of the first step in the analysis; this technique creates strain in the model as the surfaces are displaced. Only portions of surfaces that lie within the specified overclosure distance range are adjusted using the interference fit.

     - To establish a uniform overclosure prior to resolving the interference fit, toggle on **Specify interference distance**, and enter a value for the overclosure distance. Portions of surfaces that lie within the specified distance range (both overclosures and openings) are adjusted to be overclosed by the specified amount. The adjustments occur at the beginning of the analysis without creating strain in the model; the subsequent interference fit resolution during the first step in the analysis will create strain in the model.

   - Select **Specify clearance distance** to adjust certain surfaces to be separated by a specified value at the beginning of the analysis without creating strain in the model. Only portions of surfaces that lie within the specified distance range are adjusted.

4. Specify an overclosure distance range; nodes that are overclosed by a value less than the specified distance are adjusted using either strain-free adjustments or a gradual interference fit.

   - Select **Analysis default** to let Abaqus calculate a maximum overclosure adjustment distance based on the size of the underlying element facets on each surface.
   - Select **Specify value** to enter a maximum overclosure adjustment distance directly. If you enter a value that is smaller than the calculated analysis default for a surface, Abaqus uses the analysis default value for that surface.

   Warning:

   If two surfaces (or portions of two surfaces) are initially overclosed by a distance greater than the specified adjustment value, contact between these severely overclosed surface regions is not enforced for the duration of the analysis; see *Contact Initialization for General Contact in Abaqus/Standard*, for more information.

5. Specify an opening distance range; nodes separated from the opposing surface by a value less than the specified distance are adjusted using strain-free adjustments.
   - Select **Analysis default** to ignore all open nodes during the initialization adjustments.
   - Select **Specify value** to enter a maximum opening adjustment distance directly.
6. Specify other options.
   - Select **Adjust nodal coordinates** to resolve clearances/overclosures by adjusting the nodal coordinates without creating strain in the model. This option applies only to Abaqus/Explicit analyses and can be used only for clearances/overclosures defined in the first step of an analysis.
   - Select a **Secondary node set for clearance** to include in the initial clearance specification. The specified clearance is enforced at all secondary nodes in this node set irrespective of whether they are above or below their respective main surfaces. This option applies only to Abaqus/Explicit analyses when the clearance distance is specified. It cannot be used if either overclosure or opening distance is specified.
   - Select the **Step fraction for interference** value to define the fraction of the step time (between 0.0 and 1.0) in which the interference fit has to be solved. The default value is 1.0. This option applies only to Abaqus/Explicit analyses when the interference distance is specified.
7. If you want to reset the values in the dialog box to the default values, click **Defaults** at the bottom of the dialog box.
8. Click **OK** to save your contact initialization definition and to close the **Edit Contact Initialization** dialog box.

Additional information

- *Contact Initialization for General Contact in Abaqus/Standard*
- *Contact Initialization for General Contact in Abaqus/Explicit*
- *Defining general contact*
- *Specifying and modifying contact initialization assignments for general contact*

Creating contact stabilization definitions

Contact stabilization introduces viscous damping to oppose incremental relative motion between two surfaces. This damping can be used to stabilize unconstrained rigid body motion prior to contact closure without degrading the accuracy of results. Contact stabilization is based on a set of factors described in *Stabilization for General Contact in Abaqus/Standard*.

Each contact stabilization definition contains a set of stabilization factors; the general contact definition specifies the surfaces to which each stabilization definition is applied (see *Specifying and modifying contact stabilization assignments for general contact*).

1. From the main menu bar, select **Interaction->Contact Stabilization->Create**.

   The **Edit Contact Stabilization** dialog box appears.
2. Enter a **Name** for the contact stabilization definition.
3. Specify the stabilization type:
   - Select **Define new stabilization behavior** to define a standard stabilization.
   - Select **Reset values from previous steps** to define a special type of stabilization that cancels the effects of stabilization that was applied in previous analysis steps; the canceling stabilization must still be assigned to surfaces in the general contact definition. This type of stabilization does not require any additional data.
4. Specify a **Zero stabilization distance**; no stabilization is applied to surfaces separated by a distance greater than this value. A gap-dependent scale factor varies during the analysis between one (when the surfaces are touching) and zero (when the gap between surfaces exceeds the specified zero stabilization distance).
   - Select **Analysis default** to set the gap distance equal to a characteristic surface dimension.
   - Select **Specify** to enter the gap distance directly.
5. Specify a **Reduction factor** to determine how the damping value changes in successive increments. A value less than one causes the damping to decrease with each increment; a value greater than one (not recommended) causes the damping to increase with each increment.
6. Specify a **Scale factor** that is applied to stabilization damping effects in the normal direction.
7. Specify a **Tangential factor** that is applied to stabilization damping effects in the tangential direction.
8. If desired, select an **Amplitude** envelope to vary the stabilization over the course of the step.

   Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Selecting an amplitude other than the default ramp amplitude may cause the stabilization effects to span multiple analysis steps; see *Stabilization for General Contact in Abaqus/Standard* for details.
9. If you want to reset the values in the dialog box to the default values, click **Defaults** at the bottom of the dialog box.
10. Click **OK** to save your contact stabilization definition and to close the **Edit Contact Stabilization** dialog box.

Additional information

- *Stabilization for General Contact in Abaqus/Standard*
- *Defining general contact*
- *Specifying and modifying contact stabilization assignments for general contact*

Creating constraints

You can create the following constraints:

- Tie constraints that tie two separate surfaces together so that there is no relative motion between them.
- Rigid body constraints that allow you to designate a collection of regions as a rigid body.
- Display body constraints that allow you to designate a part instance that will be used for display only.
- Coupling constraints that allow you to constrain the motion of a surface to the motion of a reference node.
- Adjust points constraints that allow you to move a point or points onto a specified surface.
- Multi-point constraints that allow you to constrain the motion of the secondary nodes of a region to the motion of a single point.
- Shell-to-solid coupling constraints that allow you to couple the motion of a shell edge to the motion of an adjacent solid face.
- Embedded region constraints that allow you to embed a region of the model within a "host" region of the model or within the whole model.
- Equation constraints that describe linear constraints between individual degrees of freedom.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a constraint using the  tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. Select the desired constraint type.
3. Click **Continue** to create the constraint and to close the **Create Constraint** dialog box.
4. If applicable, select the region to which to apply the constraint. For more information, see *Selecting objects within the viewport*."
5. In the editor that appears, enter any data necessary to define the constraint.

For detailed instructions on creating different types of constraints, see the following sections:

- *Defining tie constraints*
- *Defining rigid body constraints*
- *Defining display body constraints*
- *Defining coupling constraints*
- *Defining adjust points constraints*
- *Defining MPC constraints*
- *Defining shell-to-solid coupling constraints*
- *Defining embedded region constraints*
- *Defining equation constraints*

Additional information

• *Understanding and using toolboxes and toolbars*

- *Suppressing and resuming objects*
- *Selecting objects within the viewport*
- *The Set and Surface toolsets*

Selecting a process for defining connector geometry

You must create assembly-level wire features to define the underlying geometry for modeling connectors. The wire feature contains wires connecting points from the assembly in the current viewport or connecting points from the assembly to ground.

Abaqus/CAE provides two methods that you can use to create connector geometry for an assembly:

- The **Connector Builder** enables you to perform all the steps involved in modeling a connector: create a single assembly-level wire feature, optionally create reference points at its endpoints, assign it a connector section, and specify the orientations for either of its endpoints. You should use this dialog box to add a small number of wire features. See *Creating a single connector*, for instructions on using the **Connector Builder**.
- The **Create Wire Feature** dialog box enables you to create multiple wires within a single wire feature. You should use this dialog box to define a large number of similar connectors. This dialog box creates the wire feature only; you must use other dialog boxes in the Interaction module to perform the subsequent modeling steps like creating reference points and datum coordinate systems, assigning connector sections to the wires, and specifying the orientations of the wires' endpoints. See *Creating or modifying wire features for multiple connectors*, for instructions on using the **Create Wire Feature** dialog box.

Creating a single connector

The **Connector Builder** enables you to perform all of the steps involved in modeling a single connector. These steps include:

- Creating a wire feature between two points or between a point and ground.
- Creating reference points at either endpoint, and creating a datum coordinate system along the length of the wire.
- Creating a connector section assignment for the new wire feature.
- Specifying orientations for either endpoint that the wire feature connects.

Because the **Connector Builder** creates one connector at a time, you should use it to add a small number of connectors to your assembly. If you plan to define a large number of connectors, begin by adding wire features to the assembly in the **Create Wire Feature** dialog box. See *Creating or modifying wire features for multiple connectors*, for more information.

1. From the main menu bar, select **Connector->Connector Builder**.

   Tip: You can also start the **Connector Builder** using the  tool in the Interaction module toolbox.

2. Select the points that you want to connect.
   a. Select the first point and second point for the connector in the viewport.

      Note:

      You can also click **Connect to Ground** in the prompt area to ground one side of the connector. You cannot connect both points to ground.

      The **Connector Builder** opens with the points you selected in the **Endpoints** portion of the dialog box. The first point is highlighted in red and the second point is highlighted in magenta in the viewport.

   b. If desired, click  to switch the two endpoints you selected.
   c. If desired, toggle on **Create a reference point** under either point description to create a reference point at that endpoint when you save the wire feature. If the selected point is a datum point or an interesting point, Abaqus/CAE automatically creates a reference point at that location.

3. Select a connector section. If desired, click  to create a new connector section.

   The connection type is displayed in the editor. For basic, assembled, and complex connection types, you can click  to see a schematic drawing of the connection type along with the Abaqus idealization of the connection. For MPC connection types, the motion of the second point is constrained to the motion of the first point. The connection type you select also dictates the initial values selected for the connector orientations. If you change the connector section assignment, Abaqus/CAE resets these orientation settings to the default values for that connection type.

4. If desired, modify any of the following orientation settings:
   a. Project a datum coordinate system along the axis between the endpoints.

      In the **CSYS 1** portion of the dialog box, select the **Create CSYS on axis between points** option, then select **1**, **2**, or **3** as the **Axis** for the new datum coordinate system. Abaqus/CAE may select this option by default for some connection types.

b. Specify an orientation other than the global coordinate system for the first point for the connector in the **CSYS 1** portion of the dialog box. If the connector is not connected to ground and the selected connector section assignment allows you to adjust the second point in the connector, you can specify an orientation for the second point in the **CSYS 2** portion of the dialog box as well.

In either portion of the dialog box, select **Specify CSYS**, click , then use one of the following methods to select the datum coordinate system:

- Select a predefined datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.
- Select a predefined coordinate system in the viewport.
- For the second point in the connector, select **Use CSYS 1** to use the coordinate system defined for the first point at the second point in the connector as well.

The selected orientation is highlighted red in the viewport. The **Connector Builder** reappears and displays a description of the coordinate system used for the orientation for the selected point of the connector.



Tip: You can also create a new datum coordinate system to use by clicking  from the **CSYS 1** or **CSYS 2** portion of the dialog box.

c. If desired, you can specify for either endpoint an additional rotation about an axis of the datum coordinate system. Select the **Specify additional rotation** option for one of the points, enter a value for the rotation angle in degrees, and select **1**, **2**, or **3** for the **About axis** option. The **Specify additional rotation** option is available only if you specify a datum coordinate system other than the global coordinate system.

5. Click **OK** to save your connector and to close the **Connector Builder**, or click **OK/Repeat** to save the assembly-level wire, leaving the dialog box open to create a new wire.

Additional information

• *Connectors*

Creating or modifying wire features for multiple connectors

In the Interaction module, select **Connector->Geometry->Create Wire Feature** from the main menu bar to add one or more wire features. You can add disjoint wires, chained wires, or wires that are connected to ground. When you create the wire feature, you can create a geometry set that includes all of the wires in the wire feature to use during subsequent selection procedures; for example, when you select wires for the connector section assignment definition or to apply connector loads. For detailed instructions on creating assembly-level wires features, see *Adding a point-to-point wire feature*.

You can modify assembly-level wire features by selecting **Connector->Geometry->Modify Wire Feature** from the main menu bar. You select any wire from the feature that you want to modify and make changes in the **Modify Wire Feature** dialog box, as described in the procedure below. When you click **OK** in the dialog box, Abaqus/CAE does the following:

- renames and suppresses the original wire feature;
- creates a new wire feature using the same name as the original wire feature;
- renames the geometry set that had contained the original wire feature, if it existed; and
- if applicable, creates a new geometry set containing the modified wire feature.

For example, *Figure 1* shows that `Wire-1` is renamed `OldWire-1-1` and suppressed. Similarly, `Wire-1-Set-1` is renamed `OldWire-1-Set-1`.

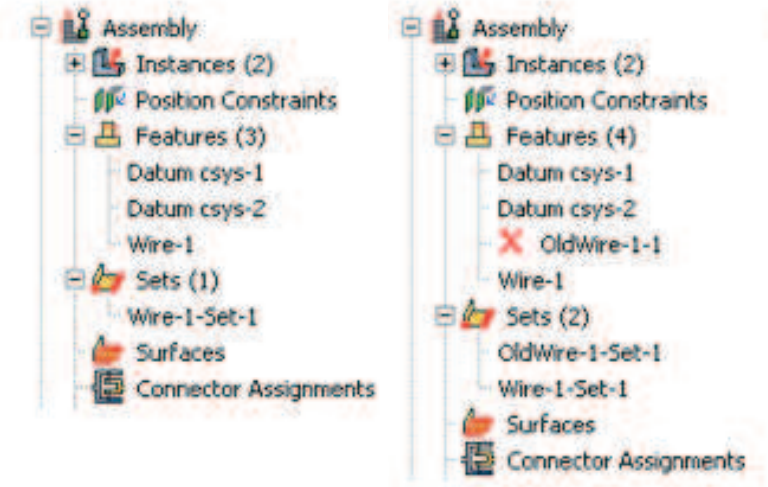


Figure 1: Model Tree showing the Features and Sets containers for the original wire feature (left) and the modified wire feature (right).

By using the default geometry set name created for a wire feature when you select wires to define connector section assignments, connector loads, and connector boundary conditions, you ensure that these objects remain valid if you modify the wire feature. Otherwise, objects that reference any part of the original wire feature are invalidated if you modify the wire feature; for example, if you define a connector force and select wires from the viewport or use a geometry set with a different name.

You can remove wires from the wire feature by selecting **Connector->Geometry->Remove Wires From Feature** from the main menu bar. The operation to remove wire edges from the feature is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar in the Interaction module, select **Connector->Geometry->Modify Wire Feature**.

Tip: You can also modify an assembly-level wire feature using the  tool in the Interaction module toolbox.

2. In the viewport, select any wire of the feature that you want to modify.

   The **Modify Wire Feature** dialog box appears and displays the point pairs that define the assembly-level wires in the original wire feature.

3. In the **Point Pairs** portion of the dialog box, you can do the following:

   - To add more point pairs, specify the method, click , and select the points from the viewport. The representation of the added wire is highlighted in magenta.
   - To edit a point, select the point in the table, click , and reselect a point. The selection highlighting in the viewport is updated to show the newly edited point.
   - To identify a specific point pair in the viewport, select the desired row. The line connecting the selected point pair is highlighted in red.
   - To remove a point pair, select the desired row and click .
   - To exchange the entries for **Point 1** and **Point 2** in a point pair, select the desired row and click .

4. In the **Set Creation** portion of the dialog box, toggle on **Create set of wires** if you want Abaqus/CAE to create a new geometry set of wires for the modified wire feature.
5. Click **OK** to modify the wire feature.

   The original wire feature is renamed and suppressed. The modified assembly-level wire feature appears in the Model Tree in the **Features** container under the assembly using the same name as the original wire feature.

Additional information

- *Adding a point-to-point wire feature*
- *Connectors*

Creating coincident point connectors

In the Interaction module, select **Connector->Coincident Builder** from the main menu bar to create a connector wire feature and section assignment for a group of coincident points. For detailed instructions on creating assembly-level wires features, see *Adding a point-to-point wire feature*.

1. From the main menu bar in the Interaction module, select **Connector->Coincident Builder**.
2. In the viewport, select the coincident points that you want to include in the connector wire feature, then click **Done** in the prompt area. For an overview of the methods you can use to select multiple points from the viewpoint, see *Drag-selecting multiple objects*.

   The **Coincident Point Builder** dialog box opens with the selected coincident points displayed in the **Endpoints** portion of the dialog box.
3. If desired, customize the content and order of the endpoints you want to use by doing the following:

   - Highlight the row for any coincident point, and click  to delete that coincident point pair.
   - Highlight the row for any coincident point, and click  to swap the first and second points in the coincident point pair.
4. Select a connector section. If desired, click  to create a new connector section.

   The connection type is displayed in the editor. For basic, assembled, and complex connection types, you can click  to see a schematic drawing of the connection type along with the Abaqus idealization of the connection. For MPC connection types the motion of the second point is constrained to the motion of the first point.
5. If desired, modify any of the following orientation settings:

   a. Specify an orientation other than the global coordinate system for the coincident point connector wire feature in the **CSYS 1** portion of the dialog box. If the selected connector section assignment allows you to adjust the second point in the connector, you can specify an orientation for the second point in the **CSYS 2** portion of the dialog box as well.

   In either portion of the dialog box, click , then use one of the following methods to select the datum coordinate system:

   - Select a predefined datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.
   - Select a predefined coordinate system in the viewport.
   - For the second point in the connector, select **Use CSYS 1** to use the coordinate system defined for the first point at the second point in the connector as well.

   The selected orientation is highlighted red in the viewport. The **Coincident Point Builder** reappears and displays a description of the coordinate system used for the orientation for the selected point of the connector.

   Tip: You can also create a new datum coordinate system by clicking  from the **CSYS 1** or **CSYS 2** portion of the dialog box.

b. If desired, you can specify an additional rotation about an axis of the datum coordinate system for either endpoint. Select the **Additional rotation angle** option for one of the points, enter a value for the rotation angle in degrees, and select **1**, **2**, or **3** for the **About axis** option. The **Additional rotation angle** option is available only if you specify a datum coordinate system other than the global coordinate system.

6. Click **OK** to create the wire feature and the connector section assignment. Abaqus/CAE also creates a set containing the assembly-level wires.

The original wire feature is renamed and suppressed. The modified assembly-level wire feature appears in the Model Tree in the **Features** container under the assembly using the same name as the original wire feature.

Additional information

• *Constraining two instances with coincident points*

Creating connector sections

You can create a connector section by selecting connection types to define the connector function.

In this section:

- *Creating connector sections for assembled, complex, and basic connection types*
- *Creating connector sections for MPC types*

Creating connector sections for assembled, complex, and basic connection types

The connector section editor allows you to specify assembled, complex, or basic connection types; connector behaviors; and section data to include in the section definition. For more information, see *Connector Elements*.

1. From the main menu bar, select **Connector->Section->Create**.

   Tip: You can also create a connector section using the  tool in the Interaction module toolbox.

   A **Create Connector Section** dialog box appears.

2. Enter a section name. For more information about naming objects, see *Using basic dialog box components*.
3. Choose one of the following connection categories:

   - Choose **Assembled/Complex** to use either predefined combinations of basic connection types or complex connection types.

     Click the arrow next to the **Assembled/Complex type** text field, and select the desired connection type from the list that appears.
   - Choose **Basic** to use translational and rotational connection types.

     1. If desired, click the arrow next to the **Translational type** text field, and select the connection type from the list that appears.
     2. If desired, click the arrow next to the **Rotational type** text field, and select the connection type from the list that appears.

     You can select one translational type, one rotational type, or one translational and one rotational type to define the connector section.

   Abaqus/CAE displays the available and constrained components of relative motion (CORM) for the connection type that you have selected. In addition, you can click  to see a schematic drawing of the connection type along with the Abaqus idealization of the connection.

   For a description of assembled, complex, and basic connection types, see *Connection Types*.
4. Click **Continue**.

   The connector section editor appears.

   Note:

   You can display help on a particular editor feature by selecting **Help->On Context** from the main menu bar and then clicking the editor feature of interest. For more information on editor features, see *Using the connector section editors*.

5. To edit the connection type, click  to the right of the connection type to display the connector section type editor. Select the connection type as described above. You must delete all behaviors from the editor before you can edit the connection type.
6. On the **Behavior Options** tabbed page, you can add, delete, and change behaviors as follows:

Adding behaviors

Click **Add** on the right side of the editor to display a list of available behaviors. Select the behaviors needed to define your connector section. You can define multiple behaviors of the same type for some behaviors. When you select a behavior, its name appears in the **Behavior Options** list, and data fields associated with the behavior appear in the data area in the bottom half of the editor. Use the data fields to enter information for the currently selected behavior. For more information, see *Using the connector section editors*.

Deleting behaviors

In the **Behavior Options** list, select the behavior that you want to delete, and click **Delete** on the right side of the editor. This procedure removes the behavior from both the behavior options list and the connector section definition.

Changing behavior data

In the **Behavior Options** list, select the behavior whose data you want to change. When the data fields associated with the behavior appear in the bottom half of the window, change the information that you have entered for the behavior as desired.

7. On the **Table Options** tabbed page, you can specify behavior settings for the regularization (Abaqus/Explicit analyses only) and the extrapolation of tabular data for all of the behavior options in a connector section. Alternatively, you can specify behavior settings for individual behavior options. The **Table Options** button is available on the **Behavior Options** tabbed page for selected behavior options. Behavior settings for individual behavior options take precedence over the connector section behavior settings. For more information, see *Specifying behavior settings for tabular data*, and *Defining Connector Behavior Using Tabular Data*.

   Specify behavior settings on the **Table Options** tabbed page as follows:

   a. In the **Regularization** portion of the page, specify the settings for the regularization of tabular data in an Abaqus/Explicit analysis. By default, Abaqus/Explicit regularizes the data into tables that are defined in terms of even intervals of the independent variables.

      - Toggle on **Regularize data** to regularize tabular data.
      - Toggle off **Regularize data** to turn off regularization of the tabular data and use the data that you define directly.

   b. If you want to regularize tabular data, specify the error tolerance.

      - Choose **Use default** to use the default value of 0.03.
      - Choose **Specify**, and enter a value for the error tolerance.

   c. In the **Extrapolation** portion of the page, specify the method for the extrapolation of tabular data. The data points that you enter make up a nonlinear curve in the constitutive space. By default, Abaqus extrapolates the dependent variables as constant values that correspond to the end points of the curve outside the specified range of the independent variables.

      - Choose **Constant** to use constant extrapolation of the dependent variables outside the specified range of the independent variables.
      - Choose **Linear** to use linear extrapolation of the dependent variables outside the specified range of the independent variables.

8. The **Section Data** tabbed page becomes available for the following connection types:

   - For the **Flow-Converter** or **Retractor** connection type, enter the following section data:

     ### Node b material flow scaling factor

     Enter the scaling factor, β_s, associated with the material flow at node *b*. Node *b* refers to the second point of a wire used to model a connector in Abaqus/CAE. The default value is 1.

     For more information, see *FLOW-CONVERTER* and *RETRACTOR*.

   - For the **Slip Ring** connection type, enter the following section data:

     ### Mass per unit reference length

     Enter the mass per unit reference length of belt material.

     ### Contact angle around node b

     The contact angle refers to the angle made by the belt wrapping around node *b*. Node *b* refers to the second point of a wire used to model a connector in Abaqus/CAE.

     For an Abaqus/Standard analysis you must specify the contact angle directly. For an Abaqus/Explicit analysis you can specify the contact angle directly, or you can let Abaqus calculate the contact angle based on the configuration of your model:

     - To let Abaqus calculate the contact angle, select **Compute**.
     - To specify the contact angle directly, select **Specify** and enter the contact angle in degrees.

     For more information, see *SLIPRING*.

9. Click **OK** to save the data and to exit the editor.

Additional information

- *Understanding connector sections and functions*
- *Connector section editors*
- *Using the connector section editors*
- *Connectors*

Creating connector sections for MPC types

The connector section editor allows you to specify MPC connection types. For more information, see *General Multi-Point Constraints*.

1. From the main menu bar, select **Connector->Section->Create**.

   Tip: You can also create a connector section using the tool in the Interaction module toolbox.

   A **Create Connector Section** dialog box appears.

2. Enter a section name. For more information about naming objects, see *Using basic dialog box components*.
3. Choose the **MPC** connection category.
4. Click the arrow next to the **MPC type** text field, and select the desired connection type from the list that appears.

   For more information, see the following sections:

   - *General Multi-Point Constraints*
   - *MPC*

5. Click **OK** to save the section definition and to close the dialog box.

Additional information

- *Understanding connector sections and functions*
- *Connector section editors*

Creating and modifying connector section assignments

You create connector section assignments by assigning connector sections to wire features (to define connectors) or attachment lines (to define discrete fasteners) and specifying local orientations associated with the endpoints of the wires or attachment lines. Depending upon the connection type, a local orientation for the first points of the selected wires or attachment lines is required, optional, or not applicable. Depending upon the connection type, a local orientation for the second points of the selected wires or attachment lines is optional or not applicable. For the local orientation requirements for each connection type, see *Connection-Type Library*. *Table 1* relates the numerical coordinate axes referred to in the connection-type library to the coordinate system axis labels used in Abaqus/CAE. You can use the Query toolset to obtain connector assignment information for selected wires or attachment lines.

Table 1: Coordinate system axis labels used for displaying local orientations associated with the endpoints of the wires or attachment lines.

Local coordinate direction	Cartesian	Cylindrical	Spherical
1	x	r	r
2	y	t	t
3	z	z	p

When you create connector section assignments, Abaqus/CAE generates identification strings to associate the assignments with connector orientations. These strings, referred to as “tags,” cannot be modified, and they are displayed in the connector section assignment manager, the Model Tree tooltips, and—optionally—the viewport (see *Controlling the display of attributes*).

1. Display the connector section assignment editor using one of the following methods:
   - To create a new connector section assignment, do the following:
     1. Select **Connector->Assignment->Create** from the main menu bar.

        Tip: You can also create a connector section assignment using the  tool in the Interaction module toolbox.

     2. Select the region (wires or attachment lines) to which you want to apply the connector section.

        Only wire features or attachment lines that have been created at the assembly level are available for selection. The best approach for selecting the region is to use the default geometry set name for the wire feature (see *Creating or modifying wire features for multiple connectors*, for more information) or attachment lines (see *Creating attachment lines by projecting points*, for more information). A single wire or attachment line feature may contain multiple wires or attachment lines. You can select individual wires or attachment lines of the feature; however, only one connector section can be associated with each individual wire or attachment line.

        Use one of the following methods to select the region:

        - Select from a list of existing sets. Click **Sets** on the right side of the prompt area, select the set of interest from the **Region Selection** dialog box that appears, and click **Continue**.
        - Select the wires or attachment lines in the viewport.

     3. Click **Done** in the prompt area.

        The connector section assignment editor appears.

- To edit an existing connector section assignment, select **Connector->Assignment->Manager** from the main menu bar to display the **Connector Section Assignment Manager**. Select the row of data that you want to change, and click **Edit**.

  The connector section assignment editor appears, and the connector symbols become highlighted in the current viewport.

  If you want to select different wires or attachment lines for the connector section assignment, click  in the **Region** portion of the editor and reselect the region.

2. On the **Section** tabbed page, select a connector section. If desired, click  to create a new connector section.

   The connection type is displayed in the editor; if applicable, you can click  to see a schematic drawing of the connection type along with the Abaqus idealization of the connection.

3. On the **Orientation 1** tabbed page, you specify the orientation for the first points of the selected wires or attachment lines if the orientation is required or optional. If the orientation is not applicable, the **Orientation 1** tabbed page is unavailable in the editor.

   a. If you want to specify an orientation other than the global coordinate system, click  and use one of the following methods to select the datum coordinate system:

   - Select a predefined datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.
   - Select a predefined coordinate system in the viewport.

   The selected orientation is highlighted red in the viewport. The connector section assignment editor reappears and displays a description of the coordinate system used for the orientation of the first points of the selected wires or attachment lines.

   b. If applicable, you can specify an additional rotation about an axis of the datum coordinate system. Select the **Additional rotation angle** option, enter a value for the rotation angle in degrees, and select **1**, **2**, or **3** for the **About axis** option. The **Additional rotation angle** option is available only if you specify a datum coordinate system other than the global coordinate system.

4. If the orientation for the second points of the selected wires or attachment lines is optional, Abaqus/CAE uses the orientation for the first points (including the additional rotation angle, if specified) as the default selection. You can specify a different coordinate system for the second points of the selected wires or attachment lines on the **Orientation 2** tabbed page. If the orientation is not applicable, the **Orientation 2** tabbed page is unavailable in the editor.

   a. On the **Orientation 2** tabbed page, toggle on **No modifications to CSYS** and click . Use one of the following methods to select the datum coordinate system:

   - Select a predefined datum coordinate system by name. Click **Datum CSYS List** from the prompt area, select a name from the list, and click **OK**.
   - Select a predefined coordinate system in the viewport.

   The selected orientation is highlighted red in the viewport. The connector section assignment editor reappears and displays a description of the coordinate system used for the orientation of the second points of the selected wires or attachment lines.

   b. If applicable, you can specify an additional rotation about an axis of the datum coordinate system. Select the **Additional rotation angle** option, enter a value for the rotation angle in degrees, and

select **1**, **2**, or **3** for the **About axis** option. The **Additional rotation angle** option is available only if you specify a datum coordinate system other than the global coordinate system.

5. Click **OK** to save your connector section assignment and to close the editor.

   Symbols appear in the viewport that represent the connector section assignment that you just created. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

- *Understanding connectors*
- *Connector section assignment editors*
- *Using the Query toolset to obtain connector assignment information*
- *Fasteners*
- *Connectors*

Editing the region to which an interaction or constraint is applied

You can edit the region to which an interaction is applied only in the step in which the interaction was created.

1. From the **Interaction** or **Constraint** menu in the main menu bar, select **Manager** to display the **Interaction Manager** or **Constraint Manager**.
2. Choose one of the following methods:

   - To edit an interaction, click the cell located in the row of the interaction that you want to modify and in the column of the step in which it was created and click **Edit**. Alternatively, you can just double-click the cell.

     Tip: You can also initiate this procedure by clicking the step in which the interaction was created from the **Step** list located in the context bar. From the **Interaction** menu in the main menu bar, select **Edit->*interaction name***.

   - To edit a constraint, select the name of the constraint and click **Edit**.

     Tip: You can also initiate this procedure by selecting **Edit->*constraint name*** from the main menu bar.

   If you are editing a foundation interaction, either you are prompted to edit the region by selecting or unselecting objects in the viewport or the **Region Selection** dialog box appears in which you can select a surface that you have already created using the Surface toolset.

   If you are editing any other type of interaction or a constraint, the appropriate editor appears. The editor for all interactions except general contact contains an **Edit Region** option for each region involved in the interaction or constraint definition.

   If your interaction or constraint definition includes both a main surface and a secondary surface or region, the main surface appears highlighted in red and the secondary surface or region appears highlighted in magenta during the editing procedure.

   If you are editing a surface-to-surface contact interaction, yellow arrows are displayed on the shell surfaces to show the shell normal that was selected when the interaction was created. On geometry, a single arrow is displayed at the centroid of each face on the shell surface. On native or orphan meshes, an arrow is displayed on each of the element faces on the shell surface. If you edit the main surface or the secondary surface, the arrows are displayed for the selected surface.

3. If you are editing a general contact interaction, modify the contact domain using the methods described in *Defining general contact*.

4. If you are editing a constraint or an interaction other than general contact or foundation, click  for the region that you want to modify. For example, if you are editing a surface-to-surface contact interaction and you want to modify the main surface, click  next to the **Main surface** label in the editor.
5. If you are editing a constraint or an interaction other than general contact, edit the region by selecting and unselecting objects in the viewport. If you are editing the host region to which an embedded region constraint is applied, first select the selection method. When you have finished editing the region, click mouse button 2. (For more information, see *Selecting objects within the viewport*.)

   Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

If you would rather select from a list of existing sets or surfaces, do the following:

a. Click **Sets** or **Surfaces** on the right side of the prompt area. (The name of the button depends on the type of object you are editing. For example, if you are editing an interaction, a **Surfaces** button appears.)

Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets or surfaces.

b. Select the set or surface of interest, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

6. Finish editing the interaction or constraint definition as desired, and then press mouse button 2 (if you are editing a foundation interaction) or click **OK** in the editor (if you are editing any other type of interaction or if you are editing a constraint).

Additional information

- *Using the Interaction module*
- *What are step-dependent managers?*

Using the Special menu in the Interaction module

Use of the Special menu in the Interaction module is discussed.

In this section:

- *The Special menu in the Interaction module*
- *Modeling cracks and seams*

The Special menu in the Interaction module

You can use the **Special** menu in the Interaction module to define inertia and crack engineering features.

- **Inertia**. You can define lumped mass, rotary inertia, and heat capacitance at a point on an assembly. In an Abaqus/Standard analysis you can also define mass and inertia proportional damping and composite damping. For more information, see *Inertia*.
- **Crack**. You can study the initiation and propagation of cracks using the following techniques:
  - An embedded seam crack with duplicate overlapping nodes
  - A contour integral analysis
  - The extended finite element method (XFEM)
  - The virtual crack closing technique (VCCT)
- **Springs/Dashpots**. You can define springs and dashpots that exhibit the same linear behavior independent of field variables. You can also define both spring and dashpot behavior on the same set of points. In an Abaqus/Explicit or an Abaqus/Standard analysis, you can model springs and dashpots that connect two points, following the line of action between the two points. In an Abaqus/Standard analysis, you can also model springs and dashpots that connect two points, acting in a fixed direction, or that connect points to ground. For more information, see *Springs and dashpots*.
- **Fasteners**. You can model point-to-point connections between two or more faces using point-based or discrete fasteners. Point-based fasteners can be defined using attachment points, reference points, or orphan nodes. Discrete fasteners can be defined using attachment lines. For more information, see *Fasteners*.

Modeling cracks and seams

When you model cracks, you assign seams to regions of your model. Abaqus/CAE places overlapping duplicate nodes along a seam when the mesh is generated. A seam cannot extend along the boundaries of a part and must be embedded within a face of a two-dimensional part or within a cell of a solid part.
Because a seam modifies the mesh, if you create a seam on a dependent part instance, it will actually be created on the underlying part, thereby affecting all instances of that part.

For fracture mechanics, a seam defines an edge or a face with overlapping nodes that can separate during an analysis. You can include a seam crack in your model. Alternatively, you can refer to the seam when creating a contour integral; however, you cannot use a seam crack with the extended finite element method (XFEM). For more information, see *Fracture mechanics*.

1. From the main menu bar in the Interaction module, select **Special->Crack->Assign seam**.
2. From the model in the viewport, select the entities representing the seam. The entities must be embedded edges within a face of a two-dimensional part or embedded faces within a cell of a solid part; you cannot select any entities that lie on the boundary of the part.
3. Click mouse button 2 to indicate that you have finished selecting the seam.

   Abaqus/CAE creates the seam.

Additional information

• *Using the Special menu in the Interaction module*

Using the interaction editors

This section explains how to enter data in the interaction editor to define specific types of interactions.

In this section:

- *Defining general contact*
- *Specifying and modifying contact property assignments for general contact*
- *Specifying and modifying contact initialization assignments for general contact*
- *Specifying and modifying contact stabilization assignments for general contact*
- *Specifying surface property assignments for general contact*
- *Specifying contact formulation assignments for general contact*
- *Defining surface-to-surface contact*
- *Defining self-contact*
- *Specifying contact controls in an Abaqus/Standard analysis*
- *Specifying contact controls in an Abaqus/Explicit analysis*
- *Defining a fluid cavity interaction*
- *Defining a fluid exchange interaction*
- *Defining a fluid exchange activation interaction*
- *Defining a fluid inflator interaction*
- *Defining a fluid inflator activation interaction*
- *Defining a model change interaction*
- *Defining a Standard-Explicit co-simulation interaction*
- *Defining pressure penetration*
- *Defining acoustic impedance*
- *Defining incident waves*
- *Defining cyclic symmetry*
- *Defining foundations*
- *Defining a cavity radiation interaction*
- *Defining a surface film condition interaction*
- *Defining a concentrated film condition interaction*
- *Defining a surface radiative interaction*
- *Defining a concentrated radiative interaction*
- *Defining an actuator/sensor interaction*
- *Defining contact mass scaling*

Defining general contact

In Abaqus/Standard you can define general contact only in the initial step; in all subsequent steps this general contact interaction is active and you can modify it. In Abaqus/Explicit you can define general contact in any analysis step or the initial step; only one general contact interaction can be active in a step.
For a brief overview of general contact, see *Understanding interactions*. For a more detailed discussion, see *About General Contact in Abaqus/Standard*, and *About General Contact in Abaqus/Explicit*.

A general contact definition can create interactions involving exterior faces, analytical rigid surfaces, feature edges, edges based on beams and trusses, and, for Abaqus/Explicit, Eulerian material boundaries. Analytical rigid surfaces can interact only with entities of element-based and node-based surfaces (that is, contact between two analytical surfaces cannot be modeled).

In Abaqus/Explicit you can obtain contact data for a specific surface in the general contact domain by using the history output request editor in the Step module. In the **Domain** section of the editor, select **General contact surface** and choose the surface from the menu that appears. For more information, see *Creating an output request*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a general contact interaction using the

   tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created. In Abaqus/Standard general contact can be created only in the initial step.
   - Select the **General contact (Standard)** or **General contact (Explicit)** type of interaction, depending on the analysis steps being defined in your model.

3. Click **Continue** to close the **Create Interaction** dialog box.
   The **Edit Interaction** dialog box appears.

4. Specify the contact domain using either of the following methods:
   - Choose **All* with self** to specify contact (including self-contact) for all allowable element faces and model entities. This is the simplest way to define the contact domain.
   - To specify individual contact surface pairings:

     1. Choose **Selected surface pairs**, and click .
        The **Edit Included Pairs** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(All*)**, **(Self)**, and Eulerian material surfaces. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
     2. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(All*)** to specify a surface that includes all allowable element faces and model entities.

Tip: You can click



to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.

3. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.
   - When multiple surfaces are selected in either column, all possible combinations will be generated in the table.
   - To specify self-contact, select either the same surface name or **(Self)** in the second column.
   - The order in which the surfaces are specified does not matter for the analysis.
4. Click the arrows

   in the middle of the dialog box to transfer the surface pair to the list of pairings that will be included in the contact domain.

   The table on the right side of the dialog box is updated to reflect your selections (the order of the surface pairings is irrelevant).
5. Repeat the above steps as needed to completely define the contact domain inclusions. If you want to delete included pairs, select the rows and click

   .
6. Click **OK** to save your selections and to close the **Edit Included Pairs** dialog box.

   The interaction editor reappears with updated information on the number of selected surface pairs for inclusion in the contact domain.

5. If necessary, select the surface pairs to exclude from the contact domain.
   a. Click **Edit** next to **Excluded surface pairs**.

      The **Edit Excluded Pairs** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(All*)**, **(Self)**, and Eulerian material surfaces. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
   b. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(All*)** to specify a surface that includes all allowable element faces and model entities.

      Tip: You can click

to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.

c. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.

- When multiple surfaces are selected in either column, all possible combinations will be generated in the table.
- To specify that self-contact should be excluded, select either the same surface name or **(Self)** in the second column.
- The order in which the surfaces are specified does not matter for the analysis.
- If the excluded regions overlap with the included regions, the contact exclusions will take precedence over the contact inclusions.

d. Click the arrows



in the middle of the dialog box to transfer the surface pair to the list of pairings that will be excluded from the contact domain.

The table on the right side of the dialog box is updated to reflect your selections (the order of the surface pairings is irrelevant).

e. Repeat the above steps as needed to completely define the contact domain exclusions. If you want to delete excluded pairs, select the rows and click



.

f. Click **OK** to save your selections and to close the **Edit Excluded Pairs** dialog box.

The interaction editor reappears with updated information on the number of selected surface pairs for exclusion from the contact domain.

6. Specify the **Attribute Assignments** at the bottom of the interaction editor. In Abaqus/Standard you can modify the contact properties or stabilizations in any step in which the general contact interaction is active, but all other attributes are assigned for the entire analysis. In Abaqus/Explicit you can specify or modify the attributes in any step in which the general contact interaction is active. You can specify the following assignments:

- **Contact Properties**. For detailed instructions, see *Specifying and modifying contact property assignments for general contact*.
- **Contact Initializations** (Abaqus/Standard only). For detailed instructions, see *Specifying and modifying contact initialization assignments for general contact*.
- **Contact Stabilizations** (Abaqus/Standard only). For detailed instructions, see *Specifying and modifying contact stabilization assignments for general contact*.
- **Surface Properties**. For detailed instructions, see *Specifying surface property assignments for general contact*.
- **Contact Formulation**. For detailed instructions, see *Specifying contact formulation assignments for general contact*.

7. Click **OK** to create the interaction and to close the editor.

Additional information

- *About General Contact in Abaqus/Standard*
- *About General Contact in Abaqus/Explicit*
- *Interaction editors*

Specifying and modifying contact property assignments for general contact

Contact properties can be assigned globally to a general contact interaction or individually to particular regions within a general contact domain.
Attributes for a general contact interaction, such as contact properties, are specified independently of the contact domain in the **Attribute Assignments** portion of the interaction editor. For a brief overview of general contact, see *Understanding interactions*.

You can change the contact property that is assigned to a general contact interaction in any step to which the original interaction definition is propagated. In Abaqus/Standard only the frictional behavior is allowed to change between the original contact property definition and the new contact property definition (see *Specifying frictional behavior for mechanical contact property options*)—Abaqus uses the original friction definition from the creation of the general contact interaction until the step in which the new property is assigned; the new friction definition is used for subsequent steps.

Additional information

- *Interaction editors*
- *Defining general contact*

Specify contact property assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).
3. If necessary, click  in the **Contact Properties** portion of the interaction editor to create a contact interaction property; see *Defining a contact interaction property*, for more information.
4. Specify the contact property assignments for the interaction using either of the following methods:
   - To assign a contact property globally to the entire contact domain, select a property from the list next to **Global property assignment**.
   - To assign different contact properties to individual surface pairs:
     1. Click  next to **Individual property assignments**.

        The **Edit Individual Contact Property Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**, **(Self)**, materials, and Eulerian material surfaces. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
     2. Select one or more combinations of surfaces and materials from the list of existing surfaces and materials in the first column on the left side. Select **(Global)** to assign a contact property between the entire contact domain and an individual surface or material.

         **Tip:** You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.



Tip: You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

3. Select the second surface or material or combinations of surfaces and materials from the list of existing surfaces and materials in the second column to define the surface pairings.
   - When multiple surfaces and materials are selected in either column, all possible combinations will be generated.
   - To assign a property for self-contact, select either the same surface name or **(Self)** in the second column.
   - Any contact property assignments for regions that fall outside of the contact domain will be ignored.
4. From the list of existing interaction properties in the third column, select the property to assign.

   Tip: You can click  to define a new interaction property and add it to the list. See *Defining a contact interaction property*, for instructions on defining interaction properties.
5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of property assignments.

   The table on the right side of the dialog box is updated to reflect your selections.
6. Repeat the above steps as needed to complete the contact property assignments. If you want to delete contact property assignments, select the rows and click .

   Note:

   The order of assignments may be relevant; when property assignments overlap, the last assignment will take precedence.
7. Click **OK** to save your selections and to close the **Edit Individual Contact Property Assignments** dialog box.

   The interaction editor reappears with updated information on the number of individual contact property assignments.



Note:

Individual contact property assignments take precedence over the global assignment.

5. Click **OK** to create the interaction and to close the editor.

Modify a contact property assignment

1. From the main menu bar, select **Interaction->Edit->*interaction name***.

   The **Edit Interaction** dialog box appears.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).

3. Click  next to **Individual property assignments**.
   The **Edit Individual Contact Property Assignments** dialog box appears.
4. Select the row or rows containing the property assignment to be modified from the table on the right side of the dialog box.
5. Select a new interaction property from the list in the third column on the left side of the dialog box.
6. Click the arrow  in the middle of the dialog box to replace the interaction property with your new selection.
7. Click **OK** to save your selections and to close the dialog box.
   The interaction editor reappears.
8. Click **OK** to save your changes and to close the interaction editor.

Specifying and modifying contact initialization assignments for general contact

Contact initializations determine how Abaqus adjusts surfaces in the general contact domain prior to the analysis. Contact initializations for surfaces are specified independently of the contact domain in the **Attribute Assignments** portion of the interaction editor. Contact initialization can be assigned only in the initial step of an analysis. For more information about contact initializations, see *Creating contact initializations*, *Contact Initialization for General Contact in Abaqus/Standard*, and *Contact Initialization for General Contact in Abaqus/Explicit*.

Additional information

- *Creating contact initializations*
- *Contact Initialization for General Contact in Abaqus/Standard*
- *Contact Initialization for General Contact in Abaqus/Explicit*

Specify contact initialization assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).
3. If necessary, click  in the **Contact Properties** portion of the interaction editor to create a contact initialization definition; see *Creating contact initializations*, for more information.
4. Click  next to **Initialization assignments**.

   The **Edit Initialization Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and **(Self)**. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
5. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to assign an initialization between the entire contact domain and an individual surface for an Abaqus/Standard analysis.

    **Tip:** You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
6. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.
   - When multiple surfaces are selected in either column, all possible combinations will be generated.
   - To assign an initialization for self-contact, select either the same surface name or **(Self)** in the second column for an Abaqus/Standard analysis.
   - In an Abaqus/Explicit analysis the default surfaces **(Self)** and **(Global)** are not supported.
   - Any initialization assignments for regions that fall outside of the contact domain will be ignored.

7. From the list of existing initializations in the third column, select the initialization to assign. You can create new contact initializations while you are working in the **Edit Initialization Assignments** dialog box (see *Creating contact initializations*).
8. From the list of existing secondary surface option type in the fourth column, select the secondary surface option type to assign. This list is available only if the contact initialization is defined with the clearance distance and **Adjust nodal coordinates** options.
9. Click the arrows in the middle of the dialog box to transfer your selections to the list of property assignments.

   The table on the right side of the dialog box is updated to reflect your selections.
10. Repeat the above steps as needed to complete the initialization assignments. If you want to delete initialization assignments, select the rows and click .

    Note:

    The order of assignments may be relevant; when initialization assignments overlap, the last assignment will take precedence.
11. Click **OK** to save your selections and to close the **Edit Initialization Assignments** dialog box.

    The interaction editor reappears with updated information on the number of individual initialization assignments.
12. Click **OK** to create the interaction and to close the editor.

Modify a contact initialization assignment

1. From the main menu bar, select **Interaction->Edit->*interaction name***.

   The **Edit Interaction** dialog box appears.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).
3. Click  next to **Initialization assignments**.

   The **Edit Initialization Assignments** dialog box appears.
4. Select the row or rows containing the initialization assignment to be modified from the table on the right side of the dialog box.
5. Select a new initialization from the list in the third column on the left side of the dialog box.
6. Click the arrow  in the middle of the dialog box to replace the initialization with your new selection.
7. Click **OK** to save your selections and to close the dialog box.

   The interaction editor reappears.
8. Click **OK** to save your changes and to close the interaction editor.

Specifying and modifying contact stabilization assignments for general contact

Contact stabilization introduces viscous damping between two surfaces to stabilize rigid body motion prior to the establishment of contact.
Contact stabilization definitions are assigned to surfaces independently of the contact domain in the **Attribute Assignments** portion of the interaction editor.

A contact stabilization definition can be assigned in any step of an Abaqus/Standard analysis other than the initial step. Since general contact interactions must be created in the initial step, stabilization assignments require you to edit a previously created general contact interaction.

Stabilization effects are typically enforced only in the step in which they are applied. However, if a nondefault amplitude is selected as part of the contact stabilization definition, the stabilization effects may carry over into subsequent steps as described in *Stabilization for General Contact in Abaqus/Standard*. To remove stabilization effects in this situation, you must create a separate contact stabilization definition using the **Reset values from previous steps** option and apply that stabilization definition in a subsequent step.

For more information about contact stabilization definitions, see *Creating contact stabilization definitions*.

Additional information

- *Creating contact stabilization definitions*
- *Stabilization for General Contact in Abaqus/Standard*

Specify contact stabilization assignments

1. From the main menu bar, select **Interaction->Edit->*interaction name***.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).
3. If necessary, click  in the **Contact Properties** portion of the interaction editor to create a contact stabilization definition; see *Creating contact stabilization definitions*, for more information.
4. Click  next to **Stabilization assignments**.

   The **Edit Stabilization Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for the default surfaces labeled **(Global)** and **(Self)**. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
5. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to introduce stabilization effects between the entire contact domain and an individual surface.

   Tip: You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
6. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.
   - When multiple surfaces are selected in either column, all possible combinations are generated.
   - To assign a stabilization definition for self-contact, select either the same surface name or **(Self)** in the second column.
   - Any stabilization assignments for regions that fall outside of the contact domain are ignored.

7. From the list of existing stabilization definitions in the third column, select the stabilization to assign. You can create new contact stabilization definitions while you are working in the **Edit Stabilization Assignments** dialog box (see *Creating contact stabilization definitions*).
8. Click the arrows  in the middle of the dialog box to transfer your selections to the list of stabilization assignments.

   The table on the right side of the dialog box is updated to reflect your selections.
9. Repeat the above steps as needed to complete the stabilization assignments. If you want to delete stabilization assignments, select the rows and click .

   Note:

   The order of assignments may be relevant; when stabilization assignments overlap, the last assignment takes precedence.
10. Click **OK** to save your selections and to close the **Edit Stabilization Assignments** dialog box.

    The interaction editor reappears with updated information on the number of individual stabilization assignments.
11. Click **OK** to save your changes and to close the editor.

Modify a contact stabilization assignment

1. From the main menu bar, select **Interaction**->**Edit**->***interaction name***.

   The **Edit Interaction** dialog box appears.
2. Click the **Contact Properties** tab in the **Attribute Assignments** portion of the interaction editor (if it is not already selected).
3. Click  next to **Stabilization assignments**.

   The **Edit Stabilization Assignments** dialog box appears.
4. Select the row or rows containing the stabilization assignment to be modified from the table on the right side of the dialog box.
5. Select a new stabilization definition from the list in the third column on the left side of the dialog box.
6. Click the arrow  in the middle of the dialog box to replace the stabilization definition with your new selection.
7. Click **OK** to save your selections and to close the dialog box.

   The interaction editor reappears.
8. Click **OK** to save your changes and to close the interaction editor.

Specifying surface property assignments for general contact

Attributes for a general contact interaction, such as surface properties, are specified independently of the contact domain in the **Attribute Assignments** portion of the interaction editor.

For a brief overview of general contact, see *Understanding interactions*. For a more detailed discussion, see *Surface Properties for General Contact in Abaqus/Standard*, and *Assigning Surface Properties for General Contact in Abaqus/Explicit*.

You can specify the following surface properties in a general contact interaction:

Surface thickness assignments

Assign nondefault surface thicknesses to shell or membrane surfaces.

Shell/Membrane offset assignments

Assign the surface offset to define the distance (as a fraction of the surface thickness) from the midsurface to the reference surface.

Surface smoothing assignments

Assign nondefault contact smoothing to model surfaces. Contact smoothing improves the contact stress and pressure accuracy for curved surfaces (see *Smoothing Contact Surfaces in Abaqus/Standard*).

Feature edge criteria assignments

Control the feature edges to be included in the general contact domain. Feature edges of a model include shell perimeter edges and geometric feature edges. Modeling of edge-to-edge contact is not allowed for Abaqus/Standard analyses.

Crush Trigger assignments

Control the crushing behaviors relating to CZone methodology in the general contact domain. CZone is an Abaqus/Explicit capability that integrates material, element, and contact modeling aspects to simulate crushing of laminated composites due to contact with other bodies (see *CZone Analysis*). Modeling of crush trigger contact is not allowed for Abaqus/Standard analyses.

Surface friction assignments

Establish friction coefficients as mathematical combinations of coefficients specified as surface properties. Modeling of friction contact is not allowed for Abaqus/Standard analyses.

Surface beam smoothing assignments

Assign nondefault beam smoothing to model surfaces.

Surface vertex criteria assignments

Control the vertex-to-surface contact formulation in the general contact domain. Vertex criteria of a model specify the vertex nodes that should be considered by the vertex-to-surface formulation (see *Vertex Nodes*).

Surface wear surface property assignments

Assign wear surface properties to model surfaces (see *Contact Wear*).

Additional information

- *Surface Properties for General Contact in Abaqus/Standard*
- *Assigning Surface Properties for General Contact in Abaqus/Explicit*
- *Interaction editors*

• *Defining general contact*

Specify surface thickness assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.
2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.
   The surface thickness can be modified only for surfaces defined on shells and membranes.
3. Click  next to **Surface thickness assignments**.
   The **Edit Surface Thickness Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**, and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
4. Select one or more surfaces and materials from the list of existing surfaces and materials in the column on the left. Select **(Global)** to assign the shell/membrane thickness to the entire contact domain.

    **Tips:**

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
   - You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.
5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of shell/membrane thickness assignments.
   The table on the right side of the dialog box is updated to reflect your selections.
6. Specify the thickness for each surface and material in the middle column of the **Surface Thickness Assignments** table.
   - Enter the word `ORIGINAL` to set the shell/membrane thickness equal to the original shell or membrane thickness (the default);
   - enter the word `THINNING` to set the shell/membrane thickness equal to the current shell or membrane thickness (this option is available only in Abaqus/Explicit); or
   - specify a value for the shell/membrane thickness.
7. Optionally, specify a scale factor for any of the shell/membrane thickness assignments in the last column of the **Surface Thickness Assignments** table.
8. Repeat the above steps as needed to complete the shell/membrane thickness assignments. If you want to delete thickness assignments, select the rows and click .

    **Note:** The order of assignments might be relevant; when shell/membrane thickness assignments overlap, the last assignment takes precedence.

9. Click **OK** to save your selections and to close the **Edit Surface Thickness Assignments** dialog box.

   The interaction editor reappears with updated information on the number of shell/membrane thickness assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify shell/membrane offset assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.

2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.

   The surface offset can be modified only for surfaces defined on shells and membranes.

3. Click  next to **Shell/Membrane offset assignments**.

   The **Edit Shell/Membrane Offset Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**, and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces and materials from the list of existing surfaces and materials in the column on the left. Select **(Global)** to assign the shell/membrane offset to the entire contact domain.

    **Tips:**

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
   - You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of shell/membrane offset assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

6. Specify the offset fraction for each surface and material in the second column of the **Shell/Membrane Offset Assignments** table. The offset fraction defines the distance (as a fraction of the thickness) from the midsurface to the reference surface.
   - Enter the word `ORIGINAL` to set the shell/membrane offset equal to the original shell or membrane offset (the default).
   - Enter `SPOS` to specify the top surface of the shell/membrane as the reference surface.
   - Enter `SNEG` to specify the bottom surface of the shell/membrane as the reference surface.
   - Specify a value between –0.5 (indicates the bottom surface of the shell/membrane) and 0.5 (indicates the top surface of the shell/membrane) for the offset fraction.

7. Repeat the above steps as needed to complete the shell/membrane offset assignments. If you want to delete offset assignments, select the rows and click .



Note: The order of assignments might be relevant; when shell/membrane offset assignments overlap, the last assignment takes precedence.

8. Click **OK** to save your selections and to close the **Edit Shell/Membrane Offset Assignments** dialog box.

   The interaction editor reappears with updated information on the number of shell/membrane offset assignments.

9. Click **OK** to create the interaction and to close the editor.

Specify surface smoothing assignments

1. Display the general contact interaction editor using one of the following methods:

   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.

   In an Abaqus/Standard general contact interaction, surface smoothing assignments can be specified only in the initial step. In an Abaqus/Explicit general contact interaction, surface smoothing assignments can be specified or modified in any analysis step.

2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.

3. Click  next to **Surface smoothing assignments**.

   The **Edit Surface Smoothing Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport and displays the detected axis or center of curvature that will be used in the smoothing calculation. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Toggle **Automatically assign smoothing for geometric faces** to determine whether Abaqus/CAE applies surface smoothing automatically to all appropriate surfaces in the general contact domain. This option is on by default for Abaqus/Standard general contact interactions and off by default for Abaqus/Explicit general contact interactions.

5. Select one or more surfaces from the list of existing surfaces in the column on the left.

   Tip: You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.

6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of surface smoothing assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

7. Specify the smoothing to apply to each surface in the second column of the **Surface Smoothing Assignments** table. The smoothing specified for these surfaces overrides the default global smoothing if it is applied.

   - Select **REVOLUTION** to apply circumferential smoothing to a curved surface that is symmetric about an axis of revolution (or a two-dimensional arc that is symmetric about a central point).
   - Select **SPHERICAL** to apply spherical smoothing to a curved surface that is symmetric about a central point.

- Select **TOROIDAL** to apply toroidal smoothing to a curved surface that is a circular arc symmetric about an axis of revolution.
- Select **NONE** to prevent smoothing from being applied to the specified surface.

8. Repeat the above steps as needed to complete the surface smoothing assignments. If you want to delete smoothing assignments, select the rows and click .

   Note: The order of assignments might be relevant; when smoothing assignments overlap, the last assignment takes precedence.

9. Click **OK** to save your selections and to close the **Edit Surface Smoothing Assignments** dialog box.

   The interaction editor reappears with updated information on the number of surface smoothing assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify feature edge criteria assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.

2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.

3. Click  next to **Feature edge criteria assignments**.

   The **Edit Feature Edge Criteria Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**, and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces and materials from the list of existing surfaces in the first column on the left side of the dialog box. Select **(Global)** to assign the surface feature to the entire contact domain.

   Tips:

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
   - You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

5. Select the configuration from the second column on the left (this option is available only for Abaqus/Explicit).

6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of surface feature assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

7. Specify the feature edge criteria for each surface and material in the second and third columns of the **Surface Feature Assignments** table by doing one of the following:

- Enter the word `PERIMETER` to include only perimeter edges in the general contact domain.
- Enter the word `ALL` to include all edges in the general contact domain (this option is available only for Abaqus/Explicit).
- Enter the word `PICKED` to include in the general contact domain only edges that were explicitly selected as part of the surface definition (this option is available only for shell geometries and elements in Abaqus/Explicit).
- Enter the word `NONE` to include no feature edges in the general contact domain.
- Specify an angle in degrees to include perimeter edges and geometric edges with feature angles greater than or equal to the specified angle in the general contact domain. The specified value must be between 0° and 180°. For examples of how the feature angle is calculated for different edges, see *Feature Edges*.

8. Repeat the above steps as needed to complete the surface feature assignments. If you want to delete surface feature assignments, select the rows and click .

   Note: The order of assignments might be relevant; when surface feature assignments overlap, the last assignment takes precedence.

9. For an Abaqus/Explicit general contact interaction, toggle **Use dynamic feature edges** to activate or deactivate the dynamic feature edge criterion for the contact surface.
10. Click **OK** to save your selections and to close the **Edit Feature Edge Criteria Assignments** dialog box.

    The interaction editor reappears with updated information on the number of feature edge criteria assignments.
11. Click **OK** to create the interaction and to close the editor.

Specify crush trigger assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.
2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.
3. Click  next to **Crush trigger assignments**.

   The **Edit Crush Trigger Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
4. Select one or more surfaces and materials from the list of existing surfaces in the first column on the left side of the dialog box. Select **(Global)** to assign the surface feature to the entire contact domain.

   Tips:

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.

- You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of crush trigger assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

6. Specify the trigger option for each surface and material in the second column of the **Crush Trigger Assignments** table by doing one of the following:

   - Enter the word `TRIGGER` to specify that node develops contact pressures consistent with the assigned contact pressure-overclosure relationship.
   - Enter the word `NO_TRIGGER` to specify that nodes will be reassigned trigger status during the simulation if adjacent nodes begin crushing or adjacent elements experience material failure.
   - Enter the word `NO_CRUSH` to specify that the nodes have no crush associated with adjacent elements and will never crush.

7. Specify the crush stress, crush initiation angle, and crush continuation angle for each surface and material in the third, fourth, and fifth columns of the **Crush Trigger Assignments** table by doing the following:

   - Specify a scalar value representing the stress required to initiate crushable behavior as a factor of the crush stress in the column **Crush Stress**.
   - Specify an angle in degrees representing the crush initiation angle.
   - Specify an angle in degrees representing the crush continuation angle.

8. Repeat the above steps as needed to complete the crush trigger assignments. If you want to delete crush trigger assignments, select the rows and click .

    **Note:** The order of assignments might be relevant; when crush trigger assignments overlap, the last assignment takes precedence.

9. Click **OK** to save your selections and to close the **Edit Crush Trigger Assignments** dialog box.

   The interaction editor reappears with updated information on the number of crush trigger assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify surface friction assignments

1. Display the general contact interaction editor using one of the following methods:

   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.

2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.

3. Click  next to **Surface friction assignments**.

   The **Edit Surface Friction Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting

does not apply for **(Global)** and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces and materials from the list of existing surfaces and materials in the column on the left. Select **(Global)** to assign the surface-based friction coefficient to the entire contact domain.

   Tips:

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
   - You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of surface friction assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

6. Specify the surface-based friction coefficients for each surface and material in the second column of the **Surface Friction Assignments** table.
7. Repeat the above steps as needed to complete the surface friction assignments. If you want to delete surface friction assignments, select the rows and click .

   Note: The order of assignments might be relevant; when surface friction assignments overlap, the last assignment takes precedence.

8. Click **OK** to save your selections and to close the **Edit Surface Friction Assignments** dialog box.

   The interaction editor reappears with updated information on the number of surface friction assignments.

9. Click **OK** to create the interaction and to close the editor.

Specify surface beam smoothing assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.
2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.
3. Click  next to **Surface beam smoothing assignments**.

   The **Edit Beam Smoothing Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport and displays the detected axis or center of curvature that is used in the smoothing calculation. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces from the list of existing surfaces in the column on the left.

   Tip: You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.

5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of beam smoothing assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

6. Specify the smoothing to apply to each surface in the second column of the **Beam Smoothing Assignments** table. The smoothing specified for these surfaces overrides the default global smoothing if it is applied.

   Specify a value between 0.0 and 0.5 for the beam smoothing.

7. Repeat the above steps as needed to complete the beam smoothing assignments. If you want to delete beam smoothing assignments, select the rows and click .

    **Note:** The order of assignments might be relevant; when beam smoothing assignments overlap, the last assignment takes precedence.

8. Click **OK** to save your selections and to close the **Edit Beam Smoothing Assignments** dialog box.

   The interaction editor reappears with updated information on the number of beam smoothing assignments.

9. Click **OK** to create the interaction and to close the editor.

Specify surface vertex criteria assignments

1. Display the general contact interaction editor using one of the following methods:

   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu.

2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor.

3. Click  next to **Surface vertex criteria assignments**.

   The **Edit Vertex Criteria Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and materials. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces and materials from the list of existing surfaces in the first column on the left side of the dialog box. Select **(Global)** to assign the surface feature to the entire contact domain.

    **Tips:**

   - You can click  to define a new surface and add it to the list. See *Creating surfaces*, for instructions on defining surfaces.
   - You can click  to define a new material and add it to the list. See *Creating or editing a material*, for instructions on defining materials.

5. Click the arrows  in the middle of the dialog box to transfer your selections to the list of vertex criteria assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

6. Specify the vertex criteria for each surface and material in the second column of the **Vertex Criteria Assignments** table by doing one of the following:
   - Enter the word `ALL_VERTICES` to specify that all vertex nodes should be considered by the vertex-to-surface formulation.
   - Enter the word `NO_VERTICES` to specify that no vertex nodes should be considered by the vertex-to-surface formulation.
   - Specify an angle in degrees representing the vertex angle threshold. The specified value must be between 10° and 90°.
7. Repeat the above steps as needed to complete the vertex criteria assignments. If you want to delete vertex criteria assignments, select the rows and click .

    **Note:** The order of assignments might be relevant; when vertex criteria assignments overlap, the last assignment takes precedence.
8. Click **OK** to save your selections and to close the **Edit Vertex Criteria Assignments** dialog box.
   The interaction editor reappears with updated information on the number of vertex criteria assignments.
9. Click **OK** to create the interaction and to close the editor.

Specify wear surface property assignments

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.
2. Click the **Surface Properties** tab in the **Attribute Assignments** portion of the interaction editor if it is not already selected.
3. If necessary, click  in the **Surface Properties** portion of the interaction editor to create a wear surface interaction property (see *Defining a wear interaction property*).
4. Specify the wear surface property assignments for the interaction using either of the following methods:
   - To assign a wear surface property globally to the entire contact domain, select a property from the list next to **Global Wear Surface Property Assignment**.
   - To assign different contact properties to individual surface pairs:

     a) Click  next to **Wear Surface Property Assignments**.

     The **Edit wear surface property assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**, **(Self)**, materials, and Eulerian material surfaces. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

     b) Select one or more combinations of surfaces and materials from the list of existing surfaces and materials in the first column on the left side. Select **(Global)** to assign a contact property between the entire contact domain and an individual surface or material.

Tips:

- You can click  to define a new surface and add it to the list. See *Creating surfaces* for instructions on defining surfaces.
- You can click  to define a new material and add it to the list. See *Creating or editing a material* for instructions on defining materials.

c) From the list of existing wear interaction properties in the third column, select the property to assign.

Tip: You can click  to define a new interaction property and add it to the list. See *Defining a wear interaction property* for instructions on defining interaction properties.

d) Click the arrows  in the middle of the dialog box to transfer your selections to the list of wear surface property assignments.

The table on the right side of the dialog box updates to reflect your selections.

e) Repeat the above steps as needed to complete the contact property assignments.

If you want to delete wear surface property assignments, select the rows and click .

Note: The order of assignments might be relevant; when wear surface property assignments overlap, the last assignment takes precedence.

f) Click **OK** to save your selections and to close the **Edit Wear Surface Property Assignments** dialog box.

The interaction editor reappears with updated information on the number of wear surface property assignments.

5. Click **OK** to create the interaction and to close the editor.

Specifying contact formulation assignments for general contact

Attributes for a general contact interaction, such as contact formulation, are specified independently of the contact domain in the **Attribute Assignments** portion of the interaction editor.

For a brief overview of general contact, see *Understanding interactions*. For a more detailed discussion of main-secondary formulations, see *Contact Controls Specific to General Contact in Abaqus/Standard* and *Contact Formulation for General Contact in Abaqus/Explicit*.

Additional information

- *Contact Controls Specific to General Contact in Abaqus/Standard*
- *Contact Formulation for General Contact in Abaqus/Explicit*
- *Interaction editors*
- *Defining general contact*

Specify main-secondary assignments for general contact

By default, Abaqus/Standard automatically assigns a main and a secondary role to each surface in an interacting pair; Abaqus/Explicit uses a balanced main-secondary formulation for all surfaces (except node-based surfaces and analytical rigid surfaces). You can override the default assignments and directly specify the main-secondary assignments for surface pairings in a general contact interaction.

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.
2. Click the **Contact Formulation** tab in the **Attribute Assignments** portion of the interaction editor.
3. Click  next to the following field:
   - **Main-secondary assignments** for Abaqus/Standard general contact definitions.
   - **Pure main-secondary assignments** for Abaqus/Explicit general contact definitions.

   The **Edit Main-Secondary Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and **(Self)**. You can turn off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
4. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to assign a pure main-secondary weighting for contact between the entire contact domain and an individual surface.

    **Tip:** You can click  to define a new surface and add it to the list. See *Creating surfaces* for instructions on defining surfaces.
5. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.
   - When multiple surfaces are selected in either column, all possible combinations will be generated.

- For Abaqus/Standard general contact definitions, select either the same surface name or **(Self)** in the second column to specify a balanced main-secondary formulation for self-contact interactions on a particular surface.
- If the main and secondary surfaces overlap, self-contact is excluded for the overlapping regions.

6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of main-secondary assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

7. Specify the type for the first surface in the last column of the **Main-Secondary Assignments** table.
   - Select **SECONDARY** to indicate the first surface is the secondary surface.
   - Select **MAIN** to indicate that the first surface is the main surface.
   - Select **BALANCED** to specify balanced main-secondary contact between the two surfaces (available only in Abaqus/Standard).

   For self-contact surface pairings, you must select **BALANCED**.

8. Repeat the above steps as needed to complete the main-secondary assignments. If you want to delete main-secondary assignments, select the rows and click .
9. Click **OK** to save your selections.

   The interaction editor reappears with updated information on the number of directly specified main-secondary assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify polarity assignments for general contact

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.

2. Click the **Contact Formulation** tab in the **Attribute Assignments** portion of the interaction editor.
3. Click  next to the **Polarity assignments** for Abaqus/Explicit general contact definitions.

   The **Edit Polarity Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)**. You can toggle off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to assign a polarity weighting for contact between the entire contact domain and an individual surface.

   Tip: You can click  to define a new surface and add it to the list. See *Creating surfaces* for instructions on defining surfaces.

5. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.

- When multiple surfaces are selected in either column, all possible combinations will be generated.
- If the surfaces overlap, self-contact is excluded for the overlapping regions.

6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of polarity assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

7. Specify the option in the third column of the **Polarity Assignments** table, to indicate the side of the (double-sided) elements in the second surface.

   - Select **SPOS** to indicate that the SPOS side of the (double-sided) elements in the second surface should be considered for contact with the first surface.
   - Select **SNEG** to indicate that the SNEG side of the (double-sided) elements in the second surface should be considered for contact with the first surface.
   - Select **TWO_SIDED** to indicate that both sides of the (double-sided) elements in the second surface should be considered for contact with the first surface.

8. Repeat the above steps as needed to complete the polarity assignments. If you want to delete polarity assignments, select the rows and click .

9. Click **OK** to save your selections.

   The interaction editor reappears with updated information on the number of directly specified polarity assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify sliding transition assignments for general contact

1. Display the general contact interaction editor using one of the following methods:

   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.

2. Click the **Contact Formulation** tab in the **Attribute Assignments** portion of the interaction editor.

3. Click  next to the **Sliding Transition assignments** for Abaqus/Standard general contact definitions.

   The **Edit Sliding Transition Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and **(Self)**. You can turn off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.

4. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to assign a sliding transition weighting for contact between the entire contact domain and an individual surface.

   Tip: You can click  to define a new surface and add it to the list. See *Creating surfaces* for instructions on defining surfaces.

5. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.

- When multiple surfaces are selected in either column, all possible combinations will be generated.
- For Abaqus/Standard general contact definitions, select either the same surface name or **(Self)** in the second column to specify a formulation for self-contact interactions on a particular surface.
- If the surfaces overlap, self-contact is excluded for the overlapping regions.

6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of sliding transition assignments.

   The table on the right side of the dialog box is updated to reflect your selections.

7. Specify the smoothness of the contact force redistribution on sliding in the third column of the **Sliding Transition Assignments** table.

   - Select **ELEMENT_ORDER_SMOOTHING** to indicate the smoothness of the contact force redistribution on sliding should be of the same order as the elements underlying the secondary surface.
   - Select **LINEAR_SMOOTHING** to indicate linear smoothness of the contact force redistribution on sliding.
   - Select **QUADRATIC_SMOOTHING** to indicate quadratic smoothness of the contact force redistribution on sliding.

8. Repeat the above steps as needed to complete the sliding transition assignments. If you want to delete sliding transition assignments, select the rows and click .
9. Click **OK** to save your selections.

   The interaction editor reappears with updated information on the number of directly specified sliding transition assignments.

10. Click **OK** to create the interaction and to close the editor.

Specify edge-to-edge formulation assignments for general contact

1. Display the general contact interaction editor using one of the following methods:

   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.

2. Click the **Contact Formulation** tab in the **Attribute Assignments** portion of the interaction editor.
3. Toggle on **Edge-to-Edge formulation assignment** for Abaqus/Standard general contact definitions.
4. Specify the **Formulation**.

   - Select **BOTH** to activate both formulations for edge-to-edge contact.
   - Select **NO** to deactivate edge-to-edge contact.
   - Select **RADIAL** to activate the radial edge-to-edge contact formulation.
   - Select **CROSS** to activate the formulation based on the cross product of the edge directions for edge-to-edge contact.

5. Click **OK** to create the interaction and to close the editor.

Specify sliding formulation assignments for general contact

1. Display the general contact interaction editor using one of the following methods:

- To create a new general contact interaction, follow the instructions in *Defining general contact*.
- To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.

2. Click the **Contact Formulation** tab in the **Attribute Assignments** portion of the interaction editor.
3. Click  next to the **Sliding formulation assignments** for Abaqus/Standard general contact definitions.
   The **Edit Sliding Formulation Assignments** dialog box appears. By default, when you select a surface from the list or the table, Abaqus/CAE highlights the surface in the viewport; however, highlighting does not apply for **(Global)** and **(Self)**. You can turn off **Highlight selected regions** at the bottom of the dialog box to turn off selection highlighting.
4. Select one or more surfaces from the list of existing surfaces in the first column on the left side. Select **(Global)** to assign a sliding formulation for contact between the entire contact domain and an individual surface.

    **Tip:** You can click  to define a new surface and add it to the list. See *Creating surfaces* for instructions on defining surfaces.

5. Select the second surface or surfaces from the list of existing surfaces in the second column to define the surface pairings.
   - When multiple surfaces are selected in either column, all possible combinations will be generated.
   - For Abaqus/Standard general contact definitions, select either the same surface name or **(Self)** in the second column to specify a formulation for self-contact interactions on a particular surface.
   - If the surfaces overlap, self-contact is excluded for the overlapping regions.
6. Click the arrows  in the middle of the dialog box to transfer your selections to the list of sliding formulation assignments.
   The table on the right side of the dialog box is updated to reflect your selections.
7. Select the sliding value in the third column of the **Sliding Formulation Assignments** table.
   - Select **Small Sliding** to specify the surface pairings to consider for the small-sliding approach.
   - Select **None** to exclude the surface pairings in the small-sliding approach.
8. Repeat the above steps as needed to complete the sliding formulation assignments. If you want to delete sliding formulation assignments, select the rows and click .
9. Click **OK** to save your selections.
   The interaction editor reappears with updated information on the number of directly specified sliding formulation assignments.
10. Click **OK** to create the interaction and to close the editor.

Specify beam cross-section for general contact

1. Display the general contact interaction editor using one of the following methods:
   - To create a new general contact interaction, follow the instructions in *Defining general contact*.
   - To edit an existing general contact interaction, select **Interaction->Edit->*interaction name*** from the main menu bar.

2. Click the **Contact Controls Assignment** tab in the **Attribute Assignments** portion of the interaction editor.
3. Toggle **Beam Cross Section assignment** for Abaqus/Explicit general contact definitions.
4. Specify the **Beam Cross Section assignment**:
   - Select **CIRCUMSCRIBED CIRCLE** to activate contact calculations based on a circumferential approximation of the beam element.
   - Select **EXACT** to activate contact calculations based on the actual cross section of the beam element.
5. Click **OK** to create the interaction and to close the editor.

Defining surface-to-surface contact

A surface-to-surface contact definition can be used as an alternative to general contact to model contact interactions between specific surfaces in a model.
Certain interaction behaviors can be defined only by using surface-to-surface contact. For a brief overview of surface-to-surface contact and other types of interactions available in Abaqus, see *Understanding interactions*, and *About Contact Interactions*.

You can define surface-to-surface contact in any step, including the initial step. Select **Interaction->Create** from the main menu bar, and select the main and secondary surfaces. You can define contact between edges of a wire or between faces of a solid or shell. Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. You can deactivate a surface-to-surface contact interaction in a step and, if desired, reactivate this interaction in a subsequent step. You can deactivate the interaction in a step if it will no longer be needed in the analysis.

If you are creating multiple surface-to-surface contact interactions, you may want to use the contact detection tool. This tool automates the process of selecting surfaces and allows you to create multiple interactions simultaneously. For more information, see *Using contact and constraint detection*.

You can obtain contact data for a specific surface-to-surface contact interaction by using the field and history output request editors in the Step module. In the **Domain** section of the editors, select **Interaction** and choose the name of the surface-to-surface contact interaction from the menu that appears. For more information, see *Creating an output request*.

The procedure for defining surface-to-surface contact depends on whether you are performing an analysis using Abaqus/Standard or Abaqus/Explicit. This section provides instructions for using the interaction editor to define the different surface-to-surface contact options.

In this section:

- *Defining surface-to-surface contact in an Abaqus/Standard analysis*
- *Defining surface-to-surface contact in an Abaqus/Explicit analysis*
- *Specifying interference fit options*

Defining surface-to-surface contact in an Abaqus/Standard analysis

Certain interaction behaviors can be defined in Abaqus/Standard only by using surface-to-surface contact; see *Contact Simulation Capabilities in Abaqus/Standard* for more information.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a surface-to-surface contact interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Surface-to-surface contact (Standard)** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the main surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate you have finished selecting. Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. For detailed information, see *About Contact Pairs in Abaqus/Standard*.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   The main surface that you select becomes highlighted in red in the viewport.
5. Select the secondary surface.
   a. In the prompt area, select one of the following:
      - Select **Surface** if you want to select a surface.
      - Select **Node Region** if you want to select a region from which to create a contact node set.

   b. Use one of the same methods described earlier to select the secondary surface or region.

The secondary surface or region that you select becomes highlighted in magenta in the viewport. The **Edit Interaction** dialog box appears.

6. The **Switch Surfaces** option allows you to interchange your main and secondary surface selections without having to start over. The **Switch Surfaces**  icon is available only if you selected **Surface** in the previous step.
7. Choose the sliding formulation.
   - Choose **Finite sliding** to use the finite-sliding formulation, which is the most general and allows any arbitrary motion of the surfaces.
   - Choose **Small sliding** to use the small-sliding formulation, which assumes that although two bodies may undergo large motions, there will be relatively little sliding of one surface along the other.

   For more information, see *Contact Formulations in Abaqus/Standard*.
8. Select the discretization method.
   - Select **Node to surface** to use the node-to-surface discretization method.
   - Select **Surface to surface** to use the surface-to-surface discretization method.

   For more information, see *Discretization of Contact Pair Surfaces*.
9. Different fields become available depending upon the combination of your sliding formulation and discretization method selections.
   - By default, shell and membrane thicknesses are included in contact calculations for the following combinations: **Small sliding** and **Node to surface**, **Small sliding** and **Surface to surface**, and **Finite sliding** and **Surface to surface**. You can toggle on **Exclude shell/membrane element thickness** to ignore shell and membrane thickness for any of these combinations.

     Contact interactions using **Finite sliding** and **Node to surface** do not account for surface thickness. For more information, see *Accounting for Shell and Membrane Thickness*.
   - For contact interactions using the **Node to surface** discretization method, you can specify a smoothing factor in the **Degree of smoothing for main surface** field. For more information, see *Smoothing Main Surfaces for the Finite-Sliding, Node-to-Surface Formulation*.
   - By default, a selective scheme of supplementary contact constraints is used for the following combinations: **Finite sliding** and **Node to surface**, **Small sliding** and **Node to surface**, and **Small sliding** and **Surface to surface**. For these combinations, you can specify when to **Use supplementary contact points** as follows:
     - Choose **Selectively** to use a selective scheme of supplementary contact constraints.
     - Choose **Never** to forgo the use of supplementary contact constraints.
     - Choose **Always** to add supplementary contact constraints when applicable.

     For more information, see *Supplementary Contact Constraints*.
   - For contact interactions using **Finite sliding** and **Surface to surface**, you can choose the **Contact tracking** method.
     - Choose **Single configuration (state)** to use the state-based tracking algorithm.
     - Choose **Two configurations (path)** to use the path-based tracking algorithm.

     For more information, see *Path-Based Versus State-Based Tracking Algorithms*.



Note:

If your contact interaction uses the surface-to-surface discretization method and one or more of the surfaces in the contact interaction is an analytical rigid surface, you should choose the state-based tracking algorithm.

10. Specify the secondary node adjustment option. For more information, see *Contact Initialization for Contact Pairs in Abaqus/Standard* and *Defining Tied Contact in Abaqus/Standard*.
11. For contact interactions using the **Surface to surface** discretization method, you can apply a smoothing to contacting surfaces that reduces inaccuracies in contact pressures caused by mesh discretization on curved geometries. Click the **Surface Smoothing** tab, and select one of the following options:

    - Choose **Do not smooth** to prevent smoothing from being applied.
    - Choose **Automatically smooth 3D geometry surfaces when applicable** to apply smoothing to axisymmetric or spherical surfaces (or portions of surfaces) that are identified automatically by Abaqus/CAE. Automatic smoothing has no effect on mesh parts or two-dimensional models.

    For more information about contact smoothing techniques, see *Smoothing Contact Surfaces in Abaqus/Standard*.
12. For contact interactions using the **Small sliding** formulation, you can specify an initial clearance between the nodes on the secondary surface and the main surface. Click the **Clearance** tab, select a clearance type from the **Initial clearance** field, and enter all of the data necessary to define the clearance and contact direction. For more information, see *Additional Contact Initialization Options for Small-Sliding Contact in Abaqus/Standard*.
13. If you specify node-to-surface discretization for your contact interaction, you can also limit bonding to secondary nodes in a particular subset. Click the **Bonding** tab, toggle on **Limit bonding to secondary nodes in subset**, and select a node set from the list.

    You can limit bonding for either of the following:

    - When you want to specify a subset of initially secondary nodes that should experience cohesive forces. Strain-free adjustments will be made for those nodes initially not in contact but specified in the node set. All secondary nodes outside of this set (including those that are initially contacting the main surface) will experience only compressive contact forces over the course of the analysis. For more information, see *Specifying cohesive behavior properties for mechanical contact property options*.
    - When you want to identify the initially bonded region of the secondary surface in a VCCT crack. The unbonded portion of the secondary surface behaves as a regular contact surface. The predetermined crack surfaces are assumed to be initially partially bonded so that the crack tips can be identified explicitly during the analysis. For more information, see *Defining Initially Bonded Crack Surfaces in Abaqus/Standard*.
14. Select a contact interaction property. If desired, click  to create the interaction property.

    For more information, see *Defining a contact interaction property* and *Contact Constraint Enforcement Methods in Abaqus/Standard*.
15. To specify interference fit options, click **Interference Fit**. Interference fit options cannot be specified in the initial step. See *Specifying interference fit options* below for more detailed instructions on entering interference fit options.

16. If desired, click the arrow next to the **Contact controls** field and select the customized contact controls to use for this interaction. Only previously created Abaqus/Standard contact controls appear in the list. For more information, see *Specifying contact controls in an Abaqus/Standard analysis*.
17. To deactivate and reactivate a contact interaction in a step, toggle **Active in this step**. The contact interaction is active in the step in which it was created. For more information, see *Removing and Reactivating Contact Pairs*.
18. Click **OK** to create the interaction and to close the editor.

Additional information

- *About Contact Pairs in Abaqus/Standard*
- *Interaction editors*
- *Customizing contact controls*

Defining surface-to-surface contact in an Abaqus/Explicit analysis

Certain interaction behaviors can be defined in Abaqus/Explicit only by using surface-to-surface contact; see *Contact Simulation Capabilities in Abaqus/Explicit* for more information.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a surface-to-surface contact interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Surface-to-surface contact (Explicit)** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the main surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate you have finished selecting. Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. For detailed information, see *About Contact Pairs in Abaqus/Explicit*.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   The main surface that you select becomes highlighted in red in the viewport.
5. Select the secondary surface.
   a. In the prompt area, select one of the following:
      - Select **Surface** if you want to select a surface.
      - Select **Node Region** if you want to select a region from which to create a contact node set.

   b. Use one of the same methods described earlier to select the secondary surface or region.

The secondary surface or region that you select becomes highlighted in magenta in the viewport.

The **Edit Interaction** dialog box appears.

6. The **Switch Surfaces** option allows you to interchange your main and secondary surface selections without having to start over. The **Switch Surfaces**  icon is available only if you selected **Surface** in the previous step.
7. Choose the mechanical constraint formulation.
   - Choose **Kinematic contact method** to use a kinematic predictor/corrector contact algorithm.
   - Choose **Penalty contact method** to use the penalty contact algorithm.

   For more information, see *Contact Constraint Enforcement Methods in Abaqus/Explicit*.
8. Choose the sliding formulation.
   - Choose **Finite sliding** to use the finite-sliding formulation, which is the most general and allows any arbitrary motion of the surfaces.
   - Choose **Small sliding** to use the small-sliding formulation, which assumes that although two bodies may undergo large motions, there will be relatively little sliding of one surface along the other.

   The small-sliding formulation can be specified for interactions created in only the initial step or the first general analysis step. Interactions created in subsequent steps always use the finite-sliding formulation by default. For more information, see *Contact Formulations for Contact Pairs in Abaqus/Explicit*.
9. For contact interactions using the **Small sliding** formulation, you can specify an initial clearance between the nodes on the secondary surface and the main surface. Clearance options are available only in the first general analysis step. Select a clearance type from the **Initial clearance** field, and enter all of the data necessary to define the clearance and contact direction. For more information, see *Specifying Initial Clearance Values Precisely*.
10. Select a contact interaction property. If desired, click  to create the interaction property; see *Defining a contact interaction property*, for more information.
11. Choose the weighting factor. For more information, see *Contact Formulations for Contact Pairs in Abaqus/Explicit*.
12. If desired, click the arrow next to the **Contact controls** field and select the customized contact controls to use for this interaction. Only previously created Abaqus/Explicit contact controls appear in the list. For more information, see *Specifying contact controls in an Abaqus/Explicit analysis*.
13. To deactivate and reactivate a contact interaction in a step, toggle **Active in this step**. The contact interaction is active in the step in which it was created.
14. Click **OK** to create the interaction and to close the editor.

Additional information

- *About Contact Pairs in Abaqus/Explicit*
- *Interaction editors*
- *Customizing contact controls*

Specifying interference fit options

When you are defining surface-to-surface contact for Abaqus/Standard, you can specify interference fit options that help Abaqus/Standard resolve excessive overclosure between surfaces in the initial configuration of a model. For more information, see *Modeling Contact Interference Fits in Abaqus/Standard*.

To open the **Interference Fit Options** dialog box, click **Interference Fit** in the Abaqus/Standard interaction editor (see *Defining surface-to-surface contact in an Abaqus/Standard analysis*" above for details).

1. In the **Interference Fit Options** dialog box, select **Gradually remove secondary node overclosure during the step** to prescribe allowable intereferences.
2. Select one of the following options:
   - Select **Automatic shrink fit (first general analysis step only)** if you want Abaqus/Standard to assign a different allowable interference to each secondary node that is equal to that node's initial penetration. If you select this option, skip to Step 6.
   - Select **Uniform allowable interference** to specify a single allowable interference that will be applied to every secondary node.
3. Click the arrow to the right of the **Amplitude** field to select the name of an amplitude curve that defines the magnitude of the prescribed interference during the step. Alternatively, you can select **(Ramp)** to apply the prescribed interference immediately at the beginning of the step and ramp it down to zero linearly over the step.

   If necessary, you can click  to define a new amplitude curve. For more information, see *Selecting an amplitude type to define*.
4. In the **Magnitude at start of step** field, enter the magnitude of the allowable interference at the start of the step.
5. If desired, select the **Interference Direction** option **Along direction** to specify a shift direction vector. The relative shift is applied to the secondary nodes before Abaqus/Standard determines the contact conditions. If you select this option, enter the following:
   - In the **X** field, enter the *X*-direction cosine of the shift direction vector.
   - In the **Y** field, enter the *Y*-direction cosine of the shift direction vector.
   - In the **Z** field, enter the *Z*-direction cosine of the shift direction vector.
6. Click **OK** to save the interference fit options that you have specified and to return to the **Edit Interaction** dialog box.

Defining self-contact

A self-contact definition can be used as an alternative to general contact to model contact interactions between different areas of a single surface.
Certain interaction behaviors can be defined only by using self-contact. For a brief overview of self-contact and other types of interactions available in Abaqus, see *Understanding interactions*, and *About Contact Interactions*.

You can define self-contact in any step, including the initial step. Select **Interaction->Create** from the main menu bar and select the surface. You can define self-contact between an edge of a wire, a face of a solid, or a face of a shell. Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. You can deactivate a self-contact interaction in a step and, if desired, reactivate this interaction in a subsequent step. You can also deactivate the interaction in a step if it will no longer be needed in the analysis.

You can obtain contact data for a specific self-contact interaction by using the field and history output request editors in the Step module. In the **Domain** section of the editors, select **Interaction** and choose the name of the self-contact interaction from the menu that appears. For more information, see *Creating an output request*.

The procedure for defining self-contact depends on whether you are performing an analysis using Abaqus/Standard or Abaqus/Explicit. This section provides instructions for using the interaction editor to define the different surface-to-surface contact options.

In this section:

- *Defining self-contact in an Abaqus/Standard analysis*
- *Defining self-contact in an Abaqus/Explicit analysis*

Defining self-contact in an Abaqus/Standard analysis

Certain interaction behaviors can be defined in Abaqus/Standard only by using self-contact; see *Contact Simulation Capabilities in Abaqus/Standard* for more information.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a self-contact interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Self-contact (Standard)** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. For detailed information, see *About Contact Pairs in Abaqus/Standard*.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   The **Edit Interaction** dialog box appears.
5. Select the discretization method.
   - Select **Node to surface** to use the node-to-surface discretization method.
   - Select **Surface to surface** to use the surface-to-surface discretization method.

   For more information, see *Discretization of Contact Pair Surfaces*.
6. Different fields become available depending upon your discretization method selection.

- For contact interactions using the **Node to surface** discretization method, you can specify the following:

  1. Enter a smoothing factor in the **Degree of smoothing** field. For more information, see *Smoothing Main Surfaces for the Finite-Sliding, Node-to-Surface Formulation*.
  2. By default, a selective scheme of supplementary contact constraints is used. You can specify when to **Use supplementary contact points** as follows:

     - Choose **Selectively** to use a selective scheme of supplementary contact constraints.
     - Choose **Never** to forgo the use of supplementary contact constraints.
     - Choose **Always** to add supplementary contact constraints when applicable.

     For more information, see *Supplementary Contact Constraints*.

- For contact interactions using the **Surface to surface** discretization method, you can specify the following:

  1. By default, shell and membrane thicknesses are included in the contact calculations. You can toggle on **Exclude shell/membrane element thickness** to ignore shell and membrane thickness. Contact interactions using the **Node to surface** discretization method do not account for surface thickness.
  2. Choose the **Constraint position**.

     - Choose **Node centered** to center contact constraints at secondary nodes.
     - Choose **Face centered** to center contact constraints within secondary faces.

     For more information, see *Defining Self-Contact*.
  3. Choose the **Contact tracking** algorithm.

     - Choose **Single configuration (state)** to use the state-based tracking algorithm.
     - Choose **Two configurations (path)** to use the path-based tracking algorithm.

     For more information, see *Path-Based Versus State-Based Tracking Algorithms*.

     Note:

     If you use the surface-to-surface discretization method and the surface for which you are defining self-contact is an analytical rigid surface, you should choose the state-based tracking algorithm.

7. Select a contact interaction property. If desired, click  to create the interaction property; see *Defining a contact interaction property* for more information.

   If you choose the **Surface to surface** discretization method, the contact interaction property that you select cannot specify a “hard” contact pressure-overclosure relationship. For more information, see *Contact Constraint Enforcement Methods in Abaqus/Standard* and *Defining mechanical contact property options*.
8. If desired, click the arrow next to the **Contact controls** field and select the customized contact controls to use for this interaction. Only previously created Abaqus/Standard contact controls appear in the list. For more information, see *Specifying contact controls in an Abaqus/Standard analysis*.

9. To deactivate and reactivate a contact interaction in a step, toggle **Active in this step**. The contact pair is active in the step in which it was created. For more information, see *Removing and Reactivating Contact Pairs*.
10. Click **OK** to create the interaction and to close the editor.

Additional information

- *About Contact Pairs in Abaqus/Standard*
- *Interaction editors*
- *Customizing contact controls*

Defining self-contact in an Abaqus/Explicit analysis

Certain interaction behaviors can be defined in Abaqus/Explicit only by using self-contact; see *Contact Simulation Capabilities in Abaqus/Explicit* for more information.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a self-contact interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Self-contact (Explicit)** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Certain connectivity restrictions apply to contact surfaces depending on the type of contact formulation. For detailed information, see *About Contact Pairs in Abaqus/Explicit*.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   The **Edit Interaction** dialog box appears.
5. Choose the mechanical constraint formulation.
   - Choose **Kinematic contact method** to use a kinematic predictor/corrector contact algorithm.
   - Choose **Penalty contact method** to use the penalty contact algorithm.

   For more information, see *Contact Constraint Enforcement Methods in Abaqus/Explicit*.

6. Select a contact interaction property. If desired, click  to create the interaction property; see *Defining a contact interaction property*, for more information.
7. If desired, click the arrow next to the **Contact controls** field and select the customized contact controls to use for this interaction. Only previously created Abaqus/Explicit contact controls appear in the list. For more information, see *Specifying contact controls in an Abaqus/Explicit analysis*.
8. To deactivate and reactivate the contact interaction, toggle **Active in this step**. The contact pair is active in the step in which it was created.
9. Click **OK** to create the interaction and to close the editor.

Additional information

- *About Contact Pairs in Abaqus/Explicit*
- *Interaction editors*
- *Customizing contact controls*

Specifying contact controls in an Abaqus/Standard analysis

You can specify contact controls for surface-to-surface contact and self-contact interactions in an Abaqus/Standard analysis. For a detailed discussion, see *Generally Applicable Contact Controls in Abaqus/Standard*.



Warning:

Contact controls are intended for advanced users. The default settings of these controls are appropriate for most analyses. Using nondefault values of these controls may greatly increase the computational time of the analysis, produce inaccurate results, or cause convergence problems.

1. From the main menu bar, select **Interaction->Contact Controls->Create**.
2. In the **Create Contact Controls** dialog box that appears, do the following:
   - Name the contact controls.
   - Select **Abaqus/Standard contact controls** as the type of contact controls.
3. Click **Continue** to close the **Create Contact Controls** dialog box.
   The **Edit Contact Controls** dialog box appears.
4. In the **Stabilization** portion of the editor, you can specify controls relating to automatic stabilization of rigid body motions in contact problems using viscous damping.
   - Select one of the following:
     - Select **Automatic stabilization** to use the default damping coefficient calculated automatically by Abaqus/Standard. If desired, you can enter a value for the **Factor** by which the default damping coefficient will be multiplied.
     - Select **Stabilization coefficient** to specify the damping coefficient directly, and enter a value.
   - In the **Tangent fraction** field, enter a value for the fraction of the normal stabilization by which to modify the tangential stabilization. By default, the tangential and normal stabilization are the same.
   - Enter a value for the **Fraction of damping at end of step**.
   - Specify the **Clearance at which damping becomes zero**.
     - Select **Computed** to use the default clearance value calculated by Abaqus/Standard.
     - Select **Specify** to enter a value for the clearance at which the damping becomes zero.
5. In the **Augmented Lagrange** portion of the editor, you can specify controls for interactions defined with a contact interaction property that uses augmented Lagrangian surface behavior.
   - In the **Stiffness scale factor** field, enter a value for the factor by which Abaqus/Standard will scale the default penalty stiffnesses to obtain the stiffnesses used for the contact pairs. The default value is 1.
   - In the **Penetration tolerance** field, select one of the following to specify the allowable penetration that is permitted to violate the impenetrability condition:
     - Select **Absolute**, and enter a value for the allowable penetration.
     - Select **Relative**, and enter the ratio of the allowable penetration to the characteristic contact surface face dimension. The default value is 0.001.

6. Click **OK** to create the contact controls and to close the editor.

Additional information

• *About Contact Pairs in Abaqus/Standard*
• *Contact controls editors*
• *Customizing contact controls*

Specifying contact controls in an Abaqus/Explicit analysis

You can specify contact controls for surface-to-surface contact and self-contact interactions in an Abaqus/Explicit analysis. For more information, see *Contact Controls for Contact Pairs in Abaqus/Explicit*.



Warning:

Contact controls are intended for advanced users. The default settings of these controls are appropriate for most analyses. Using nondefault values of these controls may greatly increase the computational time of the analysis or produce inaccurate results.

1. From the main menu bar, select **Interaction->Contact Controls->Create**.
2. In the **Create Contact Controls** dialog box that appears, do the following:
   - Name the contact controls.
   - Select **Abaqus/Explicit contact controls** as the type of contact controls.
3. Click **Continue** to close the **Create Contact Controls** dialog box.

   The **Edit Contact Controls** dialog box appears.
4. By default, the maximum number of increments between global contact searches is 4 for self-contact interactions and 100 for surface-to-surface contact interactions. If you want to specify the maximum number of increments between global searches, toggle on **Specify max number of increments** in the **Global Search Frequency** portion of the editor and enter a value.
5. By default, Abaqus/Explicit uses techniques for local contact searches (local tracking) that use a minimum amount of computational time. If you have difficulty enforcing the appropriate contact conditions, you can toggle off **Fast local tracking** to use a more conservative local contact search. This setting applies only for surface-to-surface contact interactions.
6. In the **Penalty stiffness scaling factor** field, enter a value for the factor by which Abaqus/Explicit will scale the default penalty stiffness to obtain the stiffnesses used for the penalty contact pairs. The default value is 1.
7. In the **Warp check increment** field, enter the number of increments between checks for highly warped facets on main surfaces. The default value is 20. More frequent checks will cause a slight increase in computational time.
8. In the **Angle criteria for highly warped facet (degrees)** field, enter a value for the out-of-plane warping angle (in degrees) at which Abaqus/Explicit will consider a facet to be highly warped. The out-of-plane warping angle is defined as the amount of variation of the surface normal over a facet. The default value is 20°.
9. Click **OK** to create the contact controls and to close the editor.

Additional information

- *About Contact Pairs in Abaqus/Explicit*
- *Contact Formulations for Contact Pairs in Abaqus/Explicit*
- *Common Difficulties Associated with Contact Modeling Using Contact Pairs in Abaqus/Explicit*
- *Contact controls editors*
- *Customizing contact controls*

Defining a fluid cavity interaction

A fluid cavity interaction allows you to define a liquid- or gas-filled volume within a model. You can define a fluid cavity interaction in the initial step of an Abaqus/Standard or Abaqus/Explicit analysis. The fluid cavity interaction cannot be modified or deactivated in subsequent analysis steps. For a detailed discussion, see *Fluid Cavity Definition*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a fluid cavity interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select **Initial** for the step in which the interaction will be created.
   - Select the **Fluid cavity** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Select the cavity point.

   The cavity point is a reference node used to identify the cavity. It should not be connected to any elements in the model. For symmetrical models the cavity point must lie on the axis or axes of symmetry. You can choose a cavity point from the viewport or from a saved set. The default selection method is based on the selection method you most recently employed. To change methods, click **Select in Viewport** or **Sets** on the right side of the prompt area.
5. If your model contains both mesh and geometry regions, select **Geometry** or **Mesh** in the prompt area to specify the region that contains the cavity.
6. Select the cavity surface.

   The cavity surface is made up of all the model faces that enclose the cavity. You can choose the faces of a cavity surface from the viewport or select a saved surface. The default selection method is based on the selection method you most recently employed. To change methods, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   The **Edit Interaction** dialog box appears.
7. Select a fluid cavity property. If desired, click to create the interaction property; see *Defining a fluid cavity interaction property*, for more information.
8. If desired, toggle on **Specify ambient pressure**, and enter a pressure to account for the effects of external ambient pressure on the fluid cavity.
9. For two-dimensional models, you must specify an out-of-plane thickness.

   The out-of-plane thickness is used to define the cavity volume.
10. For Abaqus/Explicit analyses using a pneumatic fluid cavity interaction property, toggle on **Use adiabatic behavior** if desired.
11. If desired, toggle off **Check surface normals** to prevent Abaqus/CAE from checking that all surface normals bounding the fluid cavity point in toward the cavity.

    This check can be computationally expensive for complex cavity geometry.
12. Click **OK** to create the interaction and to close the editor.

Symbols that represent the fluid cavity interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

- *Fluid Cavity Definition*
- *Interaction editors*

Defining a fluid exchange interaction

A fluid exchange interaction allows you to define the movement of fluid between a cavity and the surrounding environment or between two cavities. The fluid exchange interaction cannot be modified or deactivated in subsequent analysis steps. For a detailed discussion, see *Fluid Exchange Definition*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a fluid cavity interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select **Initial** for the step in which the interaction will be created.
   - Select the **Fluid exchange** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.

   Abaqus/CAE opens the **Edit Interaction** dialog box.
4. Toggle on the desired exchange definition, **To environment** or **Between cavities**.
5. Select the primary fluid cavity interaction, **Fluid cavity interaction 1**.
6. If you are defining an exchange between cavities, select **Fluid cavity interaction 2**.
7. Select a **Fluid exchange property**. If desired, click  to create the interaction property; see *Defining a fluid exchange interaction property*, for more information.
8. Enter a value for the **Effective exchange area**.

   The effective exchange area represents the cross-sectional area of a pipe that the fluid is exchanged through.
9. Click **OK** to create the interaction and to close the editor.

   Symbols that represent the fluid exchange interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

- *Fluid Cavity Definition*
- *Interaction editors*

Defining a fluid exchange activation interaction

A fluid exchange activation interaction allows you to activate a fluid exchange between two fluid cavities or between a fluid cavity and the environment.

Before you begin: You must create a fluid cavity and fluid exchange interactions.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a fluid exchange activation interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   a. Name the interaction.

      For more information about naming objects, see *Using basic dialog box components*.

   b. Select the step.

      Fluid exchange activation is active only in dynamic steps.

   c. Select the **Fluid exchange activation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.

   Abaqus/CAE opens the **Edit Interaction** dialog box.
4. Select an **Amplitude** to define a multiplier for the fluid exchange magnitude.

   Tip: Alternatively, you can click  to create a new amplitude. For more information, see *The Amplitude toolset*.

5. Enter the **Delta leakage area** to define the rate of the fluid leakage.
6. Enter a list of **Fluid exchanges** that need to be activated in this step.
7. Select **Consider blockages by contacted surface** to consider vent or leakage area obstruction by contacted surfaces.
8. Select **Allow outflow only** if the flow is allowed only from the first fluid cavity to the second fluid cavity in the fluid exchange.
9. Click **OK** to create the interaction and to close the editor.

   Symbols that represent the fluid exchange activation interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

- *Fluid Exchange Definition*
- *Interaction editors*

Defining a fluid inflator interaction

A fluid inflator interaction allows you to inflate a fluid cavity to model the flow characteristics of inflators used for airbag systems.

Before you begin: You must create a fluid cavity interaction.

Create the fluid cavity carefully, making sure that both the reference point and the cavity surfaces are contained within a single part. This helps avoid multiple definitions of the fluid inflator in the input file.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a fluid inflator interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   a. Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   b. Select **Initial** for the step in which the interaction will be created.
   c. Select the **Fluid inflator** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.

   Abaqus/CAE opens the **Edit Interaction** dialog box.
4. Select a **Fluid cavity interaction** that defines the fluid cavity reference node to identify the cavity.
5. Select a **Fluid inflator property**. Click to create the interaction property (see *Defining a fluid inflator interaction property* for more information).
6. Click **OK** to create the interaction and to close the editor.

   Symbols that represent the fluid inflator interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

- *Inflator Definition*
- *Interaction editors*

Defining a fluid inflator activation interaction

A fluid inflator activation interaction allows you to activate the inflation of fluid into a fluid cavity to model the flow characteristics of inflators used for airbag systems.

Before you begin: You must create a fluid cavity and fluid inflator interactions.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a fluid inflator activation interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   a. Name the interaction.

      For more information about naming objects, see *Using basic dialog box components*.

   b. Select the step.

      Fluid inflator activation is active only in dynamic steps.

   c. Select the **Fluid inflator activation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.

   Abaqus/CAE opens the **Edit Interaction** dialog box.
4. Select an **Inflation time amplitude** to define a mapping between the inflation time and the actual time.

   Tip: Alternatively, you can click  to create a new amplitude. For more information, see *The Amplitude toolset*.

5. Select a **Mass flow amplitude** to modify the mass flow rate.

   Tip: Alternatively, you can click  to create a new amplitude. For more information, see *The Amplitude toolset*.

6. Specify a list of **Fluid inflators** that need to be activated in this step.
7. Click **OK** to create the interaction and to close the editor.

   Symbols that represent the fluid inflator activation interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

- *Inflator Definition*
- *Interaction editors*

Defining a model change interaction

A model change interaction allows you to deactivate and reactivate elements to simulate removal of part of the model, either temporarily or for the remainder of the analysis. You can create a model change interaction in all Abaqus/Standard steps except for static, Riks steps and linear perturbation steps. For a detailed discussion, see *Element and Contact Pair Removal and Reactivation*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a model change interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Model change** type of interaction.

3. Click **Continue** to close the **Create Interaction** dialog box.

   The **Edit Interaction** dialog box appears.

4. Specify the model change definition.
   - Choose **Region** to define the region for the model change interaction for the current simulation.
   - Choose **Restart** to allow element or contact model changes in a subsequent restart analysis. Use this model change definition when there are no other model change interactions present. Do not use this definition if you deactivated contact pairs in the first analysis step, or if you created contact pairs after the first analysis step.

5. If you selected the **Region** model change definition, perform the following steps:

   a. Select the region type and region.

   - Select **Geometry** to use model geometry for the model change region, click  to select the region, and make your selection in the viewport.
   - Select **Skins** to use skins for the model change region, and click  to select the region. From the prompt area, select **(pick entire skin)** or **(pick partial skin)**, then make your selection in the viewport. If the selected part has multiple skins, Abaqus/CAE displays the ambiguous picking options in the prompt area while you make your selection.
   - Select **Stringers** to use stringers for the model change region, and click  to select the region. From the prompt area, select **(pick entire stringer)** or **(pick partial stringer)**, then make your selection in the viewport. If the selected part has multiple stringers, Abaqus/CAE displays the ambiguous picking options in the prompt area while you make your selection.
   - Select **Elements** to use elements for the model change region, and click  to select the region. From the prompt area, select **individually** or **by angle**, then make your selection in the viewport. The mesh must be visible to select elements. For more information on this selection method, see *Using the angle and feature edge method to select multiple objects*.

For all region types, you can use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select a valid set from the **Region Selection** dialog box that appears, and click **Continue**.

The default selection method is based on the selection method you most recently employed. To revert to the old method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

b. Select the activation state of the region elements.

- Select **Deactivated in this step** to deactivate the selected region in the current step.
- Select **Reactivated in this step** to activate the selected region in the current step. In subsequent steps you can select this option to reactivate the selected region if it was previously deactivated.

c. If you selected **Reactivated in this step**, you can toggle on **Reactivate elements with strain (if applicable)** to include strain for reactivated stress/displacement elements. Toggle this option off to reset the elements to the initial strain configuration.

6. Click **OK** to create the interaction and to close the editor.

Symbols that represent the model change interaction that you just created appear in the viewport. For more information, see *Understanding symbols that represent interactions, constraints, and connectors*.

Additional information

• *Element and Contact Pair Removal and Reactivation*

• *Interaction editors*

Defining a Standard-Explicit co-simulation interaction

You use a Standard-Explicit co-simulation interaction to define the interface region and coupling schemes for an Abaqus/Standard to Abaqus/Explicit co-simulation. Select **Interaction->Create** from the main menu bar, and select the region for exchanging data. For more information, see *Co-simulation*."

A Standard-Explicit co-simulation interaction can be created only in a general static, implicit dynamic, or explicit dynamic step. The interaction is valid only in the step in which it is created and is not propagated to subsequent steps. Only one Standard-Explicit co-simulation interaction can be active in a model.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a Standard-Explicit co-simulation interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Standard-Explicit co-simulation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. In the prompt area, select one of the following for the co-simulation region type:
   - Select **Surface** if you want to select a surface.
   - Select **Node Region** if you want to select nodes.
5. Use one of the following methods to select the region:
   - Use an existing set or surface to define the region. On the right side of the prompt area, click **Sets** or **Surfaces**. Select an existing set or surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

6. Choose the **Incrementation control** method. The same incrementation method must be used in the Abaqus/Standard and Abaqus/Explicit analyses. This option is not available in a static general step.
   - Choose **Allow subcycling** to allow the Abaqus/Standard increment size to differ from those in Abaqus/Explicit; fields will be exchanged as needed.
   - Choose **Lock time steps** to force Abaqus/Standard to match the increment size of Abaqus/Explicit; fields will be exchanged at each of the shared increments.

7. Choose the **Coupling step period**. In Abaqus/Standard analyses, Abaqus always uses the next increment size as the suggested coupling step size.
   - Choose **Determined by analysis** to have Abaqus use the next increment size as the suggested coupling step size.
   - Choose **Specified**, and enter a value for the coupling step size (available only in an Abaqus/Explicit analysis).

8. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Co-simulation*
- *Preparing an Abaqus Analysis for Co-Simulation*
- *Structural-to-Structural Co-Simulation*

Defining pressure penetration

A pressure penetration interaction allows you to simulate the pressure of a fluid penetrating between two surfaces involved in surface-to-surface contact. The fluid pressure is applied normal to the surfaces. The bodies forming the joint can both be deformable, as is the case with threaded connectors; or one can be rigid, as occurs when a soft gasket is used as a seal between stiffer structures.

Pressure penetration interactions can be applied in three-dimensional, planar (two-dimensional), or axisymmetric models. A pressure penetration interaction can be used only in an Abaqus/Standard analysis.

Before defining the pressure penetration interaction, you must create a surface-to-surface contact interaction to specify the main and secondary surfaces for the pressure penetration; see *Defining surface-to-surface contact in an Abaqus/Standard analysis*. When you create the surface-to-surface contact interaction, any combination of **Sliding formulation** and **Discretization method** can be used (for compatibility with pressure penetration).

In the pressure penetration definition you identify the contact surfaces, the regions on the surfaces exposed to the fluid pressure, the magnitude of the fluid pressure, and the critical contact pressure acting on the regions. In a three-dimensional model points, edges, and faces can be selected as the regions exposed to the fluid pressure. In a two-dimensional model only points can be selected. In a two-dimensional model you must identify the penetration points on both the main and secondary surfaces (unless the main surface is an analytical rigid surface).

The fluid can penetrate from either one or multiple regions on the surface. These regions are always subjected to the pressure penetration, regardless of their contact status. Fluid will penetrate into the region between the contacting bodies until a point is reached where the contact pressure is greater than the specified critical value, cutting off further penetration of the fluid.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a pressure penetration interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created.
   - Select the **Pressure penetration** type of interaction.

3. Click **Continue** to close the **Create Interaction** dialog box.
4. From the **Contact interaction** list, select the surface-to-surface contact interaction to which the pressure penetration will be applied.

   The main and secondary surfaces of the contact interaction are shown in the dialog box, and the surfaces are highlighted in the viewport.

5. For a three-dimensional model, do the following in the **Penetration Regions** table:
   a. Identify the first region on the secondary surface that is exposed to the fluid pressure.

      Double-click the empty cell in the **Region on Secondary** column, or select the cell and click the button; then use one of the following methods to select the region:

      - Use the mouse to select a face, edge, or point on the model in the viewport. If you are working in a mesh, you can use the mouse to select nodes.

- Choose an existing set to specify the face, edge, or point. On the right side of the prompt area, click **Sets**. Select an existing face, edge, or vertex from the **Region Selection** dialog box that appears, and click **Continue**.

  Note:

  The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

It is not necessary to identify the **Region on Main** for a three-dimensional model. However, doing so may help you resolve any problems that you encounter with the analysis. You can select the **Region on Main** in exactly the same way as the **Region on Secondary**.



Note:

If the selected contact interaction has an analytical rigid main surface, the **Region on Main** column appears dimmed, indicating that adding or editing main surface points is unavailable. Any main surface regions that have already been specified will be ignored.

b. Enter the **Critical Contact Pressure** below which fluid will start to penetrate. The higher this value, the easier the fluid penetrates. The default is zero, in which case the fluid penetrates only if contact is lost.

c. Enter the magnitude of the reference **Fluid Pressure**.

If the analysis step is a steady-state dynamic (linear perturbation) step, you can specify both the real (in-phase) and imaginary (out-of-phase) parts of the pressure in the **Fluid Pressure (Real)** and **Fluid Pressure (Imaginary)** columns of the table.

d. To add a row to the table and to continue selecting additional penetration regions, click the  button. This action takes you directly to the region picking step in the viewport. Repeat as needed to specify all the regions.

e. To edit a penetration region, select the region in the table, click the  button, and reselect the region.

f. To delete a region, select the row in the table and click the  button.

6. For a planar (two-dimensional) or axisymmetric model, do the following in the **Penetration Regions** table:

a. Identify the first pair of points exposed to the fluid pressure.

Double-click the empty cell in the **Region on Main** column, or select the cell and click the  button; then use one of the following methods to select the point:

- Use the mouse to select the point on the model in the viewport.
- Choose an existing set to specify the point. On the right side of the prompt area, click **Sets**. Select an existing node or vertex from the **Region Selection** dialog box that appears, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

Repeat the process to select the corresponding penetration point on the secondary surface, using the **Region on Secondary** column of the table.



Note:

If the selected contact interaction has an analytical rigid main surface, the **Region on Main** column appears dimmed, indicating that adding or editing main surface points is unavailable. Any main surface points that have already been specified will be ignored.

b. Enter the **Critical Contact Pressure** below which fluid will start to penetrate. The higher this value, the easier the fluid penetrates. The default is zero, in which case the fluid penetrates only if contact is lost.

c. Enter the magnitude of the reference **Fluid Pressure**.

If the analysis step is a steady-state dynamic (linear perturbation) step, you can specify both the real (in-phase) and imaginary (out-of-phase) parts of the pressure in the **Fluid Pressure (Real)** and **Fluid Pressure (Imaginary)** columns of the table.

d. To add a row to the table and to continue selecting penetration points, click the  button. This action takes you directly to the point picking step in the viewport. Repeat as needed to specify all the points.

e. To edit a penetration point, select the point in the table, click the  button, and reselect the point.

f. To delete a pair of penetration points, select the row in the table and click the  button.

7. In the **Penetration time** field, enter the time period for the fluid pressure penetration to reach the full current magnitude on newly penetrated surface segments. The default penetration time is 0.001 of the current step time. The penetration time is not available in a linear perturbation analysis.
8. Optionally, you can define the variation of the fluid pressure during the step by selecting an amplitude curve in the **Amplitude** list. By default, the reference magnitude is applied immediately at the beginning of the step or ramped up linearly over the step, depending on the amplitude variation assigned to the step. Fluid pressure amplitude curves are not available for some steps.
9. Click **OK** to create the interaction and to close the editor.

Additional information

- *Fluid Pressure Penetration Loads*
- *About Contact Pairs in Abaqus/Standard*
- *Interaction editors*

Defining acoustic impedance

You can model acoustic impedance by providing boundary impedances or nonreflecting boundaries for acoustic and coupled acoustic-structural analyses. Select **Interaction->Create** from the main menu bar, and select the surface to form an acoustic boundary. Acoustic impedance interactions are active only in dynamic steps that utilize the acoustic degree of freedom. You can create an acoustic impedance interaction in a static step; Abaqus ignores the acoustic effects in the static step but propagates the interaction through any applicable linear perturbation steps that follow. If you create an acoustic impedance interaction in a linear perturbation step, the interaction is not propagated to any of the subsequent steps.

For a brief overview of acoustic impedance, see *Understanding interactions*. For a more detailed discussion, see *Acoustic and Shock Loads*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create an acoustic impedance interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. Acoustic impedance is active only in dynamic steps that utilize the acoustic degree of freedom.
   - Select the **Acoustic impedance** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.
5. In **Definition** field in the **Edit Interaction** dialog box that appears, choose one of the following:

- Choose **Tabular** to define impedance using a table of admittance or impedance values in an acoustic impedance property.
- Choose **Nonreflecting** to define nonreflecting boundaries for impedance.

6. If you selected the **Tabular** definition option, select an acoustic impedance property. If desired, click  to create the interaction property; see *Defining an acoustic impedance interaction property*.
7. If you selected the **Nonreflecting** definition option, click the arrow to the right of the **Nonreflecting type** field; and select one of the options from the list that appears to define the nonreflecting geometry.
   - Select **Planar** to specify a radiation condition appropriate for plane waves normally incident to a planar boundary.
   - Select **Improved planar** to specify a radiation condition appropriate for plane waves with arbitrary angles of incidence. In linear perturbation steps, improved planar acoustic impedance interactions default to planar acoustic impedance interactions.
   - Select **Circular** to specify a radiation condition appropriate for a circular boundary in two dimensions or a right circular cylinder in three dimensions.
   - Select **Spherical** to specify a radiation condition appropriate for a spherical boundary.
   - Select **Elliptical** to specify a radiation condition appropriate for an elliptical boundary in two dimensions or a right elliptical cylinder in three dimensions.
   - Select **Prolate spheroidal** to specify a radiation condition appropriate for a prolate spheroidal boundary.
8. If you selected the **Circular** or **Spherical** nonreflecting definition option, enter the radius of the circle or sphere defining the boundary surface in the **Radius** field.
9. If you selected the **Elliptical** or **Prolate spheroidal** nonreflecting definition option, perform the following steps:
   a. In the **Axis length** field, enter the semimajor axis length, *a*, of the ellipse or prolate spheroid defining the surface. *a* is 1/2 of the maximum distance between two points on the ellipse or spheroid, analogous to the radius of a circle or sphere.
   b. In the **Eccentricity** field, enter the eccentricity, ϵ, of the ellipse or prolate spheroid. The eccentricity is the square root of one minus the square of the ratio of the semiminor axis, *b*, to the semimajor axis, *a*: $\epsilon = \sqrt{1-(b/a)^2}$.
   c. In the **Center coordinates** field, enter the *X*-coordinate, *Y*-coordinate, and *Z*-coordinate of the center of the ellipse or prolate spheroid defining the radiating surface.
   d. In the **Direction cosine** field, enter the *X*-component, *Y*-component, and *Z*-component of the direction cosine of the major axis of the ellipse or prolate spheroid defining the radiating surface.

Additional information

• *Acoustic and Shock Loads*
• *Interaction editors*

Defining incident waves

You can model incident wave loading due to external acoustic wave sources.
Select **Interaction->Create** from the main menu bar, and select the source point, the standoff point (for all incident wave loading definitions except for those using the CONWEP model), and the surface. When specifying source points and standoff points, you can select only reference points that are not associated with any other model component. You can create incident wave interactions in the following types of analyses: implicit dynamic, explicit dynamic, direct steady-state dynamic, and subspace-based steady-state dynamic. Incident wave interactions are active only in the steps in which they are created and do not propagate to any of the following steps. If your model includes incident wave interactions, you must edit the model attributes (see below) to specify an acoustic wave formulation for the model.

For a brief overview of incident wave interactions, see *Understanding interactions*. For a more detailed discussion, see *Coupled Acoustic-Structural Analysis*, and *Incident Wave Loading due to External Sources*.

Additional information

- *Coupled Acoustic-Structural Analysis*
- *Acoustic and Shock Loads*
- *Specifying model attributes*
- *Interaction editors*

Specify the acoustic wave formulation for the model

If your model contains incident wave interactions, you must edit the model attributes to specify an acoustic wave formulation. You select a scattered wave formulation or a total wave formulation using the procedure described in *Specifying model attributes*. The formulation that you select applies for all of the incident wave interactions in your model. For more information, see *Scattered and Total Wave Formulations*.

Define an incident wave interaction

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create an incident wave interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define an incident wave interaction only during an implicit, explicit, direct steady-state, or subspace-based steady-state dynamic analysis. An incident wave interaction is active only in the step in which it is created.
   - Select the **Incident wave** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Select the incident wave source point by selecting a reference point that is not associated with any other model component. For more information, see *Creating a reference point*.

   The source point that you select becomes highlighted in red in the viewport.
5. For all incident wave definitions except for those using the CONWEP model, select the incident wave standoff point by selecting a reference point that is not associated with any other model component. For more information, see *Creating a reference point*.

The standoff point that you select becomes highlighted in green in the viewport.

6. For an incident wave definition of air or surface blast data using the definitions of CONWEP data, click **CONWEP (Air/Surface blast)** in the prompt area.

   Abaqus/CAE specifies a **Definition** of **CONWEP** for this incident wave definition.

7. Use one of the following methods to select the surface:

   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport.

   The surface or region that you select becomes highlighted in magenta in the viewport, and the **Edit Interaction** dialog box appears.

8. For all incident wave definitions except for those using the CONWEP model, click the arrow to the **Definition** field and select an option from the list that appears. For direct and subspace-based steady-state dynamic analyses, you can only specify the fluid pressure time history at the standoff point.

   - Select **Pressure** to specify the fluid pressure time history at the standoff point.
   - Select **Acceleration** to specify the fluid particle acceleration time history at the standoff point.
   - Select **UNDEX** to specify UNDEX bubble data.

9. In the **Wave Data** portion of the editor, perform the following steps:

   a. Select an incident wave interaction property. If desired, click  to create the interaction property; see *Defining an incident wave interaction property*, for more information. The following requirements exist:

      - If you selected the **Pressure** definition, you must select an interaction property with a planar definition or a spherical definition with either an acoustic or generalized decay propagation model.
      - If you selected the **Acceleration** definition, you must select an interaction property with a planar definition.
      - If you selected the **UNDEX** definition, you must select an interaction property with a spherical definition and an UNDEX charge propagation model.
      - If you selected the **CONWEP** definition, you must select an interaction property with an air blast or surface blast definition.

   b. In the **Reference magnitude** field, enter the reference magnitude used to scale the values given in the amplitude definition.

10. If you selected the **Pressure** or **Acceleration** definition, perform the following steps from the **Standoff Point** portion of the editor:

    - For an implicit or explicit dynamic analysis, select a **Pressure amplitude** or an **Acceleration amplitude**, depending on the time history definition.

- For a direct or subspace-based steady-state dynamic analysis, toggle on **Real amplitude** and/or **Imaginary amplitude** and select an amplitude.

If desired, click  to create a new amplitude; see *Selecting an amplitude type to define*.

11. If you selected the **UNDEX** definition, perform the following steps in the **UNDEX Data** portion of the editor:
    a. In the **Direction cosine of fluid surface normal** field, enter the *X*-component, *Y*-component, and *Z*-component of the direction cosine of the fluid surface normal.
    b. In the **Initial depth** field, enter the initial depth of the UNDEX charge.
12. If you selected the **CONWEP** definition, perform the following steps in the **CONWEP Data** portion of the editor:
    a. In the **Time of detonation** field, specify the time when the blast occurs in terms of the total time for the entire analysis.
    b. In the **Magnitude scale factor** field, specify the factor by which you want to scale CONWEP data units to analysis data units.
13. Click **OK** to create the interaction and to close the editor.

Defining cyclic symmetry

You can define a cyclic symmetry interaction to model an entire 360° structure using only a single repetitive sector. Select **Interaction->Create**, choose **Cyclic symmetry**, and specify the geometric region by selecting a main surface or node region, a secondary surface or node region, and an axis of symmetry for the structure. You can also control the number of sectors that create the full 360° structure, select the cyclic symmetry nodal diameters for which an eigenfrequency analysis will be performed, and select the cyclic symmetry nodal diameter that will be excited for a steady-state dynamics step.

You can create a cyclic symmetry interaction in the initial step, and only one cyclic symmetry interaction can be active in a model.

For more information about cyclic symmetry in Abaqus, see *Analysis of Models that Exhibit Cyclic Symmetry*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a cyclic symmetry interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the initial step.
   - Select the **Cyclic symmetry** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Select the main surface.
   a. In the prompt area, select one of the following:
      - Select **Surface** if you want to select a surface.
      - Select **Node Region** if you want to select a region from which to create a node-based surface.

   b. Use one of the following methods to select the main surface:
      - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

        Note:

        The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

      - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

        If the model contains a combination of mesh and geometry, click one of the following from the prompt area:
        - Click **Geometry** if you want to select the surface from a geometry region.
        - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

The main surface that you select becomes highlighted in red in the viewport.

5. Select the secondary surface.
   a. In the prompt area, select one of the following:
      - Select **Surface** if you want to select a surface.
      - Select **Node Region** if you want to select a region from which to create a node-based surface.
   b. Use one of the same methods described in the previous step to select the secondary surface or region.

      The secondary surface or region that you select becomes highlighted in magenta in the viewport.

      Abaqus/CAE prompts you to define the axis of symmetry.
6. Use one of the following methods to select the first point on the axis of symmetry:
   - Use an existing set to define the region. The set must contain a single point or vertex. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
   - Use the mouse to select a node or vertex in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:
     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

   The point that you select becomes highlighted in red in the viewport.
7. Using one of the methods described in the previous step, select the second point on the axis of symmetry.

   The point that you select becomes highlighted in magenta in the viewport, and the **Edit Cyclic Symmetry** dialog box opens.
8. From the **Position Tolerance** options, specify the distance within which Abaqus will tie nodes on the secondary surface to the main surface. Choose one of the following options:
   - Select **Use computed distance** to use a distance calculated automatically based on the element formulations and types of surfaces used in the constraints.
   - Select **Specify distance**, and enter a value to set the distance within which secondary nodes will be tied to the main surface.
9. Toggle off **Adjust secondary surface initial position** to prevent Abaqus from moving all tied nodes on the secondary surface onto the main surface.

By default, this option is toggled on and all tied nodes are moved in the initial configuration without applying strains to the model.

10. Specify the **Total number of sectors** that comprise the full 360° structure.
11. From the **Extracted Nodal Diameters** options, specify the range of cyclic symmetry nodal diameters for which the eigenfrequency analysis will be performed. You can define this range in the initial step or in a frequency extraction step only. Specify this range using either of the following options:
    - Select **All possible nodal diameters** to extract every possible cyclic symmetry nodal diameter.
    - Select **Specified range** to extract every cyclic symmetry nodal diameter between a **Lowest nodal diameter** and **Highest nodal diameter** that you specify. The highest nodal diameter must be less than or equal to half the number of sectors.
12. Specify the **Excited nodal diameter** associated with the loading in the load definition. You can select the excited nodal diameter in the initial step or in a mode-based steady-state dynamics step. Only one cyclic symmetry nodal diameter can be excited.
13. Click **OK**.

    Abaqus/CAE creates the cyclic symmetry interaction.

Additional information

• *Analysis of Models that Exhibit Cyclic Symmetry*

• *Interaction editors*

Defining foundations

You can model an elastic foundation by defining the foundation stiffness per area of a selected surface. Select **Interaction->Create** from the main menu bar, and select the surface to be modeled as an elastic foundation. For a brief overview of elastic foundations, see *Understanding interactions*. For a more detailed discussion, see *Element Foundations*.

Elastic foundations allow you to model the stiffness effects of a distributed support without actually modeling the details of the support. You can create elastic foundation interactions only in the initial step. Once an elastic foundation is activated, you cannot deactivate it in later analysis steps.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create an elastic foundation interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the initial step.
   - Select the **Elastic foundation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)
   - If the model contains a combination of mesh and geometry, click one of the following from the prompt area:
     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.
5. In the text field that appears in the prompt area, enter the foundation stiffness per area.

   Abaqus/CAE creates the elastic foundation interaction.

Additional information

- *Element Foundations*
- *Interaction editors*

Defining a cavity radiation interaction

This section describes how you can create a cavity radiation interaction in Abaqus/CAE and use cavity symmetry to reduce the expense of the calculations.

In this section:

- *Defining cavity radiation properties and view factors*
- *Defining cavity radiation symmetry*

Defining cavity radiation properties and view factors

You can model heat transfer due to radiation in enclosures by creating a cavity radiation interaction. Select **Interaction->Create** from the main menu bar, and select the surface. For a brief overview of cavity radiation, see *Understanding interactions*. For a more detailed discussion, see *Cavity Radiation in Abaqus/Standard*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a cavity radiation interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define cavity radiation only during a heat transfer or coupled thermal-electrical step.
   - Select the **Cavity radiation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the methods below to select the cavity surface. Select only the portion of the surface to which one *cavity radiation interaction property* applies.
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface or vertex from a geometry region.
     - Click **Mesh** if you want to select the surface or node from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   Abaqus/CAE displays the **Edit Interaction** dialog box, and the surface name or `(Picked)` appears in the **Surface** column on the **Properties** tabbed page.
5. To define additional cavity surfaces, click mouse button 3 in the table, select **Add Row**, and do one of the following:
   - Double-click an empty cell in the **Surfaces** column.
   - Click mouse button 3 in an empty table row, and select **Edit Surface**.



Note:

You can also use these techniques to replace an existing cavity surface; Abaqus/CAE does not indicate or retain your original selection when you edit an existing surface.

6. Choose the cavity type in the **Definition** field.
   - Select **Closed** to specify a set of closed surfaces for radiation.
   - Select **Open** to include some radiation to the surroundings, and specify an **Ambient temperature** for the open cavity definition.

7. Specify the **Properties** options.
   a. Select the **Blocking surface checks**. By default, Abaqus checks for blocking surfaces within the cavity when performing radiation view factor calculations.
      - Choose **All** to indicate that all blocking checks are active.
      - Choose **None** to skip blocking checks.
      - Choose **Partial** to specify potential blocking surfaces that Abaqus should check.

      For more information, see *Controlling Checks for Surface Blocking*.

   b. If you chose the **Partial** option, double-click an empty cell in the **Blocking Surface** table to select surfaces that you want Abaqus to check.

      Tip: You can also click mouse button 3 in the **Blocking Surface** table and choose **Edit Surface**, **Add Row**, or **Delete Row** to edit the table.

      The surface selection methods are the same as those described in Step 4.

   c. Select the heat reflection behavior.
      - Choose **Yes** for gray body radiation. Gray bodies have an emissivity between zero and one, as defined by a cavity radiation interaction property.
      - Choose **No** for black body radiation. Black bodies have a fixed emissivity of one—no heat is reflected.

   d. If you chose gray body radiation in the previous step, you must specify a cavity radiation interaction property for each row in the table. You can use any of the following methods:
      - Click a cell in the **Property** column to choose a predefined cavity radiation interaction property.
      - Click mouse button 3 in the table, and select **Create Property** to create a new interaction property or **Edit Property** to edit the existing cavity radiation interaction property in the selected row. (For more information, see *Defining a cavity radiation interaction property*.)
      - Enter a value in the **Emissivity** column. Abaqus/CAE automatically creates a cavity radiation interaction property with a default name and the specified emissivity.

   When you are finished, each row in the table contains a surface and an emissivity value or `(table)`, indicating that there is a tabular emissivity defined in the interaction property. **Property** cells may be empty if you entered an emissivity value—Abaqus/CAE will add a default interaction property name when you close the **Edit Interaction** dialog box.

8. Specify the **View factors** options.
   a. If desired, toggle on **Specify blocking range** and enter a value for the distance beyond which Abaqus should not calculate view factors due to blocking effects.

b. Specify the **Accuracy tolerance**. The default value is 0.05.

The view factor tolerance indicates the allowable deviation from the ideal sum of view factors. Abaqus ends the analysis if the tolerance is exceeded for a closed cavity. If the tolerance is exceeded for an open cavity, radiation to the ambient environment occurs.

c. Specify the **Infinitesimal facet area ratio**. The default value is 64.

This value represents the ratio of the largest facet area to the smallest facet area. Abaqus calculates this ratio for each facet pair.

d. Specify the **Gauss integration points per edge**. The default value is 3.

This value is used for the numerical integration method. Possible values range from 1 to 5.

e. Specify the **Lumped area distance-square value**. The default value is 5.

This value represents the ratio of the distance between the centroids of each facet pair, squared, to the area of the larger facet. If the calculated value is greater than this setting, Abaqus uses a lumped area approximation for the integration.

f. If desired, click the **Defaults** button to reset all view factor entries to the Abaqus default values.

If the calculated value for a pair of facets is less than or equal to the **Lumped area distance-square value** setting and the **Infinitesimal facet area ratio** is exceeded, Abaqus uses an infinitesimal-to-finite area approximation. If the **Lumped area distance-square value** is exceeded but the facet area ratio is not exceeded, Abaqus completes a numerical integration of the contour integral to get an accurate value.

For more information, see *Controlling the Accuracy of View Factor Calculations*.

9. Specify the **Symmetry** options. For detailed information on the available cavity radiation symmetry types, see *Defining cavity radiation symmetry*."

10. Specify the absolute zero temperature, θ^Z, on the temperature scale being used and the Stefan-Boltzmann constant, σ, in the **Edit Model Attributes** dialog box, as described in *Specifying model attributes*.

11. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Defining a cavity radiation interaction property*
- *Cavity Radiation in Abaqus/Standard*

Defining cavity radiation symmetry

Using symmetry reduces the computational size of your cavity model. The available symmetries and combinations vary according to the model type. The table in *Combining Symmetries*, indicates the available symmetry combinations. You can use the following symmetry types to complete the cavity definition:

Reflection

Select a **Number of reflection symmetries**, and select a reference z-symmetry value (axisymmetric models), a symmetry axis (two-dimensional models), or a symmetry plane (three-dimensional models) for each reflection.

Abaqus/CAE adds the reflected surfaces to the cavity definition and reduces the remaining number of symmetries allowed in the model. Toggle on **Highlight** to view the current parameter selections in the viewport.

Periodic

Select a **Number of periodic symmetries** and number of repetitions for each periodic symmetry. For axisymmetric models, select a reference z-symmetry value and a periodic z-distance value. For two-dimensional and three-dimensional models, select a symmetry axis and a distance vector or a symmetry plane and a distance vector, respectively, for each periodic symmetry.

Abaqus/CAE adds the periodic surfaces to the cavity surfaces you selected for the **Properties** options and reduces the remaining number of symmetries allowed in the model. Toggle on **Highlight** to view the current parameter selections in the viewport.

Cyclic

Toggle on **Use cyclic symmetric** and select the total number of sectors. Select the symmetry point and a point on the axis of symmetry (two-dimensional models) or the first and second points on the axis of symmetry and a point on the symmetry plane (three-dimensional models).

Cyclic symmetry creates new sectors by rotating the original geometry clockwise about the axis of symmetry. Cyclic symmetry is not available for axisymmetric models.

The following conditions must be met:

- For two-dimensional models the selected point on the axis of symmetry must be on the clockwise side of the geometry defining the original sector.
- For three-dimensional models the selected point on the symmetry plane must be on the counterclockwise side of the geometry defining the original sector.
- The total number of sectors must define a complete circle (360°). If you change the number of sectors, you must redefine the geometry to represent the correct portion of the model.

For more information on cyclic symmetry, including figures showing the sector definitions, see *Cyclic Symmetry*.

Warning:

Abaqus/CAE does not check that the defined symmetries produce a physically realistic cavity model.

Additional information

- *Interaction editors*
- *Defining a cavity radiation interaction property*
- *Cavity Radiation in Abaqus/Standard*

Defining a surface film condition interaction

You can model heat transfer from surfaces due to convection by creating a surface film condition interaction. Select **Interaction->Create** from the main menu bar and select the surface. For a brief overview of film conditions, see *Understanding interactions*. For a more detailed discussion, see *Thermal Loads*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a surface film condition interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define convection from a surface only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step.
   - Select the **Surface film condition** type of interaction.

3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface or vertex from a geometry region.
     - Click **Mesh** if you want to select the surface or node from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

5. In the **Edit Interaction** dialog box that appears, click the arrow to the right of the **Definition** field, and select an option from the list that appears:
   - Select **Embedded Coefficient** to specify the film coefficient in this dialog box.
   - Select **Property Reference** to define the film coefficient as a function of temperature and field variables using a film condition interaction property.
   - Select **User-defined** to define nonuniform film coefficients in user subroutine *FILM*. (This option is valid only in Abaqus/Standard analyses). See the following sections for more information:

- *Specifying general job settings*
- *FILM*

- Select an analytical field to define a spatially varying film coefficient. The analytical field does not affect the sink temperature. Only analytical fields that are valid for this interaction type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

6. If you selected the **Embedded Coefficient** or analytical field definition option, perform the following steps:
   a. In the **Film coefficient** field, enter the film coefficient, *h*.
   b. If you want to vary the film coefficient with time, click the arrow to the right of the **Film coefficient amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define* for more information.
   c. In the **Sink temperature** field, enter the sink temperature, θ^0.
   d. If you want to define a spatially varying sink temperature, click the arrow to the right of the **Sink definition** field and select an analytical field, labeled with an (A), or a discrete field, labeled with a (D). Only analytical fields and discrete fields that are valid for temperature are available in the selection list. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information. Alternatively, you can click  to create a new discrete field.
   e. If you want to vary the sink temperature with time, click the arrow to the right of the **Sink amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define* for more information.
7. If you selected the **Property Reference** definition option, perform the following steps:
   a. Select a film interaction property. If desired, click  to create the interaction property; see *Defining a film condition interaction property* for more information.
   b. In the **Sink temperature** field, enter the sink temperature, θ^0.
   c. If you want to define a spatially varying sink temperature, click the arrow to the right of the **Sink definition** field and select an analytical field, labeled with an (A), or a discrete field, labeled with a (D). Only analytical fields and discrete fields that are valid for temperature are available in the selection list. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information. Alternatively, you can click  to create a new discrete field.
   d. If you want to vary the sink temperature with time, click the arrow to the right of the **Sink amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define* for more information.
8. If you selected the **User-defined** definition option, perform the following steps:
   a. In the **Film coefficient** field, enter the film coefficient, *h*.
   b. In the **Sink temperature** field, enter the sink temperature, θ^0.
   c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine *FILM*. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

9. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Using analytical expression fields*
- *Creating expression fields*
- *Thermal Loads*

Defining a concentrated film condition interaction

You can model heat transfer from one or more points in an assembly due to convection by creating a concentrated film condition interaction. Select **Interaction->Create** from the main menu bar and select one or more nodes or vertices or a saved set of nodes or vertices. For a brief overview of film conditions, see *Understanding interactions*. For a more detailed discussion, see *Thermal Loads*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a concentrated film condition interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define convection from a nodal area only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step.
   - Select the **Concentrated film condition** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the points:
   - Use an existing set of nodes or vertices to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   - Use the mouse to select nodes or vertices in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select vertices from a geometry region.
     - Click **Mesh** if you want to select nodes from a native or orphan mesh selection.

     You can use the angle method to select a group of nodes from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.
5. In the **Edit Interaction** dialog box that appears, click the arrow to the right of the **Definition** field and select an option from the list that appears:
   - Select **Embedded Coefficient** to specify the film coefficient in this dialog box.
   - Select **Property Reference** to define the film coefficient as a function of temperature and field variables using a film condition interaction property.

- Select **User-defined** to define nonuniform film coefficients in user subroutine *FILM*. (This option is valid only in Abaqus/Standard analyses). See the following sections for more information:
  - *Specifying general job settings*
  - *FILM*
- Select an analytical field to define a spatially varying film coefficient. The analytical field does not affect the sink temperature. Only analytical fields that are valid for this interaction type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

6. If desired, specify how the concentrated film condition is applied to the boundary of an adaptive mesh domain. This option is valid only for Abaqus/Explicit analyses. Click the arrow to the right of the **Adaptive mesh boundary type** field, and select an option from the list that appears. For more information, see *Defining ALE Adaptive Mesh Domains in Abaqus/Explicit*.
   - Select **Lagrangian** to apply a concentrated film to a node that follows the material (nonadaptive).
   - Select **Sliding** to apply a concentrated film to a node that can slide over the material. Mesh constraints are typically applied to the node to fix it spatially.
   - Select **Eulerian** to apply a concentrated film to a node that can move independently of the material. This option is used only for boundary regions where the material can flow into or out of the adaptive mesh domain. Mesh constraints must be used normal to an Eulerian boundary region to allow material to flow through the region. If no mesh constraints are applied, an Eulerian boundary region will behave in the same way as a sliding boundary region.
7. In the **Associated nodal area** field, enter the area associated with the node where the concentrated film condition is applied.
8. If you selected the **Embedded Coefficient** or analytical field definition option, perform the following steps:
   a. In the **Film coefficient** field, enter the film coefficient, h.
   b. If you want to vary the film coefficient with time, click the arrow to the right of the **Film coefficient amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define*, for more information.
   c. In the **Sink temperature** field, enter the sink temperature, θ^0.
   d. If you want to define a spatially varying sink temperature, click the arrow to the right of the **Sink definition** field and select an analytical field, labeled with an (A), or a discrete field, labeled with a (D). Only analytical fields and discrete fields that are valid for temperature are available in the selection list. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information. Alternatively, you can click  to create a new discrete field.
   e. If you want to vary the sink temperature with time, click the arrow to the right of the **Sink amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define*, for more information.
9. If you selected the **Property Reference** definition option, perform the following steps:
   a. Select a film interaction property. If desired, click  to create the interaction property; see *Defining a film condition interaction property*, for more information.

b. In the **Sink temperature** field, enter the sink temperature, θ^0.

c. If you want to define a spatially varying sink temperature, click the arrow to the right of the **Sink definition** field and select an analytical field, labeled with an (A), or a discrete field, labeled with a (D). Only analytical fields and discrete fields that are valid for temperature are available in the selection list. See *The Analytical Field toolset* and *The Discrete Field toolset* for more information.

Alternatively, you can click  to create a new discrete field.

d. If you want to vary the sink temperature with time, click the arrow to the right of the **Sink amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define* for more information.

10. If you selected the **User-defined** definition option, perform the following steps:

a. In the **Film coefficient** field, enter the film coefficient, *h*.

b. In the **Sink temperature** field, enter the sink temperature, θ^0.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine *FILM*. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

11. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Using analytical expression fields*
- *Creating expression fields*
- *Defining ALE Adaptive Mesh Domains in Abaqus/Explicit*
- *Thermal Loads*

Defining a surface radiative interaction

You can model heat transfer between a nonconcave surface and a nonreflecting environment due to radiation by creating a surface radiation interaction. Surface radiation can also be used to approximate cavity radiation for a closed cavity in three-dimensional models. Select **Interaction->Create** from the main menu bar, and select the surface. For a brief overview of radiative interactions, see *Understanding interactions*. For a more detailed discussion, see *Thermal Loads*. For more information about cavity radiation, see *Defining a cavity radiation interaction* and *Cavity Radiation in Abaqus/Standard*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a surface radiative interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define radiation from a surface only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step.
   - Select the **Surface radiation** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the surface:
   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface or vertex from a geometry region.
     - Click **Mesh** if you want to select the surface or node from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.
5. In the **Edit Interaction** dialog box that appears, select the **Radiation type**.
   - Select **To ambient** to specify heat transfer to the surrounding environment.
   - Select **Cavity approximation (3D only)** to approximate cavity radiation in three-dimensional models using uniform emissivity, a closed cavity, and an average cavity temperature.

6. If you selected **To ambient** in the previous step, complete the radiation definition as follows:
   a. Click the arrow to the right of the **Emissivity distribution** field, and select the option of your choice from the list that appears:
      - Select **Uniform** to define an emissivity that is uniform over the surface.
      - Select an analytical field to define a spatially varying emissivity. Only analytical fields that are valid for this interaction type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
   b. In the **Emissivity** field, enter the emissivity of the surface, ϵ.
   c. In the **Ambient temperature** field, enter the ambient temperature, θ^0.
   d. If you want to vary the ambient temperature with time, click the arrow to the right of the **Ambient temperature amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define* for more information.
7. If you selected **Cavity approximation (3D only)** in Step 5, enter the emissivity of the surface, ϵ, in the **Emissivity** field.
8. Specify the absolute zero temperature, θ^z, on the temperature scale being used and the Stefan-Boltzmann constant, σ, in the **Edit Model Attributes** dialog box, as described in *Specifying model attributes*.
9. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Defining a cavity radiation interaction*
- *Using analytical expression fields*
- *Creating expression fields*
- *Thermal Loads*
- *Cavity Radiation in Abaqus/Standard*

Defining a concentrated radiative interaction

You can model heat transfer between one or more points in an assembly and a nonreflecting environment due to radiation by creating a concentrated radiation to ambient interaction. Select **Interaction->Create** from the main menu bar, and select one or more nodes or vertices or a saved set of nodes or vertices. For a brief overview of radiative interactions, see *Understanding interactions*. For a more detailed discussion, see *Thermal Loads*.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a concentrated radiative interaction using the tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step. You can define radiation from a nodal area only during a heat transfer, coupled temperature-displacement, or coupled thermal-electrical step.
   - Select the **Concentrated radiation to ambient** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. Use one of the following methods to select the points:
   - Use an existing set of nodes or vertices to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   - Use the mouse to select nodes or vertices in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select vertices from a geometry region.
     - Click **Mesh** if you want to select nodes from a native or orphan mesh selection.

     You can use the angle method to select a group of nodes from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.
5. In the **Edit Interaction** dialog box that appears, perform the following steps:
   a. If desired, specify how the concentrated radiation condition is applied to the boundary of an adaptive mesh domain. This option is valid only for Abaqus/Explicit analyses. Click the arrow to the right of the **Adaptive mesh boundary type** field, and select an option from the list that appears. For more information, see *Defining ALE Adaptive Mesh Domains in Abaqus/Explicit*.
      - Select **Lagrangian** to apply a concentrated radiation condition to a node that follows the material (nonadaptive).

- Select **Sliding** to apply a concentrated radiation condition to a node that can slide over the material. Mesh constraints are typically applied to the node to fix it spatially.
- Select **Eulerian** to apply a concentrated radiation condition to a node that can move independently of the material. This option is used only for boundary regions where the material can flow into or out of the adaptive mesh domain. Mesh constraints must be used normal to an Eulerian boundary region to allow material to flow through the region. If no mesh constraints are applied, an Eulerian boundary region will behave in the same way as a sliding boundary region.

b. In the **Associated nodal area** field, enter the area associated with the node where the concentrated radiation condition is applied.

c. Click the arrow to the right of the **Emissivity distribution** field, and select the option of your choice from the list that appears:

- Select **Uniform** to define an emissivity that is uniform over the region.
- Select an analytical field to define a spatially varying emissivity. Only analytical fields that are valid for this interaction type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

d. In the **Emissivity** field, enter the emissivity of the surface, ϵ.

e. In the **Ambient temperature** field, enter the ambient temperature, θ^0.

f. If you want to vary the ambient temperature with time, click the arrow to the right of the **Ambient temperature amplitude** field and select an amplitude from the list that appears. If desired, click  to create a new amplitude; see *Selecting an amplitude type to define*, for more information.

6. Specify the absolute zero temperature, θ^Z, on the temperature scale being used and the Stefan-Boltzmann constant, σ, in the **Edit Model Attributes** dialog box, as described in *Specifying model attributes*.
7. Click **OK** to create the interaction and to close the editor.

Additional information

- *Interaction editors*
- *Using analytical expression fields*
- *Creating expression fields*
- *Thermal Loads*

Defining an actuator/sensor interaction

You can create an actuator/sensor interaction at a single vertex of your model. An actuator/sensor interaction provides an interface to user subroutine *UEL*. The subroutine, in turn, represents a linear or nonlinear user-defined element. Actuator/sensor interactions must be defined during the initial step and are valid only for Abaqus/Standard analyses.

Warning:

This feature is intended for advanced users only. Its use in all but the simplest test examples will require considerable coding by the user/developer. *User-Defined Elements*, should be read before proceeding.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create an actuator/sensor interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the initial step.
   - Select the **Actuator/sensor** type of interaction.
3. Click **Continue** to close the **Create Interaction** dialog box.
4. From the assembly, select the point where the interaction will be applied. Click mouse button 2 to indicate you have finished selecting the point.

   Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

   Abaqus/CAE displays the **Edit Interaction** dialog box.
5. Enter the necessary data in the **Edit Interaction** dialog box. The required data are a function of your user-defined element subroutine. You may need to create real and integer actuator/sensor interaction properties. For more information, see *Defining an actuator/sensor interaction property*.

   The following table indicates the correspondence between the fields in the **Edit Interaction** dialog box and the variables in user subroutine *UEL*.

Field	UEL variable
User element type id	`JTYPE`
Degrees of freedom	`NDOFEL`
Number of coordinate components	`MCRD`
Solution-dependent state variables	`SVARS` and `NSVARS`

The real and integer values entered into an actuator/sensor interaction property are used by the `PROPS`, `JPROPS`, `NPROPS`, and `NJPROPS` variables. For a description of all the variables that can be passed into user subroutine *UEL*, see *User-Defined Elements*.

Additional information

- *User-Defined Elements*
- *UEL*
- *Interaction editors*

Defining contact mass scaling

In Abaqus/Explicit contact mass scaling can be defined for all steps, and this contact mass scaling definition is active for all subsequent steps.

A contact mass scaling definition can apply mass scaling to the contact surfaces/elements involved in the contact definition in Abaqus/Explicit only.

1. From the main menu bar, select **Interaction->Create**.

   Tip: You can also create a general contact interaction using the  tool in the Interaction module toolbox.

2. In the **Create Interaction** dialog box that appears, do the following:
   - Name the interaction. For more information about naming objects, see *Using basic dialog box components*.
   - Select the step in which the interaction will be created. Contact mass scaling can be created in any explicit step in which contact is already defined.
   - Select the **Contact Mass Scaling (Explicit)** type of interaction.

3. Click **Continue** to close the **Create Interaction** dialog box.

   The **Edit Interaction** dialog box appears.

4. Specify the Location (The Default is Element Mass Scaling)
5. If the location selected is Specified Surfaces, then select the surfaces to include in contact mass scaling from the list of available surfaces.

   a. Click the arrows 

   in the middle of the dialog box to transfer the surfaces to the list of surfaces that will be included in the contact mass scaling.

6. Click **OK** to create the interaction and to close the editor.

Additional information

- **CONTACT MASS SCALING*

Using the interaction property editors

This section explains how to enter data in the interaction property editors to define specific types of interaction properties.

In this section:

- *Defining a contact interaction property*
- *Defining a film condition interaction property*
- *Defining a cavity radiation interaction property*
- *Defining a fluid cavity interaction property*
- *Defining a fluid exchange interaction property*
- *Defining an acoustic impedance interaction property*
- *Defining an incident wave interaction property*
- *Defining an actuator/sensor interaction property*
- *Defining a fluid inflator interaction property*
- *Defining a wear interaction property*

The contact property editor contains the following menus from which you can choose options to include in the property definition: **Mechanical**, **Thermal**, and **Electrical**.

A contact interaction property can be referred to by a general contact, surface-to-surface contact, or self-contact interaction. For more information, see *About Mechanical Contact Properties*, *Thermal Contact Properties*, and *Electrical Contact Properties*.

The **Contact Property Options** list at the top of the editor displays the options currently included in the property definition; the list is updated as you add and delete options. You can add, delete, or change property options as follows:

Adding property options

Select the options needed to define your property from the **Mechanical**, **Thermal**, and **Electrical** menus. When you select an option, its name appears in the **Contact Property Options** list, and data fields associated with the option appear in the data area in the bottom half of the editor. Use the data fields to enter information for the currently selected option.

Deleting property options

In the **Contact Property Options** list, select the option that you want to delete, and click **Delete** on the right side of the editor. This procedure removes the option from both the options list and the property definition.

Changing option data

In the **Contact Property Options** list, select the option whose data you want to change. When the data fields associated with the option appear in the bottom half of the window, change the information that you have entered for the option as desired.

In this section:

- *Defining mechanical contact property options*
- *Defining thermal contact property options*
- *Specifying gap conductance for electrical contact property options*

Defining mechanical contact property options

You can define mechanical contact property options to specify tangential behavior (friction and elastic slip), normal behavior (hard, soft, or damped contact and separation), and damping due to friction.

In this section:

- *Specifying frictional behavior for mechanical contact property options*
- *Specifying pressure-overclosure relationships for mechanical contact property options*
- *Specifying damping for mechanical contact property options*
- *Specifying cohesive behavior properties for mechanical contact property options*
- *Specifying cohesive damage properties for mechanical contact property options*
- *Specifying fracture criterion properties for crack propagation*
- *Specifying geometric properties for mechanical contact property options*

Specifying frictional behavior for mechanical contact property options

You can specify a friction model that defines the force resisting the relative tangential motion of the surfaces in a mechanical contact analysis. For more information, see *Frictional Behavior*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Tangential Behavior**.
5. In the editor that appears, click the arrow to the right of the **Friction formulation** field, and select how you want to define friction between the contact surfaces:
   - Select **Frictionless** if you want Abaqus to assume that surfaces in contact slide freely without friction.
   - Select **Penalty** to use a stiffness (penalty) method that permits some relative motion of the surfaces (an "elastic slip") when they should be sticking. While the surfaces are sticking (i.e., $\tau_{eq} < \tau_{crit}$), the magnitude of sliding is limited to this elastic slip. Abaqus will continually adjust the magnitude of the penalty constraint to enforce this condition. For more information, see *Stiffness Method for Imposing Frictional Constraints in Abaqus/Standard* and *Stiffness Method for Imposing Frictional Constraints in Abaqus/Explicit*.
   - Select **Static-Kinetic Exponential Decay** to specify static and kinetic friction coefficients directly. In this model it is assumed that the friction coefficient decays exponentially from the static value to the kinetic value. Alternatively, you can enter test data to fit the exponential model. (This **Friction formulation** option also allows you to specify elastic slip.) For more information, see *Specifying Static and Kinetic Friction Coefficients*.
   - Select **Rough** to specify an infinite coefficient of friction. For more information, see *Preventing Slipping regardless of Contact Pressure*.
   - Select **Lagrange Multiplier (Standard only)** to enforce the sticking constraints at an interface between two surfaces using the Lagrange multiplier implementation. With this method there is no relative motion between two closed surfaces until $\tau_{eq} = \tau_{crit}$. For more information, see *Lagrange Multiplier Method for Imposing Frictional Constraints in Abaqus/Standard*.
   - Select **User-defined** to define the shear interaction between the contact surfaces with user subroutine *FRIC* or *VFRIC*. For more information, see *User-Defined Friction Model*.
6. If you selected the **Penalty** or **Lagrange Multiplier (Standard only)** friction formulation, perform the following steps:
   a. Display the **Friction** tabbed page.
   b. Choose the **Directionality**:
      - Choose **Isotropic** to enter a uniform friction coefficient.
      - Choose **Anisotropic (Standard only)** to allow for different friction coefficients in the two orthogonal directions on the contact surface. For more information, see *Anisotropic Friction with Directional Preference Associated with Contact Orientation*.
   c. Toggle on **Use slip-rate-dependent data** if the friction coefficient is dependent on slip rate.

d. Toggle on **Use contact-pressure-dependent data** if the friction coefficient is dependent on the contact pressure.

e. Toggle on **Use temperature-dependent data** if the friction coefficient is dependent on temperature.

f. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the friction coefficient depends.

g. Enter the required data in the data table provided.

h. Display the **Shear Stress** tabbed page, and choose a **Shear stress limit** option:

- Choose **No limit** if you do not want to limit the shear stress that can be carried by the interface before the surfaces begin to slide.
- Choose **Specify** to enter an equivalent shear stress limit, τ_{max}. If you choose this option, sliding will occur if the magnitude of the equivalent shear stress reaches this value, regardless of the magnitude of the contact pressure stress. For more information, see *Using the Optional Shear Stress Limit*.

i. If you selected the **Penalty** friction formulation, display the **Elastic Slip** tabbed page, and specify how you want to define elastic slip:

- If you are performing an Abaqus/Standard analysis, choose an option to **Specify maximum elastic slip**:
  - Choose **Fraction of characteristic surface dimension** to calculate the allowable elastic slip as a small fraction of the characteristic contact surface length.
  - Choose **Absolute distance** to enter the absolute magnitude of the allowable elastic slip, γ_t. (For a steady-state transport analysis set this parameter equal to the absolute magnitude of the allowable elastic slip velocity ($\dot{\gamma}_t$) to be used in the stiffness method for sticking friction.)
- If you are performing an Abaqus/Explicit analysis, choose an **Elastic slip stiffness** option:
  - Choose **Infinite (no slip)** to deactivate shear softening.
  - Choose **Specify** to activate softened tangential behavior. Enter the slope of the curve that defines the shear traction as a function of the elastic slip between the two surfaces.

For more information, see *Shear Stress Versus Elastic Slip While Sticking*.

7. If you selected the **Static-Kinetic Exponential Decay** friction formulation, perform the following steps:

a. Display the **Friction** tabbed page.

b. Choose an option for defining the exponential decay friction model:

- Choose **Coefficients** to provide the static friction coefficient, the kinetic friction coefficient, and the decay coefficient directly.
- Choose **Test data** to provide test data points to fit the exponential model.

For more information, see *Specifying Static and Kinetic Friction Coefficients*.

c. If you selected the **Coefficients** definition option, enter the following in the data table provided:

- Static friction coefficient, μ_s.
- Kinetic friction coefficient, μ_k.
- Decay coefficient, d_c.

If you selected the **Test data** definition option, enter the following in the data table provided:

- In the first row, enter the static friction coefficient, μ_1.
- In the second row, enter the dynamic friction coefficient, μ_2 and the reference slip rate, $\dot{\gamma}_2$, at which μ_2 is measured.
- In the third row, enter the kinetic friction coefficient, μ_∞. This value corresponds to the asymptotic value of the friction coefficient at infinite slip rate, $\dot{\gamma}_\infty$. If this data line is omitted, Abaqus/Standard automatically calculates μ_∞ such that $(\mu_2 - \mu_\infty) / (\mu_1 - \mu_\infty) = 0.05$.

d. Display the **Elastic Slip** tabbed page, and specify how you want to define elastic slip:

- If you are performing an Abaqus/Standard analysis, choose an option to **Specify maximum elastic slip**:
  - Choose **Fraction of characteristic surface dimension** to calculate the allowable elastic slip as a small fraction of the characteristic contact surface length.
  - Choose **Absolute distance** to enter the absolute magnitude of the allowable elastic slip, γ_i. (For a steady-state transport analysis set this parameter equal to the absolute magnitude of the allowable elastic slip velocity ($\dot{\gamma}_i$) to be used in the stiffness method for sticking friction.)
- If you are performing an Abaqus/Explicit analysis, choose an **Elastic slip stiffness** option:
  - Choose **Infinite (no slip)** to deactivate shear softening.
  - Choose **Specify** to activate shear softening. Enter the slope of the curve that defines the shear traction as a function of the elastic slip between the two surfaces.

For more information, see *Shear Stress Versus Elastic Slip While Sticking*.

8. If you selected the **User-defined** friction formulation, perform the following steps:

a. Click the arrows to the right of the **Number of state-dependent variables** field to indicate the number state variables that will be defined in user subroutine *FRIC* or *VFRIC*.

b. In the **Friction Properties** table, enter the values of properties needed by user subroutine *FRIC* or *VFRIC*. (For detailed information on how to enter data, see *Entering tabular data*.)

For more information, see *User-Defined Friction Model*.

9. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying pressure-overclosure relationships for mechanical contact property options

You can define a constitutive model for the contact pressure-overclosure relationship that governs the motion of the surfaces in a mechanical contact analysis. For more information, see *Contact Pressure-Overclosure Relationships*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Normal Behavior**.
5. From the **Pressure-Overclosure** field, select **"Hard" Contact** to use the classical Lagrange multiplier method of constraint enforcement in an Abaqus/Standard analysis and to use penalty contact enforcement in an Abaqus/Explicit analysis.

   You can also toggle off **Allow separation after contact** if you want to prevent surfaces from separating once they have come into contact. This method is applicable only to Abaqus/Standard analyses.

   If you select **"Hard" Contact**, you can also customize settings for the constraint enforcement method. For more information about constraint enforcement methods, see *Contact Constraint Enforcement Methods in Abaqus/Explicit*. To specify these settings, select an option from the **Constraint enforcement method** list and do the following:

   a. Select **Default** to enforce constraints using a contact pressure-overclosure relationship.
   b. Select **Augmented Lagrange (Standard)** to enforce contact constraints using the augmented Lagrange method. This method is applicable only to Abaqus/Standard analyses.

      If you select this option, specify the following additional settings from the **Contact Stiffness** options:

      - From the **Stiffness value** field, either select **Use default** to have Abaqus calculate the penalty contact stiffness automatically or select **Specify** and enter a positive value for the penalty contact stiffness.
      - Specify a factor by which to multiply the chosen penalty stiffness in the **Stiffness scale factor** field.
      - Specify the **Clearance at which contact pressure is zero**. The default value is 0.

   c. Select the **Penalty (Standard)** constraint enforcement method to enforce contact constraints using the penalty method. This method is applicable only to Abaqus/Standard analyses.

      If you select this option, specify the following additional settings from the **Contact Stiffness** options:

      - From the **Behavior** field, either select **Linear** to use the linear penalty method for the enforcement of the contact constraint or select **Nonlinear** to use the nonlinear penalty method for the enforcement of the contact constraint. For more information, see *Penalty Method*.
      - Specify the contact stiffness.
        - For the linear penalty method, specify the contact stiffness in the **Stiffness value** field. You can select **Use default** to have Abaqus calculate the penalty contact stiffness automatically or you can select **Specify** and enter a positive value for the linear penalty stiffness.
        - For the nonlinear penalty method, specify the contact stiffness in the **Maximum stiffness value** field. You can select **Use default** to have Abaqus calculate the penalty contact stiffness

automatically or you can select **Specify** and enter a positive value for the final nonlinear penalty stiffness.

- Specify a factor by which to multiply the chosen penalty stiffness in the **Stiffness scale factor** field.
- For the nonlinear penalty method, you can specify values for the following options:
  - Enter the ratio of the initial penalty stiffness over the final penalty stiffness in the **Initial/Final stiffness ratio** field.
  - Enter the scale factor for the upper quadratic limit d, which is equal to the scale factor times the characteristic contact facet length, in the **Upper quadratic limit scale factor** field.
  - Enter the ratio $(e-c_0)/(d-c_0)$ that defines the lower quadratic limit e in the **Lower quadratic limit ratio** field.
- Specify the **Clearance at which contact pressure is zero**. The default value is 0.

d. Select **Direct (Standard)** to enforce contact constraints directly without approximation or use of augmentation iterations.

6. From the **Pressure-Overclosure** field, select **Exponential** to define an exponential pressure-overclosure relationship. If you select this option, specify the following:

a. Enter the contact pressure at zero clearance, p_0, and the clearance at which the contact pressure is zero, c_0, in the data table.

b. Specify the limit on the contact stiffness that the model can attain, k_{max} (applies only for Abaqus/Explicit analyses).

- Choose **Infinite (no slip)** to set k_{max} equal to infinity for kinematic contact and equal to the default penalty stiffness for penalty contact.
- Choose **Specify**, and enter a value for the maximum stiffness.

7. From the **Pressure-Overclosure** field, select **Linear** to define a linear pressure-overclosure relationship. If you select this option, specify the following:

- Enter a positive value for the slope of the pressure-overclosure curve, *k*, in the **Contact stiffness** field.

8. From the **Pressure-Overclosure** field, select **Tabular** to define a piecewise-linear pressure-overclosure relationship in tabular form. If you select this option, specify the following:

- Enter data in ascending order of overclosure to define the overclosure as a function of pressure. The data table must begin with a zero pressure. The pressure-overclosure relationship is extrapolated beyond the last overclosure point by continuing the same slope.

9. From the **Pressure-Overclosure** field, select **Scale Factor (General Contact, Explicit)** to define a piecewise-linear pressure-overclosure relationship based on scaling the default contact stiffness. This option is available only for the general contact algorithm in Abaqus/Explicit. If you select this option, specify the following:

a. To define the overclosure measure as a percentage of the minimum element size, select **factor** in the **Overclosure** field and enter a positive value r.

b. To define the overclosure measure directly, select **measure** in the **Overclosure** field and enter a positive value d.

c. Enter a value, s, greater than one to define the geometric scaling of the "base" stiffness in the **Contact stiffness scale factor** field.

d. Enter a positive value s_0 to define an additional scale factor for the "base" default contact stiffness in the **Initial stiffness scale factor** field. The default value is 1.

10. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying damping for mechanical contact property options

You can define a damping model that defines forces resisting the relative motions of the contacting surfaces in a mechanical contact analysis. For more information, see *Contact Damping*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Damping**.
5. In the editor that appears, click the arrow to the right of the **Definition** field, and select an option for determining the dimensionality of the damping coefficient:
   - Select **Damping coefficient** to specify the damping coefficient with units of pressure per relative velocity. For more information, see *Specifying the Damping Coefficient Such That the Damping Force Is Directly Proportional to the Rate of Relative Motion between the Surfaces*.
   - Select **Critical damping fraction (Explicit only)** to specify a unitless damping coefficient in terms of the fraction of critical damping associated with the contact stiffness; this method is available only for Abaqus/Explicit. For more information, see *Specifying the Damping Coefficient as a Fraction of Critical Damping in Abaqus/Explicit*.
6. Choose an option for specifying the **Tangent fraction** (the ratio of the tangential damping coefficient to the normal damping coefficient):
   - Choose **Use default** to accept the default tangent fraction value. For Abaqus/Standard the default is 0.0, so the damping coefficient for the tangential direction is zero. For Abaqus/Explicit the default value for the tangent fraction is 1.0, so the damping coefficient for the tangential direction is equal to the damping coefficient for the normal direction.
   - Choose **Specify value** to enter a value for the tangent fraction.

   For more information, see *Specifying the Tangential Damping Coefficient*.
7. Choose a shape for the curve that describes the relationship between clearance and the damping coefficient:
   - Choose **Step (Explicit only)** if you are performing an Abaqus/Explicit analysis. The damping coefficient will remain at the specified constant value while the surfaces are in contact and at zero otherwise.
   - Choose **Linear (Standard only)** to define a damping coefficient that increases linearly from zero at a particular clearance value (c_0) to its full value when the surfaces are in contact.
   - Choose **Bilinear (Standard only)** to define a damping coefficient that increases linearly from zero at a particular clearance value (c_0) to its full value when clearance has been reduced to another value (c). As clearance continues to decrease from c to zero, the damping coefficient remains constant at its full value.
8. Enter the appropriate data in the table provided:
   - If you are performing an Abaqus/Explicit analysis, enter a value for the damping coefficient or for the critical damping fraction (depending on your selection in Step 5.)

- If you are performing an Abaqus/Standard analysis and selected **Linear (Standard only)** in the previous step, enter the following:
  - In the first row, enter a value for the damping coefficient.
  - In the second row, enter a value for c_0, the clearance at which the damping coefficient is zero.
- If you are performing an Abaqus/Standard analysis and selected **Bilinear (Standard only)** in the previous step, enter the following:
  - In the first row, enter a value for the damping coefficient.
  - In the second row, enter a value for c, the clearance at which the damping coefficient reaches its full value.
  - In the third row, enter a value for c_0, the clearance at which the damping coefficient is zero.

9. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying cohesive behavior properties for mechanical contact property options

You can define cohesive behavior properties that are accounted for in surface contact interactions. For more information, see *Contact Cohesive Behavior*.

In addition, you complete the definition of a crack propagation capability by defining a fracture-based cohesive behavior surface interaction. You activate the crack propagation by assigning it to the pair of surfaces that are initially partially bonded. If the fracture criterion is met, crack propagation occurs between these two surfaces. Cohesive behavior is also used to specify the elastic behavior of the bonds.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Cohesive Behavior**.
5. Toggle on **Allow cohesive behavior during repeated post-failure contacts** to modify the default post-failure behavior when progressive damage has been defined. By default, cohesive behavior is not enforced for nodes on the secondary surface once ultimate failure has occurred at those nodes. When this option is toggled on, Abaqus/CAE enforces cohesive behavior for recurrent contacts at nodes on the secondary surface subsequent to ultimate failure.
6. From the **Eligible Secondary Nodes** options, select one of the following:
   - Select **Any secondary nodes experiencing contact** to define cohesive behavior not only for all nodes of the secondary surface that are in contact with the main surface at the start of a step, but also for secondary nodes that are not initially in contact but may come in contact during the course of a step.
   - Select **Only secondary nodes initially in contact** to restrict cohesive behavior to only those nodes of the secondary surface that are in contact with the main surface at the start of a step.
   - Select **Specify the bonding node set in the surface-to-surface Std interaction** to restrict cohesive behavior to a subset of secondary nodes defined when you specify initial bonded contact conditions. This option is available only for Abaqus/Standard analyses.
7. From the **Traction-separation Behavior** options, accept the default contact penalty enforcement method or toggle on **Specify stiffness coefficients** and perform the following additional steps:
   a. Specify whether you want to specify stiffness coefficients for **Uncoupled** or **Coupled** traction behavior.
   b. Toggle on **Use temperature-dependent data** if the traction-separation behavior is dependent on temperature.
   c. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the traction-separation behavior depends.
   d. Enter the required data in the data table provided.
8. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying cohesive damage properties for mechanical contact property options

You can define damage initiation, evolution, and stabilization properties that will be accounted for in surface contact interactions. For more information, see *Contact Cohesive Behavior*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Damage**.
5. From the **Initiation** tabbed page, perform the following steps:
   a. From the **Criterion** list, choose one of the following:
      - Select **Maximum nominal stress** to specify a damage initiation criterion based on the maximum nominal stress criterion for cohesive elements.
      - Select **Maximum separation** to specify a damage initiation criterion based on the maximum separation value.
      - Select **Quadratic traction** to specify a damage initiation criterion based on the quadratic traction–interaction criterion for cohesive elements.
      - Select **Quadratic separation** to specify a damage initiation criterion based on the quadratic separation–interaction criterion for cohesive elements.
   b. Toggle on **Use temperature-dependent data** if the damage initiation behavior is dependent on temperature.
   c. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the damage initiation behavior depends.
   d. Enter the required data in the data table provided.
6. If you want to specify damage evolution criteria; toggle on **Specify damage evolution**, click the **Evolution** tab, and perform the following steps:
   a. From the **Type** options, select one of the following:
      - Select **Displacement** to define the evolution of damage as a function of the total displacement (for elastic materials in cohesive elements) or the plastic displacement (for bulk elastic-plastic materials) after the initiation of damage.
      - Select **Energy** to define the evolution of damage in terms of the energy required for failure (fracture energy) after the initiation of damage.
   b. From the **Softening** options, select one of the following:
      - Select **Linear** to specify a linear softening stress-strain response (after the initiation of damage) for linear elastic materials or a linear evolution of the damage variable with deformation (after the initiation of damage) for elastic-plastic materials.
      - Select **Exponential** to specify an exponential softening stress-strain response (after the initiation of damage) for linear elastic materials or an exponential evolution of the damage variable with deformation (after the initiation of damage) for elastic-plastic materials.

- Select **Tabular** to specify the evolution of the damage variable with deformation (after the initiation of damage) in tabular form. This option is available only for damage evolution defined in terms of displacement.

c. If you want to specify mode-dependent behavior, toggle on **Specify mixed mode behavior** and select one of the following options:

- Select **Tabular** to specify the fracture energy or displacement (total or plastic) directly as a function of the shear-normal mode mix for cohesive elements. This method must be used to specify the mixed-mode behavior for cohesive elements when damage evolution is defined in terms of displacement.
- Select **Power law** to specify the fracture energy as a function of the mode mix by means of a power law mixed-mode fracture criterion.
- Select **Benzeggagh-Kenane** to specify the fracture energy as a function of the mode mix by means of the Benzeggagh-Kenane mixed-mode fracture criterion.

d. If you specified **Tabular** for the mixed mode behavior, select one of the following:

- Select **Energy** to define the mode mix in terms of a ratio of fracture energy in the different modes.
- Select **Traction** to define the mode mix in terms of a ratio of traction components.

e. If you toggled on **Specify mixed mode behavior** and selected either **Power law** or **Benzeggagh-Kenane** as the fracture criterion, you can specify the exponent in the power law or the Benzeggagh-Kenane criterion that defines the variation of fracture energy with mode mix for cohesive elements. Toggle on **Specify power-law/criterion** and enter a value for the exponent in the field.

f. Toggle on **Use temperature-dependent data** if the damage evolution behavior is dependent on temperature.

g. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the damage evolution behavior depends.

h. Enter the required data in the data table provided.

7. If you want to specify viscous regularization of the constitutive equations defining surface-based cohesive behavior; toggle on **Specify damage stabilization**, click the **Stabilization** tab, and specify a viscosity coefficient.

8. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying fracture criterion properties for crack propagation

You can specify the fracture criterion that is used to model crack propagation using the virtual crack closure technique (VCCT) in an Abaqus/Standard model. The fracture criterion specifies the critical energy release rates. For more information, see *Crack Propagation Analysis*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Fracture Criterion**.
5. Select the **Type** of criterion for crack propagation along initially partially bonded surfaces—the virtual crack closure technique (**VCCT**) criterion or the enhanced virtual crack closure technique (**Enhanced VCCT**) criterion. The virtual crack closure techniques are available only in an Abaqus/Standard analysis.
6. If you are using the crack propagation criterion in an enriched region, choose the direction of crack growth relative to the local 1-direction when the fracture criterion is satisfied. The crack can extend at a direction normal to the direction of the maximum tangential stress (default), normal to the element local 1-direction, or parallel to the element local 1-direction.
7. Select the mixed mode behavior:
   - Select **BK** to specify the fracture energy as a function of the mode mix by means of the Benzeggagh-Kenane mixed mode fracture criterion.
   - Select **Power** to specify the fracture energy as a function of the mode mix by means of a power law mixed mode fracture criterion.
   - Select **Reeder** to specify the fracture energy as a function of the mode mix by means of the Reeder mixed mode fracture criterion.
8. If desired, specify the tolerance within which the crack propagation criterion must be satisfied. The default is 0.2.
9. If desired, specify the tolerance within which the unstable crack propagation criterion must be satisfied to allow multiple nodes at and ahead of the crack tip to debond without cutting back the increment size in one increment when the VCCT criterion is satisfied for an unstable crack problem. The default value is infinity.
10. If desired, specify the viscosity coefficient used in the viscous regularization. The default value is 0.0.
11. If you selected **VCCT** for the type of fracture criterion, define the energy release rates (for both crack onset and crack propagation) at each mode: G_{IC}, G_{IIC}, and G_{IIIC}.
12. If you selected **Enhanced VCCT** for the type of fracture criterion, do the following:
    - Define the energy release rates for crack onset at each mode: G_{IC}, G_{IIC}, and G_{IIIC}.
    - Define the energy release rates for crack propagation at each mode: G^P_{IC}, G^P_{IIC}, and G^P_{IIIC}.
13. If you selected either **Reeder** or **BK** as the fracture criterion, define the exponent in the Reeder law or the Benzeggagh-Kenane model, η.

14. If you selected **Power** as the fracture criterion, define the three exponents in the power law model, a_m, a_n, and a_o.
15. Toggle on **Use temperature-dependent data** if the fracture criterion is dependent on temperature.
16. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the fracture criterion depends.
17. Enter the required data in the data table provided.
18. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying geometric properties for mechanical contact property options

You can define additional geometric properties that will be accounted for in surface contact interactions.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Geometric Properties**.
5. If you are performing an Abaqus/Standard analysis, you can specify an out-of-plane surface thickness for two-dimensional models or a cross-sectional area for every node on a node-based surface. Enter this value in the **Out-of-plane surface thickness or cross-sectional area (Standard)** field.
6. If you are performing an Abaqus/Explicit analysis, you can specify the thickness of an interfacial layer between the two interacting surfaces. Toggle on **Thickness of interfacial layer (Explicit)**, and enter the thickness.
7. If desired, toggle on **Thickness to determine the contacting surfaces to be tracked**, and enter the thickness.
8. If desired, select the **Surface interaction model type**.
   - Select **Default** if the surface interaction model is not defined in a user subroutine.
   - Select **User** to define the surface interaction model in user subroutine *UINTER* in an Abaqus/Standard analysis or in user subroutine *VUINTER* in an Abaqus/Explicit analysis.
   - Select **User Interaction** to define the surface interaction model in user subroutine *VUINTERACTION* in an Abaqus/Explicit analysis.
9. If you selected **User** or **User Interaction** as the model type:
   - Specify the number of **State-dependent variables** required in the user subroutine. The default is 0.
   - Specify the **Number of property values** needed as data to define the surface interaction model in the user subroutine. The default is 0.
10. If you are performing an Abaqus/Standard analysis and if you selected **User** or **User Interaction** as the model type, toggle on **Use unsymmetric equation solution procedures (Standard)** to use unsymmetric equation solution procedures.
11. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Defining thermal contact property options

You can define thermal contact property options to specify thermal conductance, heat generation, and thermal radiation due to friction.

In this section:

- *Specifying thermal conductance for thermal contact property options*
- *Specifying heat generation for thermal contact property options*
- *Specifying radiation for thermal contact property options*

Specifying thermal conductance for thermal contact property options

You can specify thermal conductance to define conductive heat transfer between closely adjacent (or contacting) surfaces. For more information, see *Contact Conductance between Surfaces*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Thermal->Thermal Conductance**.

   The **Edit Contact Property** dialog box appears.
5. In the editor that appears, click the arrow to the right of the **Definition** field, and select an option for defining thermal conductance:
   - Select **Tabular** to enter data relating thermal conductance to the clearance or pressure between the contact surfaces.
   - Select **User-defined** to define thermal conductance in user subroutine *GAPCON*. If you select this option, skip to Step 9.
6. Indicate whether you want to define thermal conductance as a function of the clearance between the surfaces, the contact pressure between the surfaces, or both.
7. If you want to define thermal conductance as a function of clearance, display the **Clearance Dependency** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** if the data are dependent on temperature.
   b. Toggle on **Use mass flow rate-dependent data (Standard only)** if the data are dependent on the average mass flow rate per unit area, $\overline{|\dot{m}|}$.
   c. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the data depend.
   d. In the data table, define thermal conductance as a function of gap clearance.

      The tabular data must start at zero clearance (closed gap) and define thermal conductance as clearance increases. You must provide at least two pairs of points. The value of thermal conductance drops to zero immediately after the last data point, so there is no conductance when the clearance is greater than the value corresponding to the last data point. If conductance is not also defined as a function of contact pressure, it will remain constant at the zero clearance value for all pressures.
8. If you want to define thermal conductance as a function of contact pressure, display the **Pressure Dependency** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** if the data are dependent on temperature.
   b. Toggle on **Use mass flow rate-dependent data (Standard only)** if the data are dependent on the average mass flow rate per unit area, $\overline{|\dot{m}|}$.
   c. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the data depend.
   d. In the data table, define thermal conductance as a function of contact pressure at the interface.

      The tabular data must start at zero contact pressure (or, in the case of contact that can support a tensile force, the data point with the most negative pressure) and define thermal conductance as pressure increases. The value of thermal conductance remains constant for contact pressures outside

of the interval defined by the data points. If conductance is not also defined as a function of clearance, it is zero for all positive values of clearance and discontinuous at zero clearance

9. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying heat generation for thermal contact property options

You can specify heat generation due to the dissipation of energy created by the mechanical or electrical interaction of contacting surfaces. For more information, see *Modeling Heat Generated by Nonthermal Surface Interactions*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Thermal->Heat Generation**.
5. In the editor that appears, specify the **Fraction of dissipated energy caused by friction or electric currents that is converted to heat**:
   - Choose **Use default (1.0)** to convert all of the dissipated energy to heat.
   - Choose **Specify** to enter the fraction of your choice.
6. Specify the **Fraction of converted heat distributed to secondary surface**:
   - Choose **Use default (0.5)** to distribute the heat equally between the main and secondary surfaces.
   - Choose **Specify** to enter the fraction of the heat to be distributed to the secondary surface. The remaining fraction will be distributed to the main surface.
7. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying radiation for thermal contact property options

You can specify radiative heat transfer between closely adjacent surfaces. For more information, see *Modeling Radiation between Surfaces When the Gap Is Small*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Thermal->Radiation**.
5. In the editor that appears, enter values for the emissivity, ϵ, of the main and secondary surfaces.
6. In the table provided, define the view factor as a function of clearance.

   The view factor should have a value between 0.0 and 1.0. At least two pairs of points are required. The tabular data must start at zero clearance (closed gap) and define the view factor as the clearance increases. The value of the view factor drops to zero immediately after the last data point, so there is no radiative heat transfer when the clearance is greater than the value corresponding to the last data point.
7. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Specifying gap conductance for electrical contact property options

You can specify gap conductance between closely adjacent or contacting surfaces. The conductance is proportional to the difference in electric potentials across the interface. The conduction is a function of the clearance (separation) between the surfaces and can be a function of the contact pressure. For more information, see *Electrical Contact Properties*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Electrical->Electrical Conductance**.

   The **Edit Contact Property** dialog box appears.
5. In the editor that appears, click the arrow to the right of the **Definition** field, and select an option for defining electrical conductance:
   - Select **Tabular** to enter data relating electrical conductance to the separation between the contact surfaces.
   - Select **User-defined** to define the conductance in user subroutine *GAPELECTR*. If you select this option, skip to Step 9.
6. Indicate whether you want to define electrical conductance as a function of the clearance between the surfaces, the contact pressure between the surfaces, or both.
7. If you want to define electrical conductance as a function of clearance, display the **Clearance Dependency** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** if the data are dependent on temperature.
   b. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the data depend.
   c. In the data table, define electrical conductance as a function of gap clearance.

      The tabular data must start at zero clearance (closed gap) and define electrical conductance as clearance increases. You must provide at least two pairs of points. The value of electrical conductance drops to zero immediately after the last data point, so there is no conductance when the clearance is greater than the value corresponding to the last data point. If conductance is not also defined as a function of contact pressure, it will remain constant at the zero clearance value for all pressures.
8. If you want to define electrical conductance as a function of contact pressure, display the **Pressure Dependency** tabbed page, and do the following:
   a. Toggle on **Use temperature-dependent data** if the data are dependent on temperature.
   b. Click the arrows to the right of the **Number of field variables** field to specify the number of field variables on which the data depend.
   c. In the data table, define electrical conductance as a function of contact pressure at the interface.

      The tabular data must start at zero contact pressure (or, in the case of contact that can support a tensile force, the data point with the most negative pressure) and define electrical conductance as pressure increases. The value of electrical conductance remains constant for contact pressures outside of the interval defined by the data points. If conductance is not also defined as a function of clearance, it is zero for all positive values of clearance and discontinuous at zero clearance

9. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box. Alternatively, you can select another contact property option to define from the menus in the **Edit Contact Property** dialog box.

Defining a film condition interaction property

You can define a film coefficient as a function of temperature and field variables. A film condition interaction property can be referred to only by a surface film condition interaction or a concentrated film condition interaction. For more information, see *Defining a surface film condition interaction*, and *Defining a concentrated film condition interaction*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Film condition** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. Toggle on **Use temperature-dependent data** to define a film coefficient that varies with temperature. A column labeled **Temp** appears in the data table.
5. To define a film coefficient that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
6. In the data table, enter the film coefficient, h (units of $JT^{-1}L^{-2}\theta^{-1}$), as a function of temperature and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
7. Click **OK** to create the film condition interaction property and to exit the editor.

Additional information

• *Thermal Loads*

Defining a cavity radiation interaction property

You can define the emissivity of cavity surfaces as a function of temperature and field variables. A cavity radiation interaction property can be referred to only by a cavity radiation interaction. For more information, see *Defining a cavity radiation interaction*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Cavity radiation** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. Toggle on **Use temperature-dependent data** to define a cavity radiation property that varies with temperature. A column labeled **Temp** appears in the data table.
5. To define a cavity radiation property that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
6. In the data table, enter the emissivity, ϵ, as a function of temperature and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
7. Click **OK** to create the cavity radiation interaction property and to exit the editor.

Additional information

• *Cavity Radiation in Abaqus/Standard*

Defining a fluid cavity interaction property

The fluid cavity interaction editor contains hydraulic and pneumatic fluid type definitions.

Fluid cavity interaction properties can be referred to only by fluid cavity interactions. For more information, see *Defining a fluid cavity interaction*.

In this section:

- *Defining hydraulic fluid cavity property options*
- *Defining pneumatic fluid cavity property options*

Defining hydraulic fluid cavity property options

You should use a hydraulic fluid definition to model nearly incompressible or fully incompressible fluid behavior within a cavity. Hydraulic fluids must include a density, and they may include a bulk modulus and expansion data. For more information, see *Hydraulic Fluids*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Fluid cavity** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.

   Abaqus/CAE opens the **Edit interaction property** dialog box.
4. Choose the **Hydraulic** fluid definition.
5. Enter the **Fluid density**.
6. In the **Fluid Bulk Modulus** tab, toggle on **Specify fluid bulk modulus** to enter a bulk modulus and to allow the entry of temperature and field variable data.

   The fluid bulk modulus is required for Abaqus/Explicit and optional for Abaqus/Standard.
7. Toggle on **Use temperature-dependent data** to define a fluid bulk modulus that varies with temperature. A column labeled **Temp** appears in the data table.
8. To define a fluid bulk modulus that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
9. In the data table, enter the bulk modulus as a function of temperature and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
10. In the **Fluid Expansion** tab, toggle on **Specify fluid thermal expansion coefficients** to specify expansion data.
11. Toggle on **Use temperature-dependent data** to define fluid expansion that varies with temperature. A column labeled **Temp** appears in the data table.
12. To define fluid expansion that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
13. If the fluid expansion is dependent on temperature or field variables, enter the **Reference temperature** for use in calculation of the expansion coefficient.
14. In the data table, enter the fluid expansion coefficient as a function of temperature and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
15. Click **OK** to create the hydraulic fluid cavity interaction property and to exit the editor.

Defining pneumatic fluid cavity property options

Pneumatic fluids must include an ideal gas molecular weight; and, for an Abaqus/Explicit analysis, they may include molar heat capacity data. For more information, see *Pneumatic Fluids*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Fluid cavity** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.

   Abaqus/CAE opens the **Edit interaction property** dialog box.
4. Choose the **Pneumatic** fluid definition.
5. Enter the **Ideal gas molecular weight**.
6. For an Abaqus/Explicit analysis, toggle on **Specify molar heat capacity**, if desired, to include heat transfer data.
7. If you are specifying molar heat capacity, select either **Polynomial** or **Tabular** for the data type.
8. If you chose **Polynomial** in the previous step, enter coefficients for the five terms in the polynomial equation—enter zero for any terms that are not needed. If you chose **Tabular** in the previous step, do the following to complete the data table:
   a. Toggle on **Use temperature-dependent data** to define fluid expansion that varies with temperature. A column labeled **Temp** appears in the data table.
   b. To define fluid expansion that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
   c. In the data table, enter the molar heat capacity as a function of temperature and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
9. Click **OK** to create the pneumatic fluid cavity interaction property and to exit the editor.

Defining a fluid exchange interaction property

You use a fluid exchange interaction property to define the method of fluid transfer between a cavity and the environment or between two cavities. A fluid exchange interaction property can be referred to only by a fluid exchange interaction. For more information, see *Defining a fluid exchange interaction*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Fluid exchange** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
   Abaqus/CAE opens the **Edit interaction property** dialog box.
4. Choose one of the following fluid exchange definitions:

Bulk viscosity

The fluid exchange rate is based on the viscous resistance coefficient and the hydrodynamic resistance coefficient. Both coefficients may include dependence on the average absolute pressure, average temperature, and the average value of any user-defined field variables.

Mass flux

The fluid exchange rate is based on the mass flow rate per unit area for the effective area defined in the fluid exchange interaction.

Mass rate leakage

The fluid exchange rate is based on a mass flow rate driven by the absolute value of the pressure differential between the primary cavity and the environment or the second cavity. The mass flow rate and the absolute value of the pressure difference both start at zero and must be positive. Mass rate leakage may include dependence on the average absolute pressure, average temperature, and the average value of any user-defined field variables.

Volume flux

The fluid exchange rate is based on the volumetric flow rate per unit area for the effective area defined in the fluid exchange interaction.

Volume rate leakage

The fluid exchange rate is based on a volume flow rate driven by the absolute value of the pressure differential between the primary cavity and the environment or the second cavity. The volume flow rate and the absolute value of the pressure difference both start at zero and must be positive. Volume rate leakage may include dependence on the average absolute pressure, average temperature, and the average value of any user-defined field variables.

5. If you selected the **Bulk viscosity**, **Mass rate leakage**, or **Volume rate leakage** fluid exchange definition, do the following to complete the fluid exchange property definition:
   a. Toggle on the desired options to add data columns to the table.

You can add pressure- or temperature-dependent data, and you can add field variables.

b. In the data table, enter the appropriate data as a function of temperature, pressure, and field variables. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.

6. If you selected the **Mass flux** or **Volume flux** fluid exchange definition, enter the mass flow rate per unit area or volumetric flow rate per unit area, respectively.

7. Click **OK** to create the fluid exchange interaction property and to exit the editor.

Additional information

• *Fluid Exchange Definition*

Defining an acoustic impedance interaction property

You use an acoustic impedance interaction property to define the proportionality factors between the pressure and the normal components of surface displacement and velocity in an acoustic analysis. An acoustic impedance interaction property can be referred to only by an acoustic impedance interaction. For more information, see *Defining acoustic impedance*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Acoustic impedance** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the **Data type** field, choose the type of tabular data that you will use to define the acoustic impedance.
   - Select **Impedance** to specify an impedance using real and imaginary parts of the impedance.
   - Select **Admittance** to specify an impedance using admittance values.
5. Toggle on **Use frequency-dependent data** to define an acoustic impedance that varies with frequency. A column labeled **Frequency** appears in the data table.
6. If you are using impedance data, enter the following data in the table:
   - **Re (Z)**, the real part of the surface impedance. (Units of F $L^{-3}T$.)
   - **Im (Z)**, the imaginary part of the surface impedance. (Units of F $L^{-3}T$.)
7. If you are using admittance data, enter the following data in the table:
   - $1/c_1$, the proportionality factor between pressure and velocity of the surface in the normal direction. This quantity is the real part of the complex admittance. (Units of $F^{-1}L^3T^{-1}$.)
   - $1/k_1$, the proportionality factor between pressure and displacement of the surface in the normal direction. This quantity is the imaginary part of the complex admittance, divided by the angular frequency. (Units of $F^{-1}L^3$.)
8. Click **OK** to create the acoustic impedance interaction property and to exit the editor.

Additional information

• *Acoustic and Shock Loads*

Defining an incident wave interaction property

You can define the speed of the incident wave and other characteristics of the wave loading. For spherical incident wave loading, you can optionally specify the loading effects due to an UNDEX bubble or spatial decay of the incident wave field. An incident wave interaction property can be referred to only by an incident wave interaction. Incident wave interaction properties that specify UNDEX data can be referred to only by incident wave interactions with an UNDEX definition. For more information, see *Defining incident waves*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Incident wave** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. In the **Speed of sound in fluid** field, enter the sound speed in fluid, c_f.
5. In the **Fluid density** field, enter fluid mass density, ρ_f.
6. From the **Definition** field, choose the type of wave to define the incident wave property.
   - Select **Planar** to specify a planar incident wave.
   - Select **Spherical** to specify a spherical incident wave.
   - Select **Diffuse** to specify a field of planar waves incident from multiple angles.
   - Select **Air blast** to specify air blast loading on structures using the CONWEP model.
   - Select **Surface blast** to specify surface blast loading on structures using the CONWEP model.
7. If you are specifying a spherical incident wave, select the propagation model. For more information, see *Describing Incident Wave Loading*.
   - Select **Acoustic** to specify an incident wave in which the amplitude is inversely proportional to the distance from the source.
   - Select **UNDEX charge** to specify UNDEX bubble data.
   - Select **Generalized decay** to specify spatial decay of the incident wave field.
8. If you selected the **UNDEX charge** propagation model, click the **Physical**, **Material**, and **Bubble Model** tabs and specify the following values:

Physical Data

- Ratio of specific heats for gas, γ.
- Acceleration due to gravity, g.
- Atmospheric pressure, p_{atm}.
- Flow drag coefficient, C_D.
- Flow drag exponent, E_D.
- Toggle on **Neglect wave effects in fluid and gas** to neglect the wave effects.

Material Data

- Charge material constant, *K*.
- Charge material constant, *k*.
- Charge material constant, *A*.
- Charge material constant, *B*.
- Adiabatic charge constant, K_c.
- Density of charge material, ρ_c.
- Mass of charge material, m_c.

Bubble Model Step Data

- Time duration, T_{final}.
- Maximum number of time steps for the bubble simulation, N_{steps}. The bubble amplitude simulation ceases when the number of steps reaches N_{steps} or the time duration, T_{final}, is reached.
- Relative step size control parameter, Ω_{rel}.
- Absolute step size control parameter, X_{abs}.
- Step size control exponent, β. The step size, Δt, is decreased or increased according to the error estimate: $(\Omega_{\text{rel}}\,|x| + X_{\text{abs}})^{\beta} \le \Delta t \left|\frac{dx}{dt}\right|^{\beta}$.

9. If you selected the **Generalized decay** propagation model, enter values for the dimensionless constants **A**, **B**, and **C**, where $A > -1$, $B > -1$, and $C \ge 0$.
10. If you are defining a diffuse incident wave, specify the seed number in the **Seed number** field for the diffuse source calculation.
11. If you selected the **Air blast** or **Surface blast** definition, specify the following settings in the **CONWEP Charge** portion of the dialog box:
    - In the **Equivalent mass of TNT** field, specify the equivalent mass of TNT in any preferred mass unit.
    - In the **Conversion for mass to kilograms** field, specify a multiplication factor to convert the preferred mass unit to kilograms.
    - In the **Conversion for length to meters** field, specify a multiplication factor to convert the analysis length unit to meters.
    - In the **Conversion for time to seconds** field, specify a multiplication factor to convert the analysis time unit to seconds.
    - In the **Conversion for pressure to pascals** field, specify a multiplication factor to convert the analysis pressure unit to pascals.

12. Click **OK** to create the incident wave interaction property and to exit the editor.

Additional information

• *Acoustic and Shock Loads*

Defining an actuator/sensor interaction property

An actuator/sensor interaction property provides the PROPS, JPROPS, NPROPS, and NJPROPS variables that are passed into a *UEL* user subroutine used with an actuator/sensor interaction. For more information, see *Defining an actuator/sensor interaction*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Actuator/sensor** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. Click the **Real Properties** tab, and enter the real property values to provide the PROPS and NPROPS variables that are passed into a *UEL* user subroutine.
5. Click the **Integer Properties** tab, and enter the integer property values to provide the JPROPS and NJPROPS variables that are passed into a *UEL* user subroutine.
6. Click **OK** to create the actuator/sensor interaction property and to exit the editor.

Additional information

- *User-Defined Elements*
- *UEL*

Defining a fluid inflator interaction property

You use a fluid inflator interaction property to model the deployment of an airbag. A fluid inflator interaction property can be referred to only by a fluid inflator interaction. The fluid inflator property defines the mass flow rate and temperature as a function of inflation time either directly or by entering tank test data. It also defines the mixture of gases entering the fluid cavity. For more information, see *Defining a fluid inflator interaction*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property. For more information about naming objects, see *Using basic dialog box components*.
   - Select the **Fluid inflator** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
   Abaqus/CAE opens the **Edit interaction property** dialog box.
4. In the **Fluid Inflator Property** tab, select one of the following fluid inflator definitions:

Dual pressure

The inflator mass flow rate and temperature of the gas entering the fluid cavity are calculated using the dual pressure method. You enter a table of tank pressure and inflator pressure versus inflation time; and specify the volume of the tank, the effective area, and the discharge coefficient.

Temperature and mass

The inflator mass flow rate and temperature are given directly as functions of inflation time.

Tank test

The inflator mass flow rate and the temperature of the gas entering the fluid cavity are determined by the results of a tank test. To calculate the mass flow rate using the results of a tank test, you enter a table of tank pressure and inflator temperature versus inflation time, and specify the volume of the tank.

Inflator pressure and mass flow rate

The inflator mass flow rate and inflator pressure are used to obtain the gas temperature. You can enter a table of the mass flow rate and inflator pressure versus inflation time and specify the effective area and discharge coefficient.

5. If you selected the **Dual pressure** or **Inflator pressure and mass flow rate** fluid inflator definition, enter the **Effective area**. The effective area is the total inflator orifice area.
6. If you selected the **Dual pressure** or **Tank test** fluid inflator definition, enter the **Tank volume**.
7. If you selected the **Dual pressure** or **Inflator pressure and mass flow rate** fluid inflator definition, enter the **Discharge coefficient**.
8. In the data table, enter the appropriate data. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
9. In the **Fluid Inflator Mixture** tab, enter the number of fluids to define for this inflator.

10. In the **Type** field, select whether you want to specify the mixture in terms of the mass fraction or molar fraction.
11. In the data table, for each fluid, select the fluid cavity interaction property that defines the fluid behavior. Then enter the inflation time values and enter the corresponding mass fraction or molar fraction for that fluid.
12. Click **OK** to create the fluid inflator interaction property and to exit the editor.

Additional information

• *Inflator Definition*

Defining a wear interaction property

You can specify the surface wear (erosion) due to various microscale mechanisms, such as abrasion and fretting.

For more information, see *Contact Wear*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   a. Name the interaction property.

      For more information about naming objects, see *Using basic dialog box components*.

   b. Select the **Wear** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.

   Abaqus/CAE opens the **Edit interaction property** dialog box.
4. Toggle on **Friction coefficient** to specify that the Archard's wear equation depends explicitly on the friction coefficient.
5. Toggle on **Unitless wear coefficient** to specify that the wear coefficient is dimensionless. Otherwise, the wear coefficient has dimensions of inverse of stress.
6. If **Unitless wear coefficient** is on, specify the **Reference stress** value equal to the magnitude of hardness of the material forming the wear surface.
7. To define a wear property that depends on surface temperature, toggle on **Use surface temperature dependent data**.

   Surface Temperature variable columns appear in the data table.
8. To define a wear property that depends on wear distance, toggle on **Use surface wear distance dependent data**.

   Surface Wear Distance variable columns appear in the data table.
9. To define a wear property that depends on contact pressure, toggle on **Use contact pressure dependent data**.

   Contact Pressure variable columns appear in the data table.
10. To define a wear property that depends on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables.

    Field variable columns appear in the data table.
11. In the data table, enter the appropriate data.

    You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data.

    For detailed information on each option, see *Entering tabular data*.
12. Click **OK** to create the wear interaction property and to exit the editor.

Additional information

• *Contact Wear*

Using the constraint editors

This section explains how to enter data in the constraint editor to define specific types of constraints.

In this section:

- *Defining tie constraints*
- *Defining rigid body constraints*
- *Defining display body constraints*
- *Defining coupling constraints*
- *Defining adjust points constraints*
- *Defining MPC constraints*
- *Defining shell-to-solid coupling constraints*
- *Defining embedded region constraints*
- *Defining equation constraints*

Defining tie constraints

A tie constraint ties two separate surfaces together so that there is no relative motion between them. This type of constraint allows you to fuse together two regions even though the meshes created on the surfaces of the regions may be dissimilar. You can define a tie constraint between edges of a wire or between faces of a solid or shell. For more information, see *Understanding constraints*, and *Mesh Tie Constraints*.

If you are creating multiple tie constraints, you may want to use the automatic contact detection tool. This tool automates the process of selecting surfaces and allows you to create multiple constraints simultaneously. For more information, see *Using contact and constraint detection*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a tie constraint using the  tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Tie**, then click **Continue**.
3. Select the main surface.
   a. In the prompt area, select one of the following:
      - Select **Surface** if you want to select a named surface.
      - Select **Node Region** if you want to select a region from which to create a node-based surface.

   b. Use one of the following methods to select the main surface:
      - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface name from the **Region Selection** dialog box that appears, and click **Continue**.

        Note:

        The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

      - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

        If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

        - Click **Geometry** if you want to select the surface or vertex from a geometry region.
        - Click **Mesh** if you want to select the surface or node from a native or orphan mesh selection.

        You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

      The main surface that you select becomes highlighted in red in the viewport.
4. Select the secondary surface.

a. In the prompt area, select one of the following:

- Select **Surface** if you want to select a surface.
- Select **Node Region** if you want to select a region from which to create a node-based surface.

b. Use one of the same methods described in the previous step to select the secondary surface or region.

The secondary surface or region that you select becomes highlighted in magenta in the viewport.

The constraint editor appears.

5. The **Switch Surfaces** option allows you to interchange your main and secondary surface selections without having to start over. The **Switch Surfaces**  icon is available only when the main and secondary regions are the same type—both surfaces or both node-based regions.
6. From the editor, select the **Discretization method**.
   - Select **Analysis default** to use the default discretization method: surface-to-surface for Abaqus/Standard and node-to-surface for Abaqus/Explicit.
   - Select **Node to surface** to generate the tie coefficients according to the interpolation functions at the point where the secondary node projects onto the main surface.
   - Select **Surface to surface** to generate the tie coefficients such that stress accuracy is optimized for the specified surface pairing.
7. Toggle on **Exclude shell element thickness** if you want to ignore shell thickness effects in calculations involving position tolerances and adjustments for initial gaps.
8. Choose one of the following **Position Tolerance** methods:
   - **Use computed default**. Abaqus determines the nodes to be tied using the default position tolerance. For more information, see *Mesh Tie Constraints*.
   - **Specify distance**. You can specify an absolute distance from the main surface within which all nodes on the secondary surface to be tied must lie.
9. Toggle on **Adjust secondary surface initial position** if you want Abaqus to move all the nodes of the secondary surface onto the main surface in the initial configuration.
10. Toggle on **Tie rotational DOFs if applicable** if you want Abaqus to constrain the rotational degrees of freedom that exist on both main and secondary surfaces.
11. If desired, you can specify a value for the constraint ratio. You must toggle off **Tie rotational DOFs if applicable** to make the constraint ratio option available.
12. Click **OK** to save your constraint definition and to close the editor.

Additional information

• *Understanding constraints*

Defining rigid body constraints

You can create a rigid body constraint by specifying the regions that you want to include in the rigid body and by specifying a rigid body reference point. For detailed information about rigid bodies, see *Rigid Body Definition*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a rigid body constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Rigid Body**, and then click **Continue**.

   The constraint editor appears.

3. From the editor, select all of the regions that you want to include in the rigid body.
   a. Select one of the following from the **Region type** list:

   - Select **Body** if you want to include the elements of a geometric region or orphan elements in the rigid body.
   - Select **Pin** to include nodes that will have only translational degrees of freedom associated with the rigid body.
   - Select **Tie** to include nodes that will have both translational and rotational degrees of freedom associated with the rigid body.
   - Select **Analytical Surface** to include an analytical surface in the rigid body.

   b. After you select a region type, click on the right side of the editor.
   c. Select a region of the assembly to associate with the **Region type** category selected in the previous step. Use one of the following methods to select the region:

   - Use an existing set or surface to define the region. On the right side of the prompt area, click **Sets** or **Surfaces**. Select an existing set or surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select the region in the viewport. (For more information, see *Selecting objects within the current viewport*.)

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select a geometry region.
     - Click **Mesh** if you want to select nodes from a native or orphan mesh selection.

You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

d. To remove a region type from the rigid body, select that region type, and then click  on the right side of the editor.

4. Repeat Step 3 as often as necessary to select all of the regions that you want to include in the rigid body.
5. Select the rigid body reference point:

   a. In the bottom half of the editor, click .

   b. Use one of the techniques described above to select a vertex or node to serve as the rigid body reference point. For more information, see *The Reference Point toolset*."
6. Toggle on **Adjust point to center of mass at start of analysis** if you want Abaqus to reposition the rigid body reference point at the calculated center of mass of the rigid body.
7. Toggle on **Constrain selected regions to be isothermal** to specify an isothermal rigid body for a fully coupled thermal-stress analysis.
8. Click **OK** to save your constraint definition and to close the editor.

Additional information

- *Understanding constraints*
- *Rigid Body Definition*

Defining display body constraints

You can create a display body constraint by selecting a part instance that will be displayed but will not be included in the analysis. You can constrain the display body to be fixed in space, or you can constrain it to follow selected nodes in the assembly. You can apply a display body constraint to any part instance in the model. For more information, see *Display Body Definition*. In addition, you can control the appearance of display bodies in the Visualization module; for more information, see *Customizing the appearance of display bodies*. For an example of a model that includes a display body constraint combined with connectors, see *Display bodies*."

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a display body constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Display body**; then click **Continue**.
3. Select the part instance that will be a display body.

   The constraint editor appears.
4. By default, the part instance is fixed; alternatively, you can constrain it to follow selected points in the assembly. From the **Motion control** field of the constraint editor, choose one of the following:

No motion

Choose **No motion** to fix the selected part instance in space during the analysis. This is the default option.

Follow single point

Choose **Follow single point**, click , and select the point to which the part instance will be constrained. You must select the point from a different part instance, and that part instance must not be a display body. During the analysis the display body will follow the translations and rotations of the selected point.

Follow three points

Choose **Follow three points**, click , and select the three points to which the part instance will be constrained. You must select the points from a different part instance, and that part instance must not be a display body. During the analysis the display body will follow the translations and rotations of the coordinate system defined by the selected points. The first point indicates the origin of the coordinate system, the second point indicates the *X*-direction, and the third point indicates the *X*–*Y* plane. The points should be non-colinear and should remain non-colinear during the analysis.

5. Click **OK** to save your constraint definition and to close the editor.

Additional information

- *Display bodies*
- *Understanding constraints*
- *The Reference Point toolset*
- *Rigid Body Definition*

Defining coupling constraints

You use a coupling constraint to constrain the motion of a surface to the motion of one or more points.
You can create a coupling constraint by specifying one or more control points, a constraint region, and an influence radius that defines the points in the constraint region to include in the constraint. You can specify a kinematic or distributing coupling constraint type. For detailed information about coupling constraints, see *Coupling Constraints*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a coupling constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Coupling**, and then click **Continue**.
3. Select one or more points to define the constraint control points using one of the following methods:

   - Select one or more points in the viewport.

     Tip: The **Select the Entity Closest to the Screen** tool in the **Selection** toolbar is toggled off by default. If you make an ambiguous selection, Abaqus/CAE highlights the point and displays a description of the point in the lower-left corner of the viewport. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

     (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the constraint control point from geometry or select a reference point.
     - Click **Mesh** if you want to select the constraint control point from a native or orphan mesh selection.

   - Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The points that you select become highlighted in red in the viewport.
4. In the prompt area, select one of the following to define the constraint region type:

   - Select **Surface** if you want to select a surface.
   - Select **Node Region** if you want to select a region from which to create a node-based surface.

5. Select the constraint region using one of the following methods:

   - Select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   The region that you select becomes highlighted magenta in the viewport, and the constraint editor appears.

6. From the editor, choose one of the following **Coupling type** categories:

   - Choose **Kinematic** to define a kinematic coupling constraint between the control points and the points in the constraint region, and toggle on the degrees of freedom that you want to constrain.
   - Choose **Distributing** to define a distributing coupling constraint between the control points and the points in the constraint region. Abaqus/CAE automatically constrains the translational degrees of freedom.

     1. Toggle on the rotational degrees of freedom that you want to constrain.
     2. Choose the **Translational Coupling type** option: **Translational Continuum distributing** or **Translational Structural distributing**.
     3. Choose the **Rotational Coupling type** option: **Rotational Continuum distributing** or **Rotational Structural distributing**.
     4. Click the arrow next to the **Weighting method** field, and select a weighting method from the list that appears. For more information, see *Coupling Constraints*.

7. From the editor, choose one of the following methods to define the **Influence radius**:

   - **To outermost point on the region**. Abaqus includes all points (nodes) on the specified region in the coupling definition.
   - **Specify**. You can specify the radius of a sphere, centered about the constraint control points, to limit the points in the coupling definition. For more information on selecting the coupling points, see *Coupling Constraints*.

8. If desired, toggle on **Adjust control points to lie on surface**. Abaqus/CAE will move the control points to lie on the constraint surface.

9. If you want to change the coordinate system (**CSYS**) for the coupling constraint, click  and use one of the following methods:

   - Select a predefined datum coordinate system by name.

     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.

2. Select a name from the list, and click **OK**.

- Select a predefined coordinate system in the viewport.

  Tip: The  tool in the **Selection** toolbar is toggled off by default. For coordinate systems with coincident origins, when you cycle through all the possible selections, Abaqus/CAE highlights the coordinate system and displays the description of the coordinate system in the viewport.

- Click **Use Global CSYS** from the prompt area to use the global coordinate system.

10. Click **OK** to save your constraint definition and to close the editor.

Additional information

- *Understanding constraints*
- *Coupling Constraints*

Defining adjust points constraints

You use an adjust points constraint to move a point or points onto a specified surface. For more information about adjusting points, see *Adjusting Nodal Coordinates*. This adjustment may be useful in assembled fastener template models and other applications; see *About assembled fasteners*, and *Creating assembled fasteners*.

An adjust points constraint should not be used in assembled fastener template models when the main model attachment points are located on bolt hole center points. Any point along the bolt hole centerline will be moved incorrectly to a random location along the perimeter of the hole instead of being projected along the surface normal to the center of the hole.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create an adjust points constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Adjust points**, and then click **Continue**.
3. Select the point or set of points to be moved, using one of the following methods:

   - Select one or more points in the viewport.

     Tip: The **Select the Entity Closest to the Screen** tool in the **Selection** toolbar is toggled off by default. If you make an ambiguous selection, Abaqus/CAE highlights the point and displays a description of the point in the lower left corner of the viewport. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

     Click mouse button 2 to indicate that you have finished selecting. For more information, see *Selecting objects within the current viewport*.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the points from a geometry region or select one or more reference points.
     - Click **Mesh** if you want to select the points from a native or orphan mesh selection.

   - Use an existing set to define the point or points. On the right side of the prompt area, click **Sets**. Select an existing set of points from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The points that you select become highlighted in red in the viewport.

4. In the prompt area, select the type of region onto which the points will be moved.
   - Select **Surface** if you want to select a surface.
   - Select **Node Region** if you want to select a region from which to create a node-based surface.
5. Select the surface onto which the points will be moved using one of the following methods:
   - Select a surface in the viewport. Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:
     - Click **Geometry** if you want to select the surface from a geometry region or select one or more reference points.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.
   - Use an existing surface or set to define the region. On the right side of the prompt area, click **Surfaces** or **Sets**. Select an existing surface or set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** (or **Sets**) on the right side of the prompt area.

   The region that you select is highlighted in magenta in the viewport, and the constraint editor appears.
6. Click **OK** to save your constraint definition and to close the editor.

Additional information

• *Understanding constraints*

• *Adjusting Nodal Coordinates*

Defining MPC constraints

You use an MPC constraint to constrain the motion of secondary nodes of a region to the motion of a point. You can create an MPC constraint by specifying a control point and a region composed of nodes, edges, and surfaces. For detailed information about multi-point constraints, see *General Multi-Point Constraints*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a multi-point constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **MPC Constraint**, then click **Continue**.
3. Select a point to define the constraint control point using one of the following methods:

   - Use the mouse to select a point in the viewport.

     Tip: The **Select the Entity Closest to the Screen** tool in the **Selection** toolbar is toggled off by default. If you make an ambiguous selection, Abaqus/CAE highlights the point and displays a description of the point in the lower left corner of the viewport. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

     (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the control point from a geometry region.
     - Click **Mesh** if you want to select the control point from a native or orphan mesh selection.

   - Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The point that you select becomes highlighted in red in the viewport.
4. Select the region for the secondary nodes. Use one of the following methods to select the region:

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) The region that you select can span multiple part instances. Click mouse button 2 to indicate that you have finished selecting.

If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

- Click **Geometry** if you want to select a geometry region.
- Click **Mesh** if you want to select the region from a native or orphan mesh selection.

You can use the angle method to select a group of nodes from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

- Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears and click **Continue**.



Note:

The default selection method is based on the selection method that you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

The region that you select becomes highlighted in magenta in the viewport.

The constraint editor appears.

5. From the editor, select the **MPC type**.
   - Select **Beam** to define a rigid beam connection to constrain the displacement and rotation of each secondary node to the displacement and rotation of the control point.
   - Select **Tie** to make all active degrees of freedom equal at each secondary node and the control point.
   - Select **Link** to define a pinned rigid link between each secondary node and the control point.
   - Select **Pin** to define a pinned joint between each secondary node and the control point.
   - Select **Elbow** to constrain nodes of ELBOW31 or ELBOW32 elements together (see *Pipes and Pipebends with Deforming Cross-Sections: Elbow Elements*).
   - Select **User-defined** to define a multi-point constraint in user subroutine *MPC* (for Abaqus/Standard). See the following sections for more information:
     - *Specifying general job settings*
     - *MPC*

6. If you selected a user-defined MPC type, do the following:
   a. Choose the coding mode for user subroutine *MPC*.
      - Choose **DOF-by-DOF** if you want each call to the user subroutine to constrain one individual degree of freedom.
      - Choose **Node-by-Node** if you want each call to the user subroutine to impose a set of constraints all at once.

   b. In the **Constraint type** field, enter an integer value to be used in the user subroutine to distinguish between different constraint types. The default value is 0.

7. If you want to change the coordinate system (**CSYS**) for the coupling constraint, click  and use one of the following methods:
   - Select a predefined datum coordinate system by name.

1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
2. Select a name from the list, and click **OK**.

- Select a predefined coordinate system in the viewport.

  Tip: The  tool in the **Selection** toolbar is toggled off by default. For coordinate systems with coincident origins, when you cycle through all of the possible selections, Abaqus/CAE highlights the coordinate system and displays the description of the coordinate system in the viewport.

- Click **Use Global CSYS** from the prompt area to use the global coordinate system.

8. Click **OK** to save your constraint definition and to close the editor.

Defining shell-to-solid coupling constraints

You use a shell-to-solid coupling constraint to couple the motion of a shell edge to the motion of an adjacent solid face. You can create a shell-to-solid coupling constraint by specifying a shell edge surface and a solid face region. The shell edge surface and the solid face region being coupled must belong to different part instances unless the shell edge surface is part of a midsurface region within a solid model.

For detailed information about shell-to-solid coupling constraints, see *Shell-to-Solid Coupling*.

Abaqus/CAE creates shell-to-solid coupling constraints automatically only when shell sections and solid sections exist within the same part and they are nearly perpendicular to each other with a solid surface detected on both sides of the shell surface.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create a shell-to-solid coupling constraint using the  tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Shell-to-solid coupling**, and then click **Continue**.
3. Use one of the following methods to select the shell edge surface:

   - Use an existing surface to define the region. On the right side of the prompt area, click **Surfaces**. Select an existing surface from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - Use the mouse to select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select the surface from a geometry region.
     - Click **Mesh** if you want to select the surface from a native or orphan mesh selection.

     You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   The shell edge surface that you select becomes highlighted in red in the viewport.
4. Select the solid face region.
   a. In the prompt area, click the arrow next to the text field and select one of the following:

      - Select **Surface** if you want to select a surface.
      - Select **Node Region** if you want to select a region from which to create a node-based surface.

b. Use one of the same methods described in the previous step to select the solid face region.

The solid face region that you select becomes highlighted in magenta in the viewport

The constraint editor appears.

5. From the editor, choose one of the following **Position Tolerance** methods:

- **Use computed default**. Abaqus determines the nodes on the shell edge surface to be coupled with the solid face region using the default position tolerance. For more information, see *Shell-to-Solid Coupling*.
- **Specify distance**. You can specify an absolute distance from the solid face region within which all shell nodes to be included in the coupling must lie.

6. From the editor, choose one of the following **Influence Distance** methods:

- **Use analysis default**. Abaqus determines the nodes on the solid face region to be coupled with the shell edge surface using the default influence distance. For more information, see *Shell-to-Solid Coupling*.
- **Specify value**. You can specify a distance from the shell edge surface within which all solid nodes to be included in the coupling must lie.

7. Click **OK** to save your constraint definition and to close the editor.

Additional information

• *Understanding constraints*

• *Shell-to-Solid Coupling*

Defining embedded region constraints

You use an embedded region constraint to embed a region of the model within a "host" region of the model or within the whole model. You can create an embedded region constraint by specifying the embedded region, the host region, a weight factor roundoff tolerance, and an absolute exterior tolerance or fractional exterior tolerance. For more information, see *Embedded Elements*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create an embedded region constraint using the  tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Embedded region**, and then click **Continue**.
3. Select the embedded region using one of the following methods:

   - Select a region in the viewport. (For more information, see *Selecting objects within the current viewport*.) Click mouse button 2 to indicate that you have finished selecting.

     If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

     - Click **Geometry** if you want to select a geometry region.
     - Click **Mesh** if you want to select the region from a native or orphan mesh selection.

   - Use an existing set to define the region. On the right side of the prompt area, click **Sets**. Select an existing set from the **Region Selection** dialog box that appears, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The region that you select becomes highlighted in red in the viewport.
4. Select the host region.

   In the prompt area, select one of the following:

   - **Select Region** if you want to select a region in the viewport or use an existing set to define the region. Use one of the methods described in the previous step to select the host region.

     The region that you select becomes highlighted magenta in the viewport.
   - **Whole Model** if you want to embed the embedded region within the whole model.

   The constraint editor appears.
5. In the editor, specify values for the tolerance parameters. If you specify values for both exterior tolerance parameters, Abaqus will use the smaller of the two tolerances.

   - **Weight factor roundoff tolerance**. You can specify a small value below which the weighting factors will be zeroed out. The default value is 10^{-6}.

- **Absolute exterior tolerance**. You can specify the absolute value by which a node on the embedded region may lie outside the host region. If this parameter is omitted or has a value of 0.0, the **Fractional exterior tolerance** will apply.
- **Fractional exterior tolerance**. You can specify the fractional value by which a node on the embedded region may lie outside the host region. The fractional value is based on the average element size within the host region. The default value is 0.05.

6. Click **OK** to save your constraint definition and to close the editor.

Additional information

• *Understanding constraints*

• *Embedded Elements*

Defining equation constraints

You can create an equation constraint by entering data in the **Edit Constraint** dialog box.
The terms of an equation consist of a coefficient applied to a degree of freedom of every node in a set. For detailed information about equations, see *Linear Constraint Equations*.

1. From the main menu bar, select **Constraint->Create**.

   Tip: You can also create an equation constraint using the tool in the Interaction module toolbox.

2. In the **Create Constraint** dialog box that appears, do the following:
   a. Name the constraint. For more information about naming objects, see *Using basic dialog box components*.
   b. From the **Type** list, select **Equation**, and then click **Continue**.

   The constraint editor appears.

3. In the table in the editor, enter a row of data for each term in the equation. The equation must have at least two terms. Click for a description of the relationship between the data in the table and the desired equation.

   Each row should contain the following information:

   - The coefficient.
   - The name of an existing set. (For information on creating sets, see *The Set and Surface toolsets*.) You can enter a set that contains one or more nodes in the first row of the table. Subsequent sets can contain multiple nodes as long as all the sets contain an equal number of nodes.
   - The degree of freedom.
   - The ID of the coordinate system in which you will apply the constraint. You can either accept the default coordinate system or select an existing datum coordinate system. If the desired datum coordinate system does not exist, you can create it using the Datum toolset. (For more information, see *Creating datum coordinate systems*.)

     To determine the ID of a coordinate system, select **Tools->Query** from the main menu bar. For more information, see *Obtaining general information about the model*.

4. Click **OK** to save your equation definition and to close the constraint editor.

Additional information

- *Understanding constraints*
- *Linear Constraint Equations*

Using contact and constraint detection

The contact detection tool greatly simplifies the process for defining contact interactions and tie constraints in a model. This section explains in detail how to use the contact detection dialog box to locate and create interactions or constraints automatically.

For an overview of contact detection and a discussion of the search methods used by Abaqus/CAE, see *Understanding contact and constraint detection*.

In this section:

- *Specifying search criteria for contact detection*
- *Searching for contact pairs*
- *Reviewing and modifying detected contact pairs*
- *Creating interactions for automatically detected contact pairs*

Specifying search criteria for contact detection

The contact detection tool identifies potential contact pairs in a model using an array of user-defined parameters. Select **Interaction->Find contact pairs** or **Constraint->Find contact pairs** from the main menu, then enter the contact detection parameters.

In this section:

- *Specifying general search options for contact detection*
- *Specifying naming options for contact detection*
- *Specifying entity options for contact detection*
- *Defining default contact pair parameters*
- *Specifying advanced search options for contact detection*

Specifying general search options for contact detection

General search options indicate the range in which you want Abaqus/CAE to look for contact pair candidates. You can specify which areas of the model include contact pairs, as well as the estimated distance between surfaces that will require interaction or constraint definitions.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

    **Tip:** You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Display the **Search Options** tabbed page if it is not visible already.
3. Specify the **Search domain**:
   - Select **Whole model** to include every instance in the current model.
   - Select **Instance** to include only specified part instances and child part instances. Use the following procedure to specify the instances:
     1. Click .
     2. Select the instances from the viewport (for information on selecting instances, see *Selecting objects within the viewport*").
     3. After you have selected all of the desired instances, click **Done** in the prompt area.
   - Select **Displayed entities** to include only those portions of the model that are currently displayed in the viewport. You can use the assembly display options and display groups to control the display of the model. For details, see *Controlling instance visibility*, and *Using display groups to display subsets of your model*."
4. Enter the maximum distance between potential contact pair surfaces in the **Include pairs within separation tolerance** field. For tips on selecting a separation tolerance, see *Choosing a separation tolerance and extension angle*.
5. By default, Abaqus/CAE extends surface definitions to any faces (for geometry) or facets (for meshed models) within 20° of the surfaces detected by the search. You can adjust this angle by changing the value in the **Extend each surface found by angle** field. To prevent the extension of surface definitions, toggle off **Extend each surface found by angle**. For further details on how Abaqus/CAE extends surfaces, see *Additional criteria for defining contact pairs*.
6. By default, Abaqus/CAE creates contact pairs from surfaces located on different part instances or model instances. To search for contact pairs involving different regions on the same part instance or model instance or to search for potential self-contact interactions, toggle on **Include pairs with surfaces on the same instance**. For details, see *Defining contact within the same instance and self-contact*.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Specifying naming options for contact detection

In addition to creating contact pair interactions, the automatic contact detection tool also creates named surfaces for each surface involved in a contact pair (see *Default interaction and constraint parameters*). The **Names** tabbed page provides some options for tuning this surface creation behavior.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

   Tip: You can also find contact pairs using the tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Display the **Names** tabbed page.
3. In the **Name prefix** field, enter the prefix that will be used when naming each contact pair and surface (as discussed in *Default interaction and constraint parameters*).
4. To prevent the creation of named surfaces, toggle off **Name each surface found**. This option will not affect the creation of contact pairs.
5. The **Create additional named surfaces containing** field allows you to create additional composite surfaces:
   - To create a single named surface composed of all contact pair surfaces acting as a main surface, toggle on **All main**. Any main surfaces from preexisting contact pairs are also included in the created composite surface.
   - To create a single named surface composed of all contact pair surfaces acting as a secondary surface, toggle on **All secondary**. Any secondary surfaces from preexisting contact pairs are also included in the created composite surface.
   - To create a single named surface composed of all surfaces involved in contact pair definitions, toggle on **All**. Any surfaces from preexisting contact pairs are also included in the created composite surface.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Specifying entity options for contact detection

The **Entities** tabbed page allows you to control the types of geometry and elements that are included in a search. You can use these options to ignore certain features of a model or part instance that are not essential to contact interactions or constraints; for example, a layer of membrane elements on the outside of a solid body. Removing features from a search may also improve performance; for example, if all the contacting faces in a complex model are planar, removing cylindrical and spline-based features from the contact pair search may lead to faster search times. In general, the default entity options provide acceptable performance and should be modified only if a search is taking an excessively long time to complete.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

    **Tip:** You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Display the **Entities** tabbed page.
3. By default, Abaqus/CAE factors in shell thickness and offset properties when calculating the separation between shell entities. To ignore shell section properties and calculate separations based solely on the geometric representation of shell entities in a model, toggle off **Account for shell thickness and offset during search**.
4. When searching for geometry, toggle the following features in the **Search the following geometric entities** field to include or exclude them from the search domain:
   - **Planar** refers to all flat faces in a model. These features are toggled on by default.
   - **Cylindrical/Spherical/Toroidal** refers to curved faces in a model that were generated by extruding or revolving a straight, circular, or elliptical line. These features are toggled on by default.
   - **Spline-based** refers to irregularly curved faces in a model that were generated from spline lines or paths (see *Sketching splines* for more information). These features are toggled on by default.
5. When searching for meshed geometry, toggle the following element types in the **Search the following mesh entities** field to include or exclude them from the search domain:
   - **Solid** refers to faces on solid continuum elements. These elements are toggled on by default.
   - **Shell** refers to faces on shell elements. These elements are toggled on by default.
   - **Membrane** refers to faces on membrane or surface elements. These elements are toggled off by default.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Defining default contact pair parameters

Contact detection rules allow you to specify a set of conditional options that determine the default settings assigned to detected contact pairs. You can always change the contact pair settings after they are detected, but specifying rules before a search allows you to define appropriate contact pairs quickly according to your particular modeling intent. The rules are applied when Abaqus/CAE performs a search for contact pair candidates. For more information about contact pair default settings, see *Default interaction and constraint parameters*.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

   
   Tip: You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Display the **Rules** tabbed page.
3. Click .
   Abaqus/CAE displays the **Edit Rules** dialog box.
4. To create tie constraints for all contact pairs whose surfaces lie within a particular tolerance, toggle on **Use tie constraints when the separation value does not exceed** and enter the tolerance in the field provided. This rule is disabled by default.
5. To set the position tolerance on tie constraints equal to the surface separation reported by the contact detection tool, toggle on **Set tie position tolerance to separation value when nonzero**. This rule is disabled by default. For more information about position tolerances for tie constraints, see *Mesh Tie Constraints*.
6. To set the surface adjustment setting on contact interactions equal to the surface separation reported by the contact detection tool, toggle on **Adjust separated interactions by the separation value**. This rule is disabled by default. For more information about surface adjustment options, see *Contact Initialization for Contact Pairs in Abaqus/Standard*.
7. To move any overclosed secondary nodes in a contact interaction directly onto the main surface, toggle on **Adjust interactions to remove overclosure**. The adjustment is performed during the analysis. This rule is enabled by default. For more information about surface adjustment options, see *Contact Initialization for Contact Pairs in Abaqus/Standard*.
8. Click **OK**.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Specifying advanced search options for contact detection

The advanced search options provide an added level of control over the search range and the creation of contact pair surfaces. The general search options are sufficient in most cases; however, you may want to use the advanced options to account for unique modeling conditions. See *Tips for using the contact detection tool*, for a discussion of various modeling conditions and the recommended search techniques.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

   Tip: You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.

2. Display the **Advanced** tabbed page.
3. By default, Abaqus/CAE merges any contact pairs whose adjacent surfaces lie within 20° of each other (for details on this calculation, see *Additional criteria for defining contact pairs*). You can adjust this angle by changing the value in the **Merge pairs when surfaces are within angle** field. To prevent the merging of adjacent contact pairs, toggle off **Merge pairs when surfaces are within angle**.
4. By default, Abaqus/CAE includes overclosed and intersecting surfaces when searching for contact pairs. To ignore overclosed surfaces, toggle off **Include overclosed surfaces**. For details on how the automatic contact detection tool interprets overclosed surfaces, see *Detection of overclosed surfaces*.
5. By default, Abaqus/CAE does not create a contact pair candidate if two surfaces are laterally offset from each other (i.e., the surface normals from either surface never intersect the opposite surface). To include these surfaces in the contact pair candidates table, toggle on **Include opposing surfaces that do not overlap**. Such surfaces must still meet the separation and orientation requirements for the search. See *Additional criteria for defining contact pairs*, for further discussion of this option.
6. If your search includes part instances that have been meshed from geometry, indicate how Abaqus/CAE should interpret these instances during the search:

   - To treat the instances as geometry, select **Geometry**. Abaqus/CAE uses the **Geometry** option by default.
   - To treat the instances as element meshes, select **Mesh**.

   Tip: To detect contact between geometry and mesh instances, first mesh the geometry, then perform a mesh-based search.

   For a discussion of the differences between geometry and meshes, see *Contact detection for meshed models*.

7. To save the current search parameters as defaults for future sessions of Abaqus/CAE, click  (for more information, see *Saving the search parameters*). To reset saved search parameters to the original defaults, click .

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Searching for contact pairs

Once you have indicated the appropriate contact criteria, Abaqus/CAE searches your model to locate the surfaces that meet those criteria. For a discussion of how Abaqus/CAE interprets search criteria and detects contact pair candidates, see *Use of the contact detection algorithm*.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

   Tip: You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Enter the search parameters as described in *Specifying search criteria for contact detection*.
3. Click **Find Contact Pairs**.
4. Abaqus/CAE searches for potential contact pairs. When the search is complete, Abaqus/CAE populates the contact pair candidates table.

   To end a search before it is complete, click **Stop** in the prompt area.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Reviewing and modifying detected contact pairs

This section discusses to review and modify the contact pair definitions.

After searching for contact pairs (described in *Specifying search criteria for contact detection* and *Searching for contact pairs*), you have the opportunity to review the search results before creating interactions and tie constraints. The search results are presented in the contact pair candidates table: each contact pair comprises a row in the table; each column represents a parameter of the contact pair definition. Default values are provided for every necessary interaction parameter.

In this section:

- *Confirming search results in the viewport*
- *Configuring the layout of the contact pair candidates table*
- *Reviewing detected contact pairs*
- *Modifying contact pair definitions*
- *Creating suppressed interactions and constraints*
- *Miscellaneous review procedures*

Confirming search results in the viewport

As a visual aid, toggle on **Highlight in viewport** to highlight specified entities on your model in the viewport.

Select the entities you want to highlight:

Selected pairs

Highlights the surfaces involved in the contact pair or pairs that are currently selected in the contact pair candidates table. The main surface is highlighted in red; the secondary surface is highlighted in magenta. For axisymmetric or spherical surfaces, Abaqus/CAE also displays the axis of symmetry or spherical center for the surface; the axis or center point is used in surface smoothing calculations if they are applied (see *Modifying contact pair definitions*).

Selected main surfaces

Highlights the assigned main surfaces in the contact pair or pairs that are currently selected in the contact pair candidates table. For axisymmetric or spherical surfaces, Abaqus/CAE also displays the axis of symmetry or spherical center for the surface; the axis or center point is used in surface smoothing calculations if they are applied (see *Modifying contact pair definitions*).

Selected secondary surfaces

Highlights the assigned secondary surfaces in the contact pair or pairs that are currently selected in the contact pair candidates table. For axisymmetric or spherical surfaces, Abaqus/CAE also displays the axis of symmetry or spherical center for the surface; the axis or center point is used in surface smoothing calculations if they are applied (see *Modifying contact pair definitions*).

Search domain

Highlights the part instances in the model specified by the current **Search domain** field (see *Specifying general search options for contact detection*).

Selected pairs + search domain

Highlights both the currently selected contact pairs and the entire search domain.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Configuring the layout of the contact pair candidates table

A number of options are available to configure the display of the contact pair candidates table. You can customize the table to display only those elements that are essential to your modeling goals.

1. Toggle on **Show previously created interactions and ties** to add any previously defined contact interactions and tie constraints to the contact pair candidates table. You can edit previously defined interactions and ties in the same manner as newly detected contact pairs.
2. To limit the table display to only those contact pair candidates whose name contains a particular string, type that string in the **Name filter** field and press **[Enter]**. Click  to see examples of valid filtering syntax.

   Abaqus/CAE still creates interactions and constraints for contact pair candidates that are hidden by a filter.
3. Add or remove columns from the table display as necessary:
   a. Click mouse button 3 anywhere on the contact pair candidates table, and select **Edit Visible Columns** from the menu that appears.

      The **Edit Visible Columns** dialog box appears.
   b. In the **Edit Visible Columns** dialog box, toggle on the columns you want to display; toggle off the columns you want to hide:

Suppression

Indicates whether the contact pair will be active or suppressed upon creation (see *Creating suppressed interactions and constraints*). This column is hidden by default.

Name

Displays the name of the contact pair. This column is displayed by default.

Separation

Displays the distance between the two surfaces in the contact pair (see *Use of the contact detection algorithm*). This column is displayed by default.

Main surface name

Displays the name of the surface acting as the main in the contact pair. This column is hidden by default.

Secondary surface name

Displays the name of the surface acting as the secondary in the contact pair. This column is hidden by default.

Main instance name

Displays the name of the part instance on which the main surface is located. This column is hidden by default.

Secondary instance name

Displays the name of the part instance on which the secondary surface is located. This column is hidden by default.

Type

Indicates whether Abaqus/CAE is creating a contact interaction or a tie constraint for the contact pair. This column is displayed by default.

Sliding

Indicates the tracking approach used in the contact interaction formulation. This column is displayed by default.

Discretization

Indicates the surface discretization used in the contact interaction or tie constraint formulation. This column is displayed by default.

Property

Indicates the contact interaction property used in the contact formulation. This column is displayed by default.

Contact controls

Indicates the contact controls used in the contact formulation. This column is hidden by default and is unavailable in the initial step.

Adjust

For contact interactions, displays the range within which secondary nodes will be adjusted to lie precisely on the main surface. If nodes in a set are being adjusted, this column displays the name of the set.

For tie constraints, indicates whether or not all tied secondary nodes will be adjusted to lie precisely on the main surface.

This column is displayed by default.

Create step

Displays the name of the step in which Abaqus/CAE is activating the contact interaction or tie constraint. This column is hidden by default.

Surface smoothing

Indicates whether or not surface smoothing is applied to axisymmetric or spherical surfaces in the contact pair (see *Smoothing Contact Surfaces in Abaqus/Standard*). This column is displayed by default.

c. Do one of the following:

- Click **OK** to apply the settings and to close the **Edit Visible Columns** dialog box.

- Click **Apply** to apply the settings and to leave the **Edit Visible Columns** dialog box open.
- Click **Cancel** to ignore any changes and to close the **Edit Visible Columns** dialog box.

4. Sort the table contents according to parameter values by clicking the column heading for the parameter of interest. An up arrow in the column heading indicates the values are listed in ascending order; a down arrow indicates the parameter values are listed in descending order.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Reviewing detected contact pairs

You should review the contact pair candidates in the contact pair candidates table to ensure the surfaces involved are accurate and sufficient for your modeling needs.

Click on each contact pair name with the **Highlight in viewport** option toggled on to see the locations of each potential surface interaction or tie. If necessary you can add, remove, or modify the contact pair candidates.

1. Add a row to the table for any unidentified contact pairs you would like to include in your model:
   a. Click mouse button 3 anywhere in the contact pair candidates table, and select **Add** from the menu that appears. You can also click the  button above the contact pair candidates table.
   b. Select a main surface in the viewport, and click **OK** in the prompt area.

      For details on surface selection, see *Selecting objects within the viewport*.
   c. Select a secondary surface in the viewport, and click **OK**.

      Abaqus/CAE adds a contact pair candidate row to the contact pair candidates table based on the surfaces you selected. Default values are provided for each column in the row.
2. Remove any contact pair candidates that are not necessary for your modeling purposes. Click mouse button 3 anywhere on the row representing the contact pair candidate you want to remove, and select **Delete** from the menu that appears. You can also click the row you want to remove and then click the  button above the contact pair candidates table.
3. If multiple contact pairs will use identical parameters in their contact interaction or tie constraint definition, you can combine them into a single contact pair:
   a. In the contact pair candidates table, highlight the rows representing the contact pairs you want to combine (see *Selecting multiple items from lists and tables*, for instructions on selecting multiple rows in a table).
   b. Click mouse button 3 on any highlighted cell, and select **Merge** from the menu that appears. You can also click the  button above the contact pair candidates table.

      Abaqus/CAE replaces the selected contact pairs with a single contact pair. The selected main surfaces are merged into a single surface, and the selected secondary surfaces are merged into a single surface. The merged contact pair uses the parameters from the selected contact pair that appeared highest in the contact pair candidates table.

       **Note:**

      When combining contact pairs, the resultant merged surfaces must meet the orientation and connectivity requirements discussed in *About Contact Pairs in Abaqus/Standard* and *About Contact Pairs in Abaqus/Explicit*.

Modifying contact pair definitions

The contact pair candidates table presents all of the identified contact pair candidates in a format that allows you to easily review and modify the parameters of each contact pair definition.
Abaqus/CAE supplies default values for each parameter of the contact pair definition (see *Default interaction and constraint parameters*).

You can customize some of the default parameter settings to suit your modeling needs by using the **Rules** tabbed page (see *Defining default contact pair parameters*). However, you may still need to modify some parameters manually before creating interactions and constraints. The **Rules** options cannot account for such modeling issues as the appropriateness of small-sliding contact or assignment of varying frictional contact properties. You also may need to swap the assignment of main and secondary roles in contact pairs to ensure continuity and robustness.

To modify a parameter, change the value that is displayed in the corresponding cell of the contact pair candidates table. You modify cell values in one of four ways:

- To change a name parameter, click on the cell and type a new name. The old name is overwritten as you type.
- For cells that accept specific values, first click the cell to activate it. Then click the arrow next to the cell and select the desired value from the list that appears.
- For cells that require more detailed information to define the parameter, double-click the cell and enter the appropriate information in the dialog box that appears.
- For all cells other than name parameters, click mouse button 3 on the cell and select **Edit Cells** from the menu that appears. You can also click the cell and then click the  button above the contact pair candidates table. Enter the appropriate information in the dialog box that appears.

You cannot modify values for the following parameters:

- **Separation**
- **Main Instance**
- **Secondary Instance**

Each row of the contact pair candidates table represents a single contact pair candidate. The following instructions apply to a single row. Repeat them as often as necessary for each row in the table. For instructions on editing parameters for multiple contact pair candidates simultaneously, see *Miscellaneous review procedures*.

1. In the **Name** column, change the name of the contact pair, if desired.
2. In the **Main Surface** or **Secondary Surface** column, change the name of the generated main surface or secondary surface, if desired.

   The **Main Surface** and **Secondary Surface** columns are not displayed in the contact pair candidates table by default. See *Configuring the layout of the contact pair candidates table*, for instructions on adding these columns to the table.
3. Click mouse button 3 on any cell in the row, and select **Switch Surfaces** from the menu that appears to reverse the role of the main and secondary surfaces for the selected contact pair. You can also click the row and then click the  button above the contact pair candidates table. (To determine the role assignments in a contact pair, see *Confirming search results in the viewport*.)
4. To indicate whether a contact interaction or a tie constraint is created for the contact pair candidate, select **Interaction** or **Tie** in the **Type** column.

   You cannot change the **Type** for previously created tie constraints or interactions that include node-based regions.
5. To specify the tracking approach that will be used in an Abaqus/Standard contact formulation, select a value in the **Sliding** column:

- Select **Finite** to use the finite-sliding tracking approach.
- Select **Small** to use the small-sliding tracking approach.

Abaqus/CAE ignores the **Sliding** column when creating tie constraints and for Abaqus/Explicit analyses. For more information on the tracking approach, see *Contact Formulations in Abaqus/Standard*.

6. To specify the surface discretization that will be used in a contact or tie formulation, select a value in the **Discretization** column:

   - Select **Node-Surf** to use node-to-surface discretization.
   - Select **Surf-Surf** to use surface-to-surface discretization.

   For more information on surface discretization, see *Mesh Tie Constraints*; *Contact Formulations in Abaqus/Standard*; and *Contact Formulations for Contact Pairs in Abaqus/Explicit*.

7. For contact interactions, select the interaction property in the **Property** column.

   If the appropriate contact property is not available, you can create a new one:

   a. Click mouse button 3 anywhere in the contact pair candidates table, and select **Create Property** from the menu that appears.
   b. Create a contact interaction property as described in *Defining a contact interaction property*.

      When you have finished using the interaction property editor, the property you created will be available under the **Property** column in the contact pair candidates table.

   Abaqus/CAE ignores the **Property** column when creating tie constraints.

8. For contact interactions, select the contact controls in the **Contact Controls** column.

   The **Contact Controls** column is not displayed in the contact pair candidates table by default. See *Configuring the layout of the contact pair candidates table* for instructions on adding this column to the table.

   If the appropriate contact controls are not available, you can create them:

   a. Click mouse button 3 anywhere in the contact pair candidates table, and select **Create Contact Controls** from the menu that appears.
   b. Create the contact controls as described in *Customizing contact controls*.

      When you have finished specifying the contact controls, the controls you created will be available under the **Contact Controls** column in the contact pair candidates table.

   Abaqus/CAE ignores the **Contact Controls** column when creating tie constraints.

9. Use the adjustment options to reposition a secondary surface (or portions of a secondary surface) precisely onto the main surface at the beginning of an analysis:

   a. Double-click the cell in the **Adjust** column.

      The **Secondary Node/Surface Adjustment Options** dialog box appears.

   b. The options available in the **Secondary Node/Surface Adjustment Options** dialog box depend on the value in the **Type** column for the associated contact pair candidate:

      - If you are creating a contact interaction, select one of the following options:

No adjustment

Abaqus/CAE will not reposition any nodes or surfaces.

Adjust only to remove overclosure

Abaqus/CAE will reposition any initially overclosed secondary nodes or constraint points precisely onto the main surface.

Specify tolerance for adjustment zone

Abaqus/CAE will reposition any secondary nodes or constraint points located within the specified distance from the main surface precisely onto the main surface.

Adjust secondary nodes in set

Abaqus/CAE will adjust the secondary nodes or constraint points included in the specified geometry set precisely onto the main surface.

Abaqus/CAE ignores the **Adjust** column for contact interactions in an Abaqus/Explicit analysis. For more information about adjustment options for contact interactions, see *Contact Initialization for Contact Pairs in Abaqus/Standard*.

- If you are creating a tie constraint, you can modify both the adjustment options and the tie position tolerance in the **Secondary Node/Surface Adjustment Options** dialog box:
  - By default, Abaqus adjusts all tied secondary nodes so they lie precisely on the main surface. To prevent adjustment of the secondary nodes (but still tie the secondary node degrees of freedom to the main surface), toggle off **Adjust secondary surface initial position**.
  - Abaqus ties only those secondary nodes that lie within a specific distance from the main surface. To let Abaqus calculate a reasonable distance automatically, select **Use computed default**. To specify a distance directly, select **Specify distance**, and enter the distance in the field provided.

  For more information about adjustment options and position tolerances for tie constraints, see *Mesh Tie Constraints*.

c. Click **OK**.

10. To change the analysis step in which the contact pair is activated, select a step name in the **Create Step** column.

The **Create Step** column is not displayed in the contact pair candidates table by default. See *Configuring the layout of the contact pair candidates table*, for instructions on adding this column to the table.

11. For contact interactions using surface-to-surface discretization, select whether or not contact smoothing will be applied in the **Surface Smoothing** column:

- Select **Automatic** to let Abaqus/CAE identify any surfaces (or portions of surfaces) in the contact pair that are axisymmetric or spherical. If the **Highlight in viewport** option is enabled, Abaqus/CAE displays the axis of symmetry or spherical center for any identified surfaces in the viewport. Abaqus smooths these surfaces during contact calculations to minimize inaccuracies caused by mesh discretization. The **Automatic** option is available only for geometry models.
- Select **None** to prevent contact smoothing from being applied to any surfaces in the contact pair.

Abaqus/CAE ignores the **Surface Smoothing** column for tie constraints and for contact interactions in an Abaqus/Explicit analysis. For more information about contact surface smoothing, see *Smoothing Contact Surfaces in Abaqus/Standard*.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Creating suppressed interactions and constraints

In certain situations you may not be certain about the necessity of certain contact pair candidates. You can delete such candidates from the contact pair candidates table and avoid creating interactions for them, but you may later have to go back and recreate them if they prove necessary.
To address these situations, the contact detection tool offers the option to create interactions and constraints that are suppressed initially. Suppressed interactions and constraints are fully defined, but Abaqus ignores them when writing an input file or performing an analysis. A simple procedure activates these interactions and constraints and, thus, reintroduces them to the model definition (see *Suppressing and resuming objects*, for details).

There are two ways to create an initially suppressed contact pair:

- Click mouse button 3 anywhere in the row for the appropriate contact pair, and select **Suppress** from the menu that appears. The background of the row for the contact pair changes to gray, indicating that the created interaction or constraint will be suppressed initially. To return the contact pair to active status, click mouse button 3 on the appropriate row and select **Resume** from the menu that appears.
- Click the cell in the **Suppression** column for the appropriate contact pair. Click the cell again to return the contact pair to active status.



Note:

The **Suppression** column is not displayed by default. See *Configuring the layout of the contact pair candidates table*, for instructions on adding this column to the table.

A green check mark in the **Suppression** cell indicates that the created interaction or constraint is active. A red “X” in this cell indicates that the created interaction or constraint is suppressed initially.

Miscellaneous review procedures

Abaqus/CAE offers additional tools for manipulating data in the contact pair candidates table.

Recalculating the separation between surfaces

For part instances that are meshed from geometry, you can specify whether the contact detection tool treats these instances as geometry or an element mesh during the search (see *Specifying advanced search options for contact detection*). After performing the search, you can recalculate the separation between meshed surfaces in a contact pair candidate based on either the geometry or mesh representation. You can also use this technique to calculate the separation between surfaces in previously defined interactions or ties that appear in the contact pair candidates table (see *Configuring the layout of the contact pair candidates table*).

1. Click mouse button 3 on the row representing a contact pair, and select **Recalculate Separation** from the menu that appears. You can also click the row and then click the  button above the contact pair candidates table.
2. In the **Recalculate Separation** dialog box that appears, select **Mesh** to calculate surface separation based on the meshed representation of the model. Select **Geometry** to calculate surface separation based on the geometry representation of the model.
3. Click **OK**.

   Abaqus/CAE recalculates the separation between surfaces in the highlighted contact pair candidate and updates the cell in the **Separation** column.

Editing multiple cells

You can set multiple cells to the same value using a single procedure:

1. Highlight multiple cells within the same column (see *Selecting multiple items from lists and tables*).
2. Click mouse button 3 anywhere within the highlighted cells, and select **Edit Cells** from the menu that appears.

   You can also click the  button above the contact pair candidates table.
3. In the dialog box that appears, select the desired value.
4. Click **OK**.

   Abaqus/CAE sets all of the highlighted cells to the selected value.

Highlighting the entire table

To quickly highlight every cell in the table, click mouse button 3 anywhere in the table and select **Select All** from the menu that appears.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*

Creating interactions for automatically detected contact pairs

Abaqus/CAE creates contact interactions and tie constraints for every item in the contact pair candidates table based on the parameters you have specified.

1. From the main menu, select either **Interaction->Find contact pairs** or **Constraint->Find contact pairs**.

   Tip: You can also find contact pairs using the  tool in the Interaction module toolbox.

   Abaqus/CAE displays the contact detection dialog box.
2. Enter the search parameters as described in *Specifying search criteria for contact detection*.
3. Search for contact pairs as described in *Searching for contact pairs*.
4. In the contact pair candidates table, enter all of the parameters necessary to define the interactions and ties as described in *Reviewing and modifying detected contact pairs*.
5. Click **OK**.

   Abaqus/CAE creates the specified contact interactions and tie constraints.

Additional information

- *Understanding contact and constraint detection*
- *Using contact and constraint detection*
- *Mesh Tie Constraints*
- *About Contact Pairs in Abaqus/Standard*
- *About Contact Pairs in Abaqus/Explicit*

Using the connector section editors

After you specify a connection type for a connector section, you can define connector behaviors for connection types that have one or more available components of relative motion. If you change the connection type after you have defined connector behaviors, you will have to delete the connector behaviors and redefine them for the new connection type. This section explains how to enter data related to connector behaviors in the connector section editor.

In this section:

- *Defining elasticity*
- *Defining damping*
- *Defining friction*
- *Configuring tangential behavior for connector friction*
- *Specifying predefined friction parameters or contact forces*
- *Defining plasticity*
- *Defining damage*
- *Defining damage evolution*
- *Defining a stop*
- *Defining a lock*
- *Defining failure*
- *Defining a reference length*
- *Defining time integration*
- *Specifying behavior settings for tabular data*
- *Specifying connector derived components*
- *Specifying potential terms*

Defining elasticity

You can define spring-like elasticity behavior for the available components of relative motion. For more information, see *Connector Elastic Behavior*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new elasticity behavior, click **Add** and select **Elasticity** from the menu that appears.
   - To edit an existing elasticity behavior, select the behavior from the **Behavior Options** list to display the associated data fields for that option.
3. Choose the **Definition**.
   - Choose **Linear** to define linear elastic stiffnesses.
   - Choose **Nonlinear** to define forces/moments as tabular functions of available components of relative motion.
   - Choose **Rigid** to define rigid-like elastic behavior.
4. If you are defining linear elastic behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments consistent with the available components of relative motion for which you are defining elastic behavior.
   b. In the **Coupling** field, choose one of the following:
      - Choose **Uncoupled** to specify the individual elastic stiffnesses for the available components of relative motion; for example, **D11**, **D22**, and **D33**, as determined by **Force/Moment** selections. You can use a single elasticity behavior to specify all of the stiffnesses, even if the values are different for each available component of relative motion.
      - Choose **Coupled** to specify elastic stiffnesses coupled with the available components of relative motion; for example, **D11**, **D12**, and **D13**, as determined by **Force/Moment** selections.
5. If you are defining nonlinear, uncoupled behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments that are consistent with the available components of relative motion for which you are defining elastic behavior. If the behavior is the same for multiple components, you can define a single elasticity behavior that will use this one function. If the behavior is different for multiple components, you must define separate elasticity behaviors.
   b. Choose **Uncoupled** in the **Coupling** field to specify forces/moments as a tabular function of their respective available components of relative motion.
6. If you are defining nonlinear, coupled behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments that are consistent with the available components of relative motion for which you are defining elastic behavior.
   b. Choose **Coupled on position** or **Coupled on motion** in the **Coupling** field to specify forces/moments as functions of one or more relative position or constitutive displacement/rotation components, respectively.

c. In the **Independent components** field, toggle on the available components of relative position or constitutive motion for which you are defining coupled elastic behavior. You may need to use the unsymmetric equation solver in the step editor to improve convergence.

7. If you are defining rigid elastic behavior, toggle on the available components of relative motion for which you are defining rigid elastic behavior.
8. To define behavior data that depend on frequency, temperature, or field variables, do the following:
   a. If available, toggle on **Use frequency-dependent data** to define behavior data that vary with frequency. A column labeled **Frequency** appears in the data table.
   b. Toggle on **Use temperature-dependent data** to define behavior data that vary with temperature. A column labeled **Temp** appears in the data table.
   c. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
9. Enter data in the table. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
10. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*.
11. Select one of the following:
    - If you want to continue defining behaviors, click **Add**, select the desired behavior, and continue the connector section definition. For instructions on defining other behaviors, see *Using the connector section editors*.
    - If you want to view or edit an existing behavior, select the behavior from the **Behavior Options** list. For instructions on editing behaviors, see *Using the connector section editors*.
    - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector Elastic Behavior*
- *Connector section editors*
- *Creating connector sections*

Defining damping

You can define dashpot-like damping behavior for the available components of relative motion. For more information, see *Connector Damping Behavior*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new damping behavior, click **Add** and select **Damping** from the menu that appears.
   - To edit an existing damping behavior, select the behavior from the **Behavior Options** list to display the associated data fields for that behavior.
3. Choose the **Damping type**.
   - Choose **Viscous** to specify velocity proportional damping.
   - Choose **Structural** to specify displacement proportional damping.
4. Choose the **Definition**.
   - Choose **Linear** to define linear damping coefficients.
   - Choose **Nonlinear** to define forces/moments as tabular functions of the relative velocities.
5. If you are defining linear damping behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments consistent with the relative velocities for which you are defining damping behavior.
   b. In the **Coupling** field, choose one of the following:
      - Choose **Uncoupled** to specify the individual damping coefficients associated with the relative velocities; for example, **C11**, **C22**, and **C33**, as determined by **Force/Moment** selections. You can use a single damping option to specify all the coefficients, even if the values are different for each relative velocity.
      - Choose **Coupled** to specify damping coefficients coupled associated with the relative velocities; for example, **C11**, **C12**, and **C13**, as determined by **Force/Moment** selections.
6. If you are defining nonlinear, uncoupled behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments that are consistent with the relative velocities for which you are defining damping behavior. If the behavior is the same for multiple components, you can define a single damping behavior that will use this one function. If the behavior is different for multiple components, you must define separate damping behaviors.
   b. Choose **Uncoupled** in the **Coupling** field to specify forces/moments as a tabular function of their respective available components of relative velocity.
7. If you are defining nonlinear, coupled behavior,
   a. In the **Force/Moment** field, toggle on the forces or moments that are consistent with the relative velocities for which you are defining damping behavior.
   b. Choose **Coupled on position** or **Coupled on motion** in the **Coupling** field to specify forces/moments as functions of one or more relative position or velocity components, respectively.

c. In the **Independent components** field, toggle on the available components of relative position or constitutive motion for which you are defining coupled damping behavior. You might need to use the unsymmetric equation solver in the step editor to improve convergence.

8. To define behavior data that depend on frequency, temperature, or field variables, do the following:
   a. If available, toggle on **Use frequency-dependent data** to define behavior data that vary with frequency. A column labeled **Frequency** appears in the tabular data area.
   b. Toggle on **Use temperature-dependent data** to define behavior data that vary with temperature. A column labeled **Temp** appears in the tabular data area.
   c. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the tabular data area.
9. Enter damping data in the table. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
10. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*.
11. Select one of the following:
    - If you want to continue defining behaviors, click **Add**, select the desired behavior, and continue the connector section definition. For instructions on defining other behaviors, see *Using the connector section editors*.
    - If you want to view or edit an existing behavior, select it from the **Behavior Options** list. For instructions on editing behaviors, see *Using the connector section editors*.
    - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector Damping Behavior*
- *Connector section editors*
- *Creating connector sections*

Defining friction

You can define frictional effects for the available components of relative motion. For more information, see *Connector Friction Behavior*.

1. Display the connector section editor.
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new friction behavior, click **Add** and select **Friction** from the menu that appears.
   - To edit an existing friction behavior, select the behavior from the **Behavior Options** list to display the associated data fields for that option.
3. Choose the **Friction model**. For more information, see *What types of friction models are available?*.
   - Choose **Predefined** to use a predefined friction model. Abaqus displays the slip direction along which friction occurs.
   - Choose **User-defined** to specify the slip direction in which friction forces or moments are applied and to define friction-generating contact force contributions.
4. If you are specifying a user-defined friction model, choose the **Slip direction**.
   - Choose **Specify direction** to specify the available component of relative motion along which friction forces or moments are applied. You can define multiple friction behaviors to model friction for more than one component, but Abaqus allows only one friction behavior for each component.
   - Choose **Compute using force potential** to compute the instantaneous slip direction using a force potential. Select the **Force Potential** tabbed page, and define at least one force potential term. For more information, see *Specifying potential terms*.
5. Choose the **Stick stiffness**.
   - Choose **Use default** to have Abaqus compute an appropriate stick stiffness.
   - Choose **Specify** to enter the stick stiffness associated with frictional behavior in the direction specified by the slip direction.
6. Configure the **Tangential Behavior**. For more information, see *Configuring tangential behavior for connector friction*.
7. If you are specifying a predefined friction model, specify the **Predefined Friction Parameters**. If you are specifying a user-defined friction model, specify the **Contact Force**. For more information, see *Specifying predefined friction parameters or contact forces*.
8. Select one of the following:
   - If you want to continue defining behaviors, click **Add**, select the desired behavior, and continue the connector section definition. For instructions on defining other behaviors, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior, select the behavior from the **Behavior Options** list. For instructions on editing behaviors, see *Using the connector section editors*.

- If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

• *Connector Friction Behavior*
• *What types of friction models are available?*
• *Connector section editors*
• *Creating connector sections*
• *Specifying potential terms*

Configuring tangential behavior for connector friction

You can select a friction formulation for predefined and user-defined connector friction models.

You use the **Tangential Behavior** tabbed page in the connector section editor to make the selection. For more information, see *Using the Basic Coulomb Friction Model*. You can choose from the following:

- **Penalty**. Specify a friction coefficient.
- **Static-Kinetic Exponential Decay**. Specify separate static and kinetic friction coefficients with a smooth transition zone defined by an exponential curve.

Additional information

• *Defining friction*

Configure tangential behavior using the Penalty friction formulation

1. Display the connector section editor by following the procedure outlined in *Defining friction*, and select the **Tangential Behavior** tabbed page.
2. From the **Friction formulation** field, select **Penalty**.
3. On the **Friction Coefficient** tabbed page, enter the following data:

Friction Coeff

Friction coefficient, μ.

Slip Rate

If desired, toggle on **Use slip-rate-dependent data** and enter the slip rate, $\dot{\gamma}_{eq}$.

Contact Pressure

If desired, toggle on **Use contact-pressure-dependent data** and enter the contact pressure, p.

Temp

If desired, toggle on **Use temperature-dependent data** and enter the average temperature at the contact point, $\overline{\theta}$, between the two contact surfaces.

Field 1, Field 2, etc.

If desired, specify the **Number of field variables** and enter the average value of the first field variable, $\overline{f}^1$, the average value of the second field variable, $\overline{f}^2$, etc.

4. On the **Shear Stress** tabbed page, specify the shear stress limit.

No limit

Indicates no limit on the equivalent shear stress.

Specify

Enter the equivalent shear stress limit, τ_{max}; that is, the maximum achievable value of the equivalent shear stress.

5. On the **Elastic Slip** tabbed page, choose a method for modifying the allowable elastic slip (applies only to Abaqus/Standard analyses).

Fraction of characteristic model dimension

Enter the ratio of the allowable maximum elastic slip to a characteristic model dimension. The default value is 0.0001.

Absolute distance

Enter the absolute magnitude of the allowable elastic slip, γ_i, to be used in the stiffness method for sticking friction.

Configure tangential behavior using the Static-Kinetic Exponential Decay friction formulation

1. Display the connector section editor by following the procedure outlined in *Defining friction*, and select the **Tangential Behavior** tabbed page.
2. From the **Friction formulation** field, select **Static-Kinetic Exponential Decay**.
3. On the **Friction Coefficients** tabbed page, choose the exponential decay definition and enter the coefficients or test data.

- **Coefficients**. Provide the static friction coefficient, kinetic friction coefficient, and decay coefficient directly.

Static Coeff

Enter the static friction coefficient, μ_s.

Kinetic Coeff

Enter the kinetic friction coefficient, μ_k.

Decay Coeff

Enter the decay coefficient, d_c.

- **Test data**. Provide test data points to fit the exponential model.

Friction Coeff

- Enter the static coefficient of friction specified at $\dot{\gamma}_{eq} = 0.0$.
- Enter the dynamic friction coefficient, an experimental measurement taken at a reference slip rate, $\dot{\gamma}_2$.
- Enter the kinetic friction coefficient, μ_∞. This value corresponds to the asymptotic value of the friction coefficient at infinite slip rate, $\dot{\gamma}_\infty$.

Slip Rate

Enter the reference slip rate, $\dot{\gamma}_2$, used to measure the dynamic friction coefficient.

4. On the **Elastic Slip** tabbed page, choose a method for modifying the allowable elastic slip (applies only to Abaqus/Standard analyses).

Fraction of characteristic model dimension

Enter the ratio of the allowable maximum elastic slip to a characteristic model dimension. The default value is 0.0001.

Absolute distance

Enter the absolute magnitude of the allowable elastic slip, γ_i, to be used in the stiffness method for sticking friction.

Specifying predefined friction parameters or contact forces

You can specify geometric constants and internal contact forces for predefined connector friction models and a friction-generating contact force and internal contact forces for user-defined connector friction models.
You use the **Predefined Friction Parameters** tabbed page in the connector section editor to specify geometric constants and internal contact forces for predefined connector friction models. You use the **Contact Force** tabbed page in the connector section editor to specify a friction-generating contact force and internal contact forces for user-defined connector friction models. For more information, see *Connector Friction Behavior*.

Additional information

- *Defining friction*
- *Specifying connector derived components*

Specify predefined friction parameters

1. Display the connector section editor by following the procedure outlined in *Defining friction*, and enter the necessary data.
2. Select **Predefined** as the **Friction model**.
3. Select the **Predefined Friction Parameters** tabbed page.
4. Specify the geometric constants and internal contact forces depending on the connection type that you have selected. To display the connection type description that defines the required data entries, see *Connection types*.

Specify a friction-generating contact force and internal contact forces

1. Display the connector section editor by following the procedure outlined in *Defining friction*, and enter the necessary data.
2. Select **User-defined** as the **Friction model**.
3. Select the **Contact Force** tabbed page.
4. If desired, toggle on **Specify component** to define a friction-generating contact force/moment using an intrinsic connector component or a derived connector component.
   - Choose **Intrinsic component**, and select the force/moment that is consistent with the available component of relative motion for which you are defining the friction-generating contact force/moment.
   - Choose **Derived component**, and click  to define a connector derived component. For more information, see *Specifying connector derived components*.
5. If **Specify component** is toggled off, you must enter a value for the internal contact force in the data table in the **Internal Contact Force** portion of the **Contact Force** tabbed page. If desired, you can do the following to specify the internal contact force:

Use independent components

Toggle on **Use independent components**, choose **Position** or **Motion**, and select the available components of relative position or constitutive motion to be used as independent variables. Enter the connector relative position or constitutive motion in the direction specified by the independent component in the corresponding field of the table.

Accum Slip

Toggle on **Use accumulated slip dependence**, and enter the accumulated slip in the slip direction.

Temp

Toggle on **Use temperature-dependent data**, and enter the temperature.

Field 1, Field 2, etc.

Specify the **Number of field variables**, and enter the value of the first field variable, the value of the second field variable, etc.

Internal Contact Force

Specify the internal contact force data.

6. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*.

Defining plasticity

You can define plasticity behavior for the available components of relative motion. For more information, see *Connector Plastic Behavior*. If you specify a plasticity behavior option, you must also specify an elasticity behavior option.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new plasticity behavior, click **Add** and select **Plasticity** from the menu that appears.
   - To edit an existing plasticity behavior, select the behavior from the **Behavior Options** list to display the associated data fields for that behavior.
3. If you are defining uncoupled plasticity behavior,
   a. Choose **Uncoupled** in the **Coupling** field to specify forces/moments as a tabular function of their respective available components of relative motion.
   b. In the **Force/Moment** field, toggle on the forces or moments that are consistent with the available components of relative motion for which you are defining plasticity behavior. If the behavior is the same for multiple components, you can define a single plasticity behavior that will use this one function. If the behavior is different for multiple components, you must define separate plasticity behaviors.
4. If you are defining coupled plasticity behavior,
   a. Choose **Coupled** in the **Coupling** field.
   b. Select the **Force Potential** tabbed page, and define at least one force potential term. For more information, see *Specifying potential terms*.
5. Select the hardening behavior.
   - Toggle on **Specify isotropic hardening** to define the initial yield value and, optionally, the evolution of the yield surface size, F^0, as a function of the equivalent plastic relative motion, $\overline{u}^{pl}$.
   - Toggle on **Specify kinematic hardening** to define the translation of the yield surface in force space through the backforce, α.

   At least one hardening behavior, isotropic or kinematic, must be defined. You can select both types of hardening to define a combined isotropic/kinematic hardening behavior.
6. If you toggled on **Specify isotropic hardening**:
   a. Select the **Isotropic Hardening** tabbed page.
   b. Choose the **Definition**.
      - Choose **Tabular** to specify the force-constitutive motion data directly in tabular form.
      - Choose **Exponential Law** to specify the material parameters of the exponential law used to calculate the equivalent force defining the size of the yield surface.
7. If you toggled on **Specify kinematic hardening**:
   a. Select the **Kinematic Hardening** tabbed page.
   b. Choose the **Definition**.

- Choose **Half-cycle** to specify the force-constitutive motion data obtained from the first half cycle of a unidirectional tension or compression experiment.
- Choose **Stabilized** to specify the force-constitutive motion data obtained from the stabilized cycle of a specimen that is subjected to symmetric cycles.
- Choose **Parameters** to specify the material parameters directly.

8. To define behavior data that depend on temperature or field variables, do the following on the **Isotropic Hardening** or **Kinematic Hardening** tabbed page:
   a. Toggle on **Use temperature-dependent data** to define behavior data that vary with temperature. A column labeled **Temp** appears in the tabular data area.
   b. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the tabular data area.
9. Enter plastic hardening data in the tables on the **Isotropic Hardening** and/or **Kinematic Hardening** tabbed pages. You can enter data into the tables using the keyboard. Alternatively, you can click mouse button 3 anywhere in the tables to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
10. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*. For an Abaqus/Explicit analysis that includes isotropic hardening using a **Tabular** definition, you can also specify settings for the evaluation of rate-dependent data.
11. Select one of the following:
    - If you want to continue defining behaviors, click **Add**, select the desired behavior, and continue the connector section definition. For instructions on defining other behaviors, see *Using the connector section editors*.
    - If you want to view or edit an existing behavior, select it from the **Behavior Options** list. For instructions on editing behaviors, see *Using the connector section editors*.
    - If you want to save your connector section definition and exit the editor, click **OK**.

Additional information

- *Connector Plastic Behavior*
- *Connector section editors*
- *Creating connector sections*
- *Specifying potential terms*

Defining damage

You can define damage behavior for the available components of relative motion. For more information, see *Connector Damage Behavior*. If you specify a damage behavior option, you must also specify an elasticity behavior option. In addition, if you are defining plastic motion–based damage initiation behavior, you must also specify a plasticity behavior option.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new damage behavior, click **Add** and select **Damage** from the menu that appears.
   - To edit an existing damage behavior, select the behavior from the **Behavior Options** list to display the associated data fields for that behavior.
3. In the **Coupling** field, choose one of the following:
   - Choose **Uncoupled** to specify damage criteria for each available component of relative motion independently.
   - Choose **Coupled** to specify damage criteria that couple all or some of the available components of relative motion.
4. If you are defining uncoupled damage behavior, toggle on the forces or moments that are consistent with the available components of relative motion for which you are defining damage behavior in the **Force/Moment** field. If the behavior is the same for multiple components, you can define a single damage behavior that will use this one function. If the behavior is different for multiple components, you must define separate damage behaviors.
5. Choose the **Initiation criterion**.
   - Choose **Force** to specify the damage initiation criterion in terms of forces/moments in the connector. You provide the lower (compression) limit, the upper (tension) limit, or both limits for the force/moment damage initiation values.
   - Choose **Motion** to specify the damage initiation criterion in terms of relative constitutive displacements/rotations in the connector. You provide the lower (compression) limit, the upper (tension) limit, or both limits for the constitutive displacement/rotation damage initiation values.
   - Choose **Plastic motion** to specify the damage initiation criterion in terms of an equivalent relative plastic motion in the connector. You provide the relative equivalent plastic displacement/rotation at which damage will be initiated as a function of the relative equivalent plastic rate. You must also specify a plasticity behavior option; see *Defining plasticity*, for more information.
6. If you are defining coupled damage behavior, you must specify a connector potential as follows:
   - If you selected **Force** or **Motion** as the **Initiation criterion**, you must specify a connector potential to define an equivalent force magnitude or an equivalent motion magnitude. Select the **Initiation Potential** tabbed page, and define at least one force potential term. For more information, see *Specifying potential terms*.
   - If you selected **Plastic motion** as the **Initiation criterion**, you must specify a connector potential in a coupled connector plasticity behavior option to define an equivalent relative plastic motion. To define coupled connector plasticity behavior using a force potential, see *Defining plasticity*.

If the coupled plasticity definition includes at least two terms in the force potential, you can provide the **Mode-Mix Ratio** in the data table on the **Initiation** tabbed page of the damage behavior option to reflect the relative weight of the first two terms in their contribution to the potential. See *Mode Mix Ratio*, for information on how this quantity is defined.

7. To define the damage initiation criterion, do the following on the **Initiation** tabbed page:
   a. To define a damage initiation criterion that depends on temperature or field variables:
      1. Toggle on **Use temperature-dependent data** to define behavior data that vary with temperature. A column labeled **Temp** appears in the data table.
      2. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.

   b. Enter the appropriate damage initiation criterion data in the table. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
   c. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*. For an Abaqus/Explicit analysis that includes plastic motion–based damage initiation criteria, you can also specify settings for the evaluation of rate-dependent data.
8. If desired, define the damage evolution law that specifies how the damage variable evolves, as described in *Defining damage evolution*.
9. Select one of the following:
   - If you want to continue defining behaviors, click **Add**, select the desired behavior, and continue the connector section definition. For instructions on defining other behaviors, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior, select it from the **Behavior Options** list. For instructions on editing behaviors, see *Using the connector section editors*.
   - If you want to save your connector section definition and exit the editor, click **OK**.

Additional information

- *Connector Damage Behavior*
- *Connector section editors*
- *Creating connector sections*
- *Defining plasticity*
- *Defining damage evolution*

Defining damage evolution

Connector damage evolution specifies the evolution law for the damage variable. Upon evolution, the connector response will be degraded. If you do not specify a damage evolution law for a particular damage behavior, the associated damage variable is held fixed at 0.0 and the damage behavior does not contribute to degrading the response in the connector. For more information, see *Connector Damage Behavior*.

1. Define a damage behavior as described in *Defining damage*.
2. In the **Edit Connector Section** dialog box, toggle on **Specify damage evolution** and select the **Evolution** tabbed page.
3. If you are defining uncoupled force-based or constitutive motion-based damage initiation behavior,
   - Choose **Motion** as the **Evolution type** to define a motion-based damage evolution law, and choose the **Evolution softening** type.
     - Choose **Linear** to define a linear damage evolution law. You provide the difference between the constitutive relative motion at ultimate failure and the constitutive relative motion at damage initiation.
     - Choose **Exponential** to define an exponential damage evolution law. You provide the difference between the relative motions at ultimate failure and at damage initiation and the exponential coefficient.
     - Choose **Tabular** to define the damage variable directly as a tabular function of the differences between the relative motions at ultimate failure and the relative motions at damage initiation.
   - Choose **Energy** as the **Evolution type** to define an energy-based damage evolution law. You provide the post-damage initiation dissipated energy at ultimate failure.
4. If you are defining uncoupled plastic motion–based damage initiation behavior,
   a. Define an uncoupled plasticity behavior option. For more information, see *Defining plasticity*.
   b. Choose the **Evolution type**.
      - Choose **Motion** to define a motion-based damage evolution law, and choose the **Evolution softening** type. The equivalent plastic relative motion is calculated from the associated plasticity behavior definition.
        - Choose **Linear** to define a linear damage evolution law. You provide the difference between the associated equivalent plastic relative motion at ultimate failure and the associated equivalent plastic relative motion at damage initiation.
        - Choose **Exponential** to define an exponential damage evolution law. You provide the difference between the equivalent relative plastic motions at ultimate failure and at damage initiation and the exponential coefficient.
        - Choose **Tabular** to define the damage variable directly as a tabular function of the differences between the equivalent relative plastic motions at ultimate failure and the relative motions at damage initiation.
      - Choose **Energy** to define an energy-based damage evolution law. You provide the post-damage initiation dissipated energy at ultimate failure.
5. If you are defining coupled force-based or constitutive motion-based damage initiation behavior,
   - Choose **Motion** as the **Evolution type** to define a motion-based damage evolution law and do the following:

1. Choose the **Evolution softening** type.
   - Choose **Linear** to define a linear damage evolution law. You provide the difference between the equivalent motion at ultimate failure and the equivalent motion at damage initiation.
   - Choose **Exponential** to define an exponential damage evolution law. You provide the difference between the relative motions at ultimate failure and at damage initiation and the exponential coefficient.
   - Choose **Tabular** to define the damage variable directly as a tabular function of the differences between the relative motions at ultimate failure and the relative motions at damage initiation.
2. Select the **Evolution Potential** tabbed page, and define at least one force potential term. For more information, see *Specifying potential terms*.

- Choose **Energy** as the **Evolution type** to define an energy-based damage evolution law. You provide the post-damage initiation dissipated energy at ultimate failure.

6. If you are defining coupled plastic motion–based damage initiation behavior,
   a. Define a coupled plasticity behavior option. For more information, see *Defining plasticity*.

      If the associated coupled plasticity definition includes at least two terms in the force potential, the data that you provide for damage evolution may also be a function of the mode mix ratio. The **Mode-Mix Ratio** defines the relative weight of the first two terms in their contribution to the potential. See *Mode Mix Ratio*, for information on how this quantity is defined.
   b. Choose the **Evolution type**.
      - Choose **Motion** to define a motion-based damage evolution law, and choose the **Evolution softening** type. The equivalent plastic relative motion is calculated from the associated plasticity behavior definition.
        - Choose **Linear** to define a linear damage evolution law. You provide the difference between the associated equivalent plastic relative motion at ultimate failure and the associated equivalent plastic relative motion at damage initiation.
        - Choose **Exponential** to define an exponential damage evolution law. You provide the difference between the equivalent relative plastic motions at ultimate failure and at damage initiation and the exponential coefficient.
        - Choose **Tabular** to define the damage variable directly as a tabular function of the differences between the equivalent relative plastic motions at ultimate failure and the relative motions at damage initiation.
      - Choose **Energy** to define an energy-based damage evolution law. You provide the post-damage initiation dissipated energy at ultimate failure.

7. On the **Evolution** tabbed page, toggle on **Specify affected components** and toggle on the forces or moments that are consistent with the available components of relative motion that will be affected by the damage evolution law.
8. Choose the **Degradation** type to specify the contribution of this damage behavior to the overall damage effect if several damage behaviors are defined for the same connector section.
   - Choose **Maximum** to compare the damage value associated with this behavior to the damage values from any other damage behaviors defined for this connector section and consider only the maximum value for the overall damage.
   - Choose **Multiplicative** to combine the damage values for all the damage behaviors associated with this connector section in a multiplicative fashion to obtain the overall damage.

9. To define a damage evolution law that depends on temperature or field variables:
   a. Toggle on **Use temperature-dependent data** to define behavior data that vary with temperature. A column labeled **Temp** appears in the data table.
   b. To define behavior data that depend on field variables, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables. Field variable columns appear in the data table.
10. Enter the appropriate damage evolution law data in the table. You can enter data into the table using the keyboard. Alternatively, you can click mouse button 3 anywhere in the table to view a list of options for specifying tabular data. For detailed information on each option, see *Entering tabular data*.
11. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*.

Additional information

- *Connector Damage Behavior*
- *Connector section editors*
- *Creating connector sections*
- *Defining elasticity*
- *Defining plasticity*
- *Defining damage*

Defining a stop

You can limit values of the admissible range of motion for one or more available components of relative motion. For more information, see *Connector Stops and Locks*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new stop option, click **Add** and select **Stop** from the option menu that appears.
   - To edit an existing stop option, select the option from the **Behavior Options** list to display the associated data fields for that option.
3. In the **Components** field, toggle on all of the available components of relative motion for which you are defining a stop. If you want to specify different limits for different components, you must specify multiple stop options.
4. In the **Admissible Positions at Stop** portion of the editor, enter the limits for the admissible positions of the available components of relative motion that will trigger the stop.
5. Select one of the following:
   - If you want to continue defining behavior options, click **Add**, select the desired behavior option, and continue the connector section definition. For instructions on defining other behavior options, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior option, select the option from the **Behavior Options** list. For instructions on editing behavior options, see *Using the connector section editors*.
   - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector Stops and Locks*
- *Connector section editors*
- *Creating connector sections*

Defining a lock

You can specify locking criteria for any component of relative motion. The locking criteria may depend on the motions of the available components of relative motion and on the forces/moments in both available and constrained components of relative motion. The locking occurs when any one of the criteria are met, as opposed to when all of the criteria are met. For more information, see *Connector Stops and Locks*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new lock option, click **Add** and select **Lock** from the option menu that appears.
   - To edit an existing lock option, select the option from the **Behavior Options** list to display the associated data fields for that option.
3. In the **Components** field, toggle on the components of relative motion to be used to define locking criteria. If you want to specify different limits for different components, you must define multiple lock options.
4. Click the arrow to the right of the **Lock** field, and select an option from the list that appears:
   - Select **All** to lock all components of relative motion when any of the locking criteria are satisfied.
   - Select **Specify** to choose the available component of relative motion to lock when any of the locking criteria are satisfied.
5. To define locking criteria based on motion, in the **Position Locking Criteria** portion of the editor, enter the limits for the relative positions of the available components of relative motion that will trigger the lock.
6. To define locking criteria based on forces/moments, in the **Force/Moment Locking Criteria** portion of the editor, enter the limits for the forces/moments of the available or constrained components of relative motion that will trigger the lock.
7. Select one of the following:
   - If you want to continue defining behavior options, click **Add**, select the desired behavior option, and continue the connector section definition. For instructions on defining other behavior options, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior option, select the option from the **Behavior Options** list. For instructions on editing behavior options, see *Using the connector section editors*.
   - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector Stops and Locks*
- *Connector section editors*
- *Creating connector sections*

Defining failure

For an Abaqus/Standard analysis, you can specify failure criteria for any available components of relative motion. For an Abaqus/Explicit analysis, the failure criteria may depend on the motions of the available components of relative motion and on the forces/moments in both available and constrained components of relative motion. The failure occurs when any of the criteria are met, as opposed to when all of the criteria are met. For more information, see *Connector Failure Behavior*.

When you define failure criteria that depends on forces/moments, the sum of the forces/moments from all of the behavior options is compared to the failure criteria. In an Abaqus/Standard analysis, for the failure criteria to be valid, you must define a behavior option, such as elasticity, that generates a force/moment for the available components of relative motion used to define the failure criteria.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new failure option, click **Add** and select **Failure** from the option menu that appears.
   - To edit an existing failure option, select the option from the **Behavior Options** list to display the associated data fields for that option.
3. In the **Components** field, toggle on the components of relative motion to be used to define failure criteria. If you want to specify different limits for different components, you must define multiple failure options.
4. Click the arrow to the right of the **Release** field, and select an option from the list that appears:
   - Select **All** to release all of the components of relative motion when any of the failure criteria are satisfied. The connector will have no effect on the analysis beyond that point.
   - Select **Specify** to choose the component of relative motion to release when any of the failure criteria are satisfied.
5. To define failure criteria based on motion, in the **Position Failure Criteria** portion of the editor, enter the limits for the relative positions of the available components of relative motion that will cause failure.
6. To define failure criteria based on forces/moments, in the **Force/Moment Failure Criteria** portion of the editor, enter the limits for the forces/moments of the available components of relative motion that will cause failure. In an Abaqus/Explicit analysis, you can also enter the limits for the forces/moments of the constrained components of relative motion that will cause failure.
7. Select one of the following:
   - If you want to continue defining behavior options, click **Add**, select the desired behavior option, and continue the connector section definition. For instructions on defining other behavior options, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior option, select the option from the **Behavior Options** list. For instructions on editing behavior options, see *Using the connector section editors*.

- If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

• *Connector Behavior*
• *Connector section editors*
• *Creating connector sections*

Defining a reference length

You can define the translational or angular positions at which constitutive forces and moments are zero for the available components of relative motion. Reference lengths/angles are subtracted from the components of relative motion before being used in elasticity and friction behavior. You can define only one reference length option. For more information, see *Connector Behavior*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new reference length option, click **Add** and select **Reference Length** from the option menu that appears.
   - To edit an existing reference length option, select the option from the **Behavior Options** list to display the associated data fields for that option.
3. In the **Constitutive Reference Length and Angle** portion of the editor, enter the reference lengths and reference angles associated with the available components of relative motion. If you do not enter a value, the reference lengths and reference angles are determined by the initial configuration of the connector.
4. Select one of the following:
   - If you want to continue defining behavior options, click **Add**, select the desired behavior option, and continue the connector section definition. For instructions on defining other behavior options, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior option, select the option from the **Behavior Options** list. For instructions on editing behavior options, see *Using the connector section editors*.
   - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector Behavior*
- *Connector section editors*
- *Creating connector sections*

Defining time integration

You can specify implicit or explicit time integration for elasticity, damping, and friction in an Abaqus/Explicit analysis. By default, implicit time integration is used. You can define only one integration option. For more information, see *Connector Behavior*.

1. Display the connector section editor using one of the following methods:
   - To create a new connector section, follow the procedure outlined in *Creating connector sections*.
   - To edit an existing connector section, select **Connector->Section->Manager** from the main menu bar, select the connector section from the list that appears, and click **Edit**.
2. In the **Edit Connector Section** dialog box, do one of the following:
   - To define a new integration option, click **Add** and select **Integration** from the option menu that appears.
   - To edit an existing integration option, select the option from the **Behavior Options** list to display the associated data fields for that option.
3. Click the arrow to the right of the **Integration** field and select an option from the list that appears:
   - Select **Implicit** to use implicit time integration.
   - Select **Explicit** to use explicit time integration.
4. Select one of the following:
   - If you want to continue defining behavior options, click **Add**, select the desired behavior option, and continue the connector section definition. For instructions on defining other behavior options, see *Using the connector section editors*.
   - If you want to view or edit an existing behavior option, select the option from the **Behavior Options** list. For instructions on editing behavior options, see *Using the connector section editors*.
   - If you want to save your connector section definition and exit the editor, click **OK.**

Additional information

- *Connector section editors*
- *Creating connector sections*

Specifying behavior settings for tabular data

You can specify behavior settings for the regularization (Abaqus/Explicit analyses only) and the extrapolation of tabular data that you use to define elasticity, damping, user-defined friction, plasticity, and damage behavior options and connector derived component terms. You can also specify settings for the evaluation of rate-dependent data in an Abaqus/Explicit analysis for selected behavior options. For more information, see *Defining Connector Behavior Using Tabular Data*.

You can modify the behavior settings for all of the behavior options in a connector section, or you can specify the behavior settings for individual behavior options. Behavior settings for individual behavior options take precedence over the connector section behavior settings. You specify behavior settings for all of the behavior options in a connector section on the **Table Options** tabbed page of the connector section editor. For more information, see *Creating connector sections*.

To display context-sensitive help for options in the **Edit Table Options** dialog box, you must select the option of interest and then press **[F1]**. (The **Help** menu in the main menu bar is unavailable while the dialog box is displayed.)

1. Display the **Edit Table Options** dialog box.
   - For elasticity, damping, user-defined friction, plasticity, or damage behavior using tabular data, click **Table Options** on the **Behavior Options** tabbed page of the connector section editor. For more information, see the following sections:
     - *Defining elasticity*
     - *Defining damping*
     - *Specifying predefined friction parameters or contact forces*
     - *Defining plasticity*
     - *Defining damage*
     - *Defining damage evolution*
   - For connector derived component terms, click **Table Options** from the **Edit Derived Component Term** dialog box. For more information, see *Specifying connector derived components*.
2. In the **Regularization** portion of the dialog box, specify the settings for the regularization of data in an Abaqus/Explicit analysis.
   a. Specify which regularization behavior settings to use for this behavior option.
   - Toggle on **Use behavior settings** to use the regularization behavior settings that apply to all of the behavior options in the connector section.
   - Toggle off **Use behavior settings** to use the regularization behavior settings that you specify in this dialog box.

   b. By default, Abaqus/Explicit regularizes the data into tables that are defined in terms of even intervals of the independent variables.
   - Toggle on **Regularize data** to regularize tabular data.
   - Toggle off **Regularize data** to turn off regularization of the tabular data and use the data that you define directly.

   c. If you want to regularize tabular data, specify the error tolerance.
   - Choose **Use default** to use the default value of 0.03.
   - Choose **Specify**, and enter a value for the error tolerance.

3. The **Rate Options** portion of the dialog box is available if you are defining plasticity behavior that includes isotropic hardening or damage initiation behavior. You can specify settings for the evaluation of rate-dependent data in an Abaqus/Explicit analysis. These settings apply only for plasticity behavior that includes isotropic hardening using a **Tabular** definition (the force-constitutive motion data are specified directly in tabular form) or damage behavior with plastic motion–based initiation criteria.
   a. Specify the value of the rate filter factor, ω.
      - Choose **Use default** to use the default value of 0.9.
      - Choose **Specify**, and enter a value ($0 < \omega \leq 1$). A value of $\omega = 1$ provides no filtering and should be used with caution.
   b. Specify the interpolation method.
      - Choose **Linear** to use linear intervals for the equivalent relative plastic motion rate while interpolating rate-dependent damage initiation data.
      - Choose **Logarithmic** to use logarithmic intervals for the equivalent relative plastic motion rate while interpolating rate-dependent damage initiation data.
4. In the **Extrapolation** portion of the dialog box, specify the settings for the extrapolation of data.
   a. Specify which extrapolation behavior settings to use for this behavior option.
      - Toggle on **Use behavior settings** to use the extrapolation behavior settings that apply to all of the behavior options in the connector section.
      - Toggle off **Use behavior settings** to use the extrapolation behavior settings that you specify in this dialog box.
   b. Specify the method for the extrapolation of tabular data. The data points that you enter make up a nonlinear curve in the constitutive space. By default, Abaqus extrapolates the dependent variables as constant values that correspond to the end points of the curve outside the specified range of the independent variables.
      - Choose **Constant** to use constant extrapolation of the dependent variables outside the specified range of the independent variables.
      - Choose **Linear** to use linear extrapolation of the dependent variables outside the specified range of the independent variables.
5. Click **OK** to save your behavior settings and to return to the connector section editor.

Additional information

- *Connector section editors*
- *Creating connector sections*

Specifying connector derived components

You can create a connector derived component to specify the friction-generating normal force in connectors as a complex combination of connector forces and moments or to use as an intermediate result in a connector potential function. For more information on connector derived components, see *Connector Functions for Coupled Behavior*.

To display context-sensitive help for options in the derived component editor or the **Edit Derived Component Term** dialog box, you must select the option of interest and then press **[F1]**. (The **Help** menu in the main menu bar is unavailable while the editor and dialog box are displayed.)

1. To display the connector derived component editor, click  from the connector section editor when you are defining a user-defined friction model or from potential contribution editors. For more information, see *Specifying predefined friction parameters or contact forces*, and *Specifying potential terms*.
2. In the **Edit Derived Component** dialog box you can add, edit, and delete derived component terms as follows:

Adding terms

Click  to display the **Edit Derived Component Term** dialog box.

Modifying terms

In the **Derived Component Terms** list, select the term whose data you want to change, and click  to display the **Edit Derived Component Term** dialog box.

Deleting terms

In the **Derived Component Terms** list, select the terms that you want to delete, and click .

3. In the **Edit Derived Component Term** dialog box, do the following:
   a. Select a **Term operator** to specify the method used to compute the derived component term.

Square root of sum of squares

Abaqus computes the derived component term as the square root of the sum of the squares of each intrinsic component contribution.

Direct sum

Abaqus computes the derived component term as the direct sum of the intrinsic component contributions.

Macauley sum

Abaqus computes the derived component term as the sum of the intrinsic component contributions with a Macauley bracket function applied to each contribution.

   b. Choose the **Overall term sign**.

Positive

Select a positive sign for the derived component term.

Negative

Select a negative sign for the derived component term.

c. Choose the intrinsic components to be used in the definition of the derived component, and enter the scaling factor (α_1) for each component in the data table.

d. If desired, toggle on **Use local directions**, choose **Independent position components** or **Independent constitutive motion components**, and select the available components of relative position or constitutive motion to be used as independent variables in the definition of the connector derived component. Enter the connector relative position or constitutive motion in the direction specified by the independent component in the corresponding field of the table. For more information, see *Defining Nonlinear Connector Behavior Properties to Depend on Relative Positions or Constitutive Displacements/Rotations*.

e. If desired, toggle on **Use temperature-dependent data** to define data that vary with temperature. A column labeled **Temp** appears in the data table.

f. If desired, click the arrows to the right of the **Number of field variables** field to increase or decrease the number of field variables to define data that depend on field variables. Field variable columns appear in the data table.

4. To modify the behavior settings for the regularization (Abaqus/Explicit analyses only) or the extrapolation of the data, use the procedure described in *Specifying behavior settings for tabular data*.

5. Click **OK** to save the data and to exit the editor.

Additional information

- *Understanding connector sections and functions*
- *Connector section editors*
- *Specifying predefined friction parameters or contact forces*
- *Specifying potential terms*

Specifying potential terms

Connector potentials are user-defined mathematical functions that represent yield surfaces, limiting surfaces, or magnitude measures in the space spanned by components of relative motion in a connector.
You use potentials to define coupled friction, plasticity, or damage behavior options. For more information, see *Connector Functions for Coupled Behavior*.

To display context-sensitive help for options in the potential contribution editor, you must select the option of interest and then press **[F1]**. (The **Help** menu in the main menu bar is unavailable while the editor is displayed.)

1. Display the potential tabbed page.
   - For friction behavior, use the procedure outlined in *Defining friction* to display the **Force Potential** tabbed page.
   - For plasticity behavior, use the procedure outlined in *Defining plasticity* to display the **Force Potential** tabbed page.
   - For damage behavior, use the procedures outlined in *Defining damage* and *Defining damage evolution* to display the **Initiation Potential** and **Evolution Potential** tabbed pages.
2. Choose the **Operator**.

Sum

Define the potential as the sum of the contributions.

Maximum

Define the potential as the contribution that yields the maximum value.

3. If you selected **Sum** in the **Operator** field, enter a positive value in the **Exponent** field to specify the inverse of the overall exponent in the potential definition, β, that defines a general elliptical form of the potential. The default value of `2` defines a quadratic form of the potential.
4. In the **Potential Contributions** portion of the tabbed page, select one of the following options:
   - To define a contribution to the potential, P_i, click .

     The potential contribution editor appears.
   - To edit an existing contribution, select a contribution from the list, and click .

     The potential contribution editor appears.
   - To delete an existing contribution, select a contribution from the list, and click .
5. In the potential contribution editor, do the following:
   a. Specify the component to use in the potential contribution.

Specify component

Choose an available component of relative motion to define the potential contribution.

Specify derived component

Click  to specify a connector derived component to define the potential contribution. For more information, see *Specifying connector derived components*.

b. Choose the **Sign**.

Positive

Select a positive sign for the potential contribution.

Negative

Select a negative sign for the potential contribution.

c. Enter the following data:

Scaling factor

Enter the nonzero scaling factor R. The default value is 1.

Positive exponent

Enter a value for the positive exponent α. The positive exponent value is ignored if you selected **Maximum** in the **Operator** field.

Shift factor

Enter the shift factor a. The default value is 0.

H-function

Select the function H to be used to generate the contribution. The default value is **ABS**.

- **ABS**. Use the absolute value function to generate the contribution.
- **MACAULEY**. Use the Macauley bracket ($\langle X \rangle = 0 \quad \text{if} \quad X \leq 0 \quad \text{and} = X \quad \text{if} \quad X > 0$) to generate the contribution.
- **IDENTITY**. Select the identity function only if the following is true:
  - **Positive exponent** (α) = **Exponent** (β) = 1

Additional information

- *Defining friction*
- *Defining plasticity*
- *Defining damage*
- *Specifying connector derived components*

Using the Query toolset to obtain connector assignment information

You can use the Query toolset to display detailed information about connector section assignments.

1. From the main menu bar, select **Tools->Query**.

    **Tip:** You can also query the model by clicking the  tool in the Query toolset.

   Abaqus/CAE displays the **Query** dialog box.

   You can request either a general query or a module-specific query. For a discussion of the information displayed by general queries, see *Obtaining general information about the model*. The **Connector assignments** query is specific to the Interaction module.

2. From the **Interaction Module Queries** list, select **Connector assignments**.

   A summary of connector section assignments appears.

3. Select the wires or attachment lines in the viewport that you want to query. (For more information, see *Selecting objects within the viewport*.)

4. Click **Done** or click mouse button 2 to display detailed information about the connector section assignment for the selected wires or attachment lines.

5. To exit the querying procedure, click the cancel button  in the prompt area.

Additional information

- *Creating and modifying connector section assignments*
- *Understanding the role of the Query toolset*

The Load module

You use the Load module to define and manage the following prescribed conditions:

- Loads
- Boundary conditions
- Predefined fields
- Load cases (see *Load cases*)

For information on modeling a bolt load, see *Bolt loads*.

In this section:

- *Understanding the role of the Load module*
- *Entering and exiting the Load module*
- *Managing prescribed conditions*
- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Transferring results between Abaqus analyses*
- *Using the Load module toolbox*
- *Using the Load module*
- *Using the load editors*
- *Using the boundary condition editors*
- *Using the predefined field editors*

Understanding the role of the Load module

Prescribed conditions in Abaqus/CAE are step-dependent objects, which means that you must specify the analysis steps in which they are active. You can use the load, boundary condition, and predefined field managers to view and manipulate the stepwise history of prescribed conditions. You can also use the **Step** list located in the context bar to specify the steps in which new loads, boundary conditions, and predefined fields become active by default.

You can use the Amplitude toolset in the Load module to specify complicated time or frequency dependencies that can be applied to prescribed conditions. The Set and Surface toolsets in the Load module allow you to define and name regions of your model to which you would like to apply prescribed conditions. The Analytical Field toolset and the Discrete Field toolset allow you to create fields that you can use to define spatially varying parameters for selected prescribed conditions.

Load cases are sets of loads and boundary conditions used to define a particular loading condition. You can create load cases in static perturbation and steady-state dynamic, direct steps. For information on load cases, see *Load cases*.

Additional information

- *About Prescribed Conditions*
- *Multiple Load Case Analysis*
- *The Amplitude toolset*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *The Set and Surface toolsets*

Entering and exiting the Load module

You can enter the Load module at any time during an Abaqus/CAE session by clicking **Load** in the **Module** list located in the context bar. The **Load**, **BC**, **Predefined Field**, **Load Case**, **Feature**, and **Tools** menus appear on the main menu bar. A **Step** list appears in the context bar.

To exit the Load module, specify another module in the **Module** list in the context bar. You need not take any specific action to save your prescribed conditions before exiting the module; they are saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

Managing prescribed conditions

Prescribed condition managers are dialog boxes that you use to organize and manipulate the prescribed conditions associated with a given model. Each kind of prescribed condition that you can define in the Load module has a separate manager. You access the managers by selecting **Manager** from the appropriate menus on the main menu bar. The Load module provides the following managers:

- **Load Manager**
- **Boundary Condition Manager**
- **Predefined Field Manager**

Prescribed condition managers contain alphabetical lists of all the prescribed conditions of a certain type that you have created. For example, the **Load Manager** shown in *Figure 1* contains a list of loads.

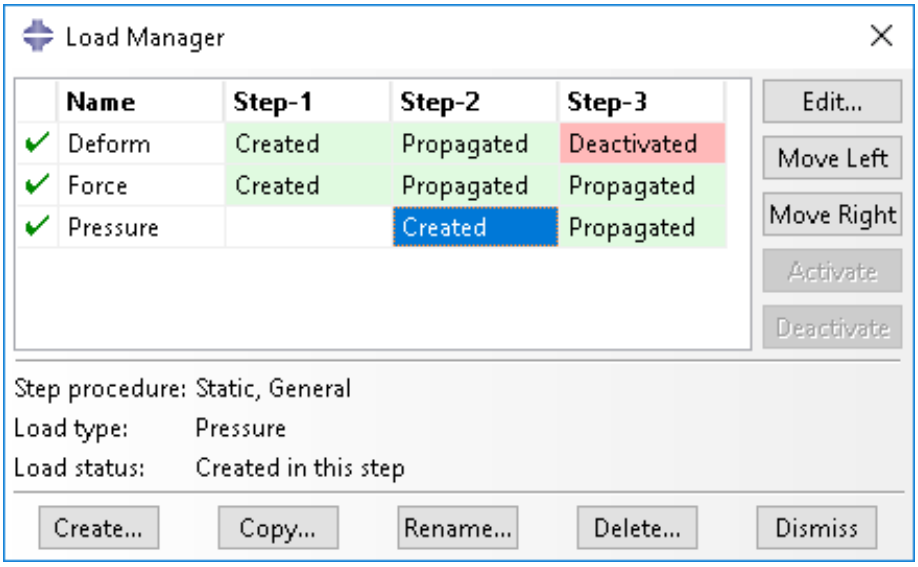


Figure 1: The Load Manager.

The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the managers allow you to create new prescribed conditions or to edit, copy, rename, and delete existing ones. You can also initiate the create, edit, copy, rename, and delete procedures by using the **Load**, **BC**, and **Predefined Field** menus in the main menu bar. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

The prescribed condition managers are step-dependent managers, which means that they contain additional information concerning the history of each load, boundary condition, and predefined field in the model. You can use the icons in the column along the left side of the managers to suppress prescribed conditions or to resume previously suppressed prescribed conditions for an analysis. The suppress and resume procedures are also available from the **Load**, **BC**, and **Predefined Field** menus in the main menu bar. For more information, see *Suppressing and resuming objects*.

You can use the **Copy** button in manager dialog boxes, the corresponding menu command, or the Model Tree to copy a load, boundary condition, or predefined field. You can copy these objects from any step to any valid step, with some restrictions. For more details, see *Copying step-dependent objects using manager dialog boxes*.

The **Move Left**, **Move Right**, **Activate**, and **Deactivate** buttons allow you to manipulate the stepwise history of prescribed conditions. For more information, see *Modifying the history of a step-dependent object*.

Note:

The **Activate** and **Deactivate** buttons are not available in the **Predefined Field Manager**.

For detailed instructions on creating, editing, and manipulating prescribed conditions, see the following sections:

- *Using the Load module*
- *Using the load editors*
- *Using the boundary condition editors*
- *Using the predefined field editors*

Additional information

- *Managing objects*
- *What are step-dependent managers?*
- *Changing the status of an object in a step*

Creating and modifying prescribed conditions

To create a load, boundary condition, or predefined field, select **Create** from the appropriate menu in the main menu bar. A **Create** dialog box will appear in which you can provide a name for the prescribed condition and choose the type of the prescribed condition that you want to create.

When you click **Continue** in the **Create** dialog box, you are prompted to select the region to which you want to apply the prescribed condition, unless the prescribed condition is applied to the whole model. You can apply connector loads and connector boundary conditions (displacement, velocity, and acceleration) only to wires that are associated with a connector section assignment. If you are select multiple wires, the connector sections assigned to the wires in the connector section assignments must have the available components of relative motion for which you are defining loads and boundary conditions. You can apply connector material flow boundary conditions only to endpoints of wires that are associated with a connector section assignment. Once you have selected the region, an editor appears in which you can specify additional information about the prescribed condition, such as its magnitude.

The top panel of each prescribed condition editor displays the name and type of the prescribed condition, the analysis step you are currently in, and the region of the model to which the prescribed condition will be applied. If you are editing a prescribed condition in the step in which it was first created, an **Edit Region** ( ) button appears next to the **Region** field; this button allows you to edit the region to which the prescribed condition is applied. If editing the region requires a complete redefinition of the prescribed condition (for example, if the prescribed condition is applied to the whole model or refers to subregions within the originally selected region), the **Edit Region** button does not appear. For more information, see *Editing the region to which a prescribed condition is applied*.

The format of the rest of the editor depends on the type of prescribed condition you are defining and on the step specified at the top of the editor. For example, the editor for concentrated forces is shown in *Figure 1*.

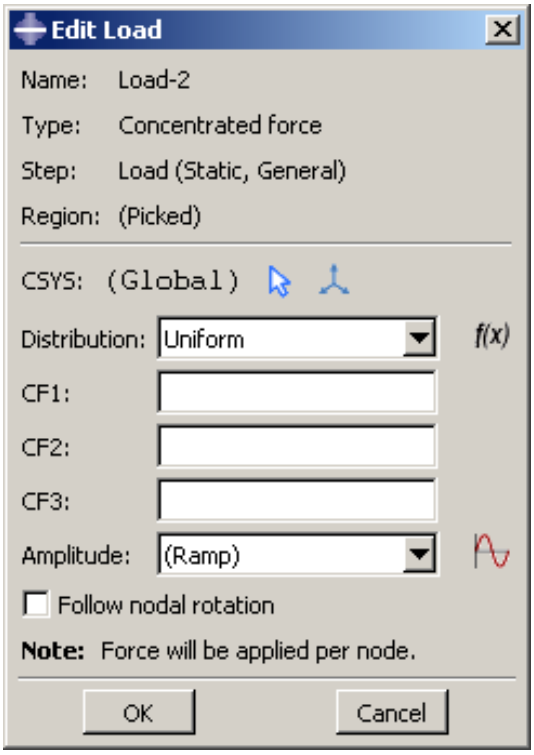


Figure 1: The editor for concentrated forces.

This editor contains special text fields in which you can specify the components of the force in the 1-, 2-, and 3-directions. The editor also contains an **Amplitude** text field that allows you to vary the magnitude of the prescribed condition as

a function of time. You can accept the default amplitude, select an amplitude that you have defined using the Amplitude toolset, or click  to define a new amplitude. (For more information, see *The Amplitude toolset*.)

You can specify the coordinate system in which you will apply the following loads or boundary conditions:

Loads

- Concentrated force
- Moment
- General and shear surface traction
- General shell edge load
- Inertia relief
- Current density

Boundary conditions

- Symmetry/antisymmetry/encastre
- Displacement/rotation
- Velocity/angular velocity
- Acceleration/rotational acceleration
- Eulerian mesh motion
- Magnetic vector potential

All other prescribed conditions use the global coordinate system, with the exception of pressures, which are applied normal to the selected surfaces.

If the load or boundary condition allows you to specify the coordinate system, you can select an existing datum coordinate system or you can accept the global coordinate system. If the desired datum coordinate system does not exist, you can create it using the Datum toolset. (For more information, see *Creating datum coordinate systems*.) Alternatively, you can refer to an Abaqus/Standard user subroutine that defines the coordinate system (see *ORIENT*).



Note:

If you delete or suppress the datum coordinate system, the orientation of the load or boundary condition reverts to the global coordinate system.

The rules for creating and modifying predefined fields vary depending on the predefined field type:

- Some predefined fields require that you specify only the initial conditions. You can create and edit this type of predefined field only in the initial step. Abaqus computes subsequent values for the predefined field as the analysis progresses. The predefined fields of this type are initial velocity specifications, hardening specifications, and material assignments (for Eulerian analyses). For more information, see *Initial Conditions*.
- You can create predefined temperature fields for any step in the analysis. You can define the temperatures for the current model either by entering the values for the desired steps or by reading the temperature values computed by Abaqus in a previous analysis with thermal components. For more information, see “Temperature,” in *Predefined Fields*.



Note:

If you do not define initial values for a predefined field, that field is assumed to have a value of zero at the start of the analysis.

Once you have created a prescribed condition, you can modify the prescribed condition in the following ways:

- You can modify some or all of the data that you entered in the editor when you created the prescribed condition.
- You can use the managers to modify the stepwise history of the prescribed condition. (For more information, see *What are step-dependent managers?*.)

To display help on a particular manager or editor feature, select **Help->On Context** from the main menu bar and then click the feature of interest.

Additional information

- *What are step-dependent managers?*
- *Using the load editors*
- *Using the boundary condition editors*
- *Using the predefined field editors*
- *The Datum toolset*
- *The Amplitude toolset*

Understanding symbols that represent prescribed conditions

This section explains how to interpret the symbols that represent prescribed conditions.

When you apply prescribed conditions to a region, you can choose to display symbols in the viewport that indicate the following:

- The regions to which you applied the prescribed condition.
- The type of the prescribed condition.
- If applicable, the degrees of freedom to which you applied the prescribed condition.
- If applicable, the direction (negative or positive) in which you applied the prescribed condition.
- If applicable, the spatial variation of the prescribed condition.

For information about controlling the visibility of these symbols, see *Controlling the display of attributes*.

In this section:

- *Understanding prescribed condition symbol type, color, and size*
- *What do single-headed and double-headed arrows represent?*
- *Understanding symbol location and direction*

Understanding prescribed condition symbol type, color, and size

The type, color, and size of the symbols that represent prescribed conditions can vary with

- the type of prescribed condition that the symbols represent,
- the degrees of freedom to which you apply the prescribed condition, and
- the spatial variation of the prescribed condition (for analytical field distributions).

Refer to *Symbols used to represent prescribed conditions*, for summaries of the significance of the symbol types and colors.

For example, *Figure 1* displays a concentrated force applied to vertices. All of the arrows that represent the different components of the concentrated force are yellow.

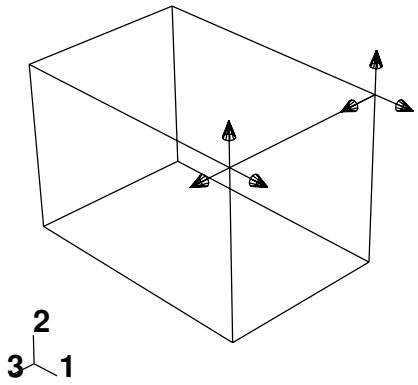


Figure 1: A concentrated force.

On the other hand, *Figure 2* shows a **Velocity/Angular Velocity** boundary condition that is applied to both translational and rotational degrees of freedom. The sandy brown arrows represent components of the boundary condition that are applied to translational degrees of freedom. The magenta arrows represent components of the boundary condition that are applied to rotational degrees of freedom.

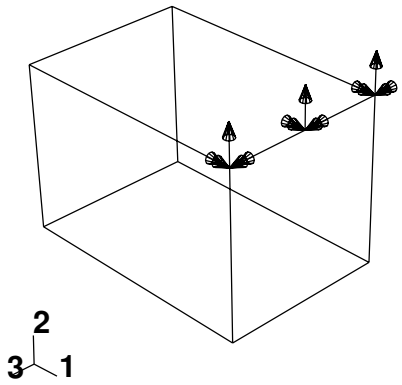


Figure 2: A boundary condition applied to an edge.

Note:

When a boundary condition fixes a degree of freedom in place, the arrow representing that component lacks a stem.

Figure 3 displays a uniform temperature field applied to a face.

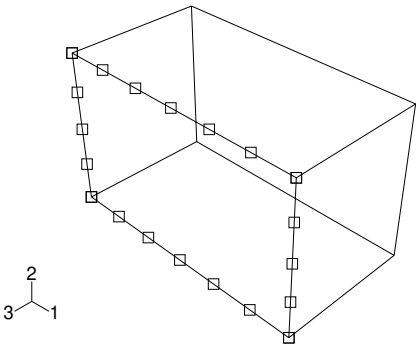


Figure 3: A uniform temperature field.

In general, the size of the symbols is uniform and unrelated to the magnitude of the prescribed condition. For prescribed conditions that use analytical field distributions, the symbols are scaled based on the analytical field value. *Figure 4* shows a pressure load that uses an analytical field to specify a spatially varying magnitude.

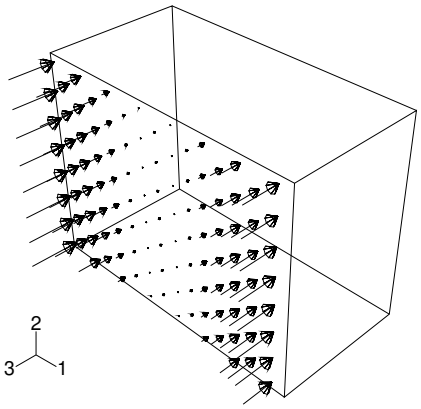


Figure 4: Variation of a pressure load over a face.

In addition, for symbols other than arrows, a plus sign (+) or a minus sign (−) is displayed inside each symbol to indicate whether the magnitude of the prescribed condition is positive or negative at that location.*Figure 5* shows a temperature boundary condition. For clarity, the symbol size has been increased.

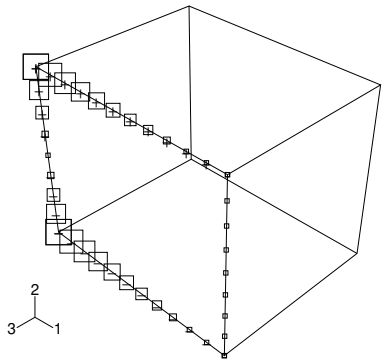


Figure 5: A temperature boundary condition using an analytical field distribution.

For information on controlling symbol size and scaling, see *Controlling the display of attributes*.

In some circumstances Abaqus/CAE displays scaled-down symbols for prescribed conditions, such as when a specified prescribed condition has no effect on the analysis or when an analytical field evaluates to zero for a portion of its region. These scaled-down symbols are noticeably smaller than the default symbol size. For example, if you specify a shear surface traction load with a direction vector normal to the surface, Abaqus/CAE cannot apply this type of load normal to the reference surface and displays very small arrow symbols to represent the load in the viewport.

Additional information

- *Displaying symbols for interactions and prescribed conditions that use analytical fields*
- *Controlling the display of attributes*
- *Symbols used to represent prescribed conditions*

What do single-headed and double-headed arrows represent?

In many cases Abaqus/CAE uses arrows to represent prescribed conditions in the viewport. These arrows represent each component of the prescribed condition (except for fluid boundary conditions, in which case the arrows represent the resultant direction). For example, the arrows that appear in *Figure 1* represent the three components of a concentrated force that is applied to two vertices.

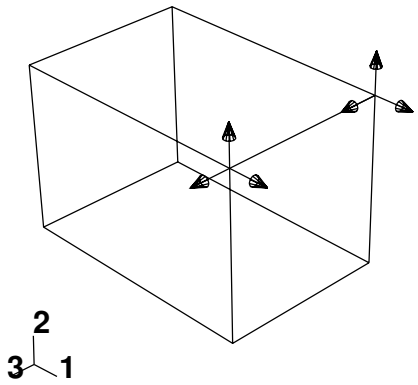


Figure 1: A concentrated force with three components.

An arrow with a single arrowhead represents a component of a prescribed condition that is applied to a translational degree of freedom. For example, the three components of the concentrated force in *Figure 1* are applied to degrees of freedom 1 through 3; therefore, each arrow in the figure has a single arrowhead.

When a component of a prescribed condition is applied to a rotational degree of freedom, that component appears as a double-headed arrow. The arrows in *Figure 2* indicate that a **Velocity/Angular Velocity** boundary condition is applied to degrees of freedom 4 and 6 of the vertices.

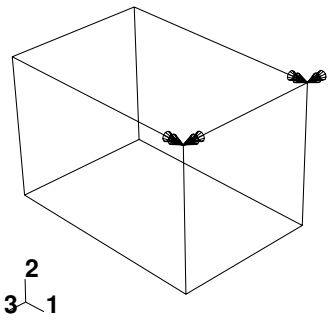


Figure 2: A boundary condition applied to rotational degrees of freedom.

A magnified view of the double-headed arrows appears in *Figure 3*.

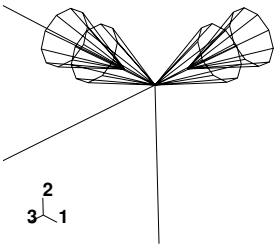


Figure 3: Magnified double-headed arrows.

If you apply a prescribed condition to both translational and rotational degrees of freedom, both the single-headed and the double-headed arrows appear. For example, a **Velocity/Angular Velocity** boundary condition is applied to degrees of freedom 1, 3, 4, and 6 of the vertex in *Figure 4*.

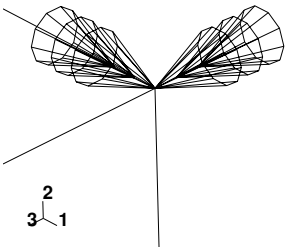


Figure 4: Magnified view of a boundary condition applied to translational and rotational degrees of freedom.

In this figure the single-headed arrows are sandy brown and indicate that degrees of freedom 1 and 3 of the vertex are fixed.The double-headed arrows are magenta and appear directly behind the single-headed arrows; the double-headed arrows indicate that degrees of freedom 4 and 6 of the vertex are fixed.

For information on arrow color, see *Understanding prescribed condition symbol type, color, and size*. For information on when to expect arrows to point toward or away from a region, see *Understanding symbol location and direction*.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Controlling the display of attributes*

Understanding symbol location and direction

The placement of symbols on a model can depend on the type of prescribed condition that the symbols represent and the type of region to which the prescribed condition is applied. *Table 1* indicates where symbols appear on geometric models, and *Table 2* indicates where symbols appear on meshed models.

Table 1: Symbol location on geometry.

Region type to which the prescribed condition is applied	Location of symbols on the model
Vertex	At the vertex
Edge	Equally spaced along the edge
Assembly-level wire	At the midpoint of the wire
Face	Equally spaced over the interior of the face for directional prescribed conditions (e.g., pressure load)
	Equally spaced along the edges of the face for nondirectional prescribed conditions (e.g., surface charge and boundary conditions)
Cell	Equally spaced along each edge of the cell
Whole model	At the point required to define the rigid body motion (inertia relief load only); otherwise, at the triad indicating the origin and orientation of the global coordinate system

Table 2: Symbol location on meshes.

Region type to which the prescribed condition is applied	Location of symbols on the model
Node	At the node
Element edge (for two-dimensional meshes)	At the midpoint of the element edge
Element face (for three-dimensional meshes)	At the centroid of the element face
Assembly-level wire	At the midpoint of the wire

For example, *Figure 1* shows a concentrated force applied to two vertices and a boundary condition applied to a surface of a geometric model.

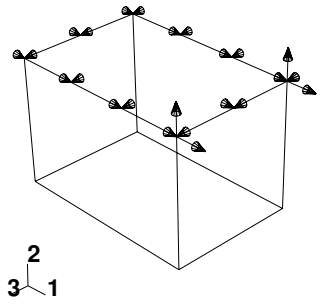


Figure 1: A concentrated force and a boundary condition.

Figure 2 shows a boundary condition applied to four nodes and a pressure load applied to several element faces of a mesh.

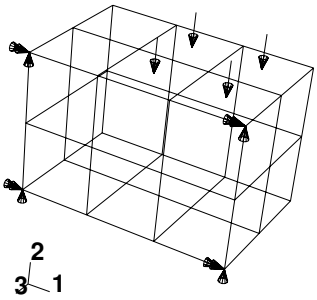


Figure 2: A pressure load and a boundary condition.

Note:

If you apply a pressure load to a planar geometry face where the surface area is small compared to the enclosed area (such as a ring formed by two concentric circles), the load symbols may not be distributed evenly, regardless of the symbol density settings in the **Assembly Display Options** dialog box.

When a boundary condition fixes a degree of freedom in place, the arrow representing that component points into the region and lacks a stem. For example, the boundary condition in *Figure 3* fixes degrees of freedom 1, 2, and 3 in place.

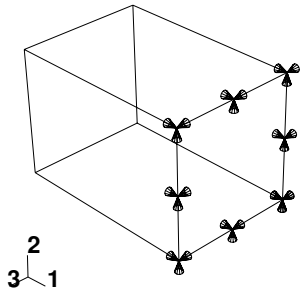


Figure 3: A boundary condition fixing degrees of freedom in place.

Likewise, if a positive pressure load or an Eulerian inflow boundary condition is applied to a region, the arrows representing that pressure load or boundary condition point into the region, as illustrated in *Figure 4*.

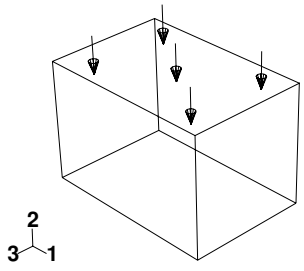


Figure 4: A positive pressure load.

If a load is defined to have a complex magnitude and the real and imaginary parts have different signs (for example, $2 - 3i$), the load will appear as an arrow with two ends. Similarly, an Eulerian boundary condition that includes both inflow and outflow components will appear as an arrow with two ends.

In all other cases, arrows representing components of a prescribed condition point out from the region.



Note:

When a component of a concentrated force is zero, no arrow appears for that component. Likewise, when a boundary condition leaves a degree of freedom unconstrained, no arrow appears for that component.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Controlling the display of attributes*

Transferring results between Abaqus analyses

You can select part instances from your model and associate an initial state field with the instances. An initial state field applies a deformed mesh and its associated material state to the instances using data imported from a previous Abaqus/Standard or Abaqus/Explicit analysis. Abaqus/CAE allows you to select the job name corresponding to the analysis from which the initial state field is imported. You can also specify the particular step and increment of the analysis from which to import data. Abaqus/CAE imports data from several of the files created by the previous analysis. As a result, the files from the analysis must reside in the directory from which you started the current Abaqus/CAE session.

You can use this capability to drive an Abaqus/Explicit analysis with the results of an Abaqus/Standard analysis and vice versa. This is useful if your problem can be broken down into different stages; for example, you can use Abaqus/Explicit to analyze a metal forming problem and Abaqus/Standard to analyze the following springback. You can also use this capability to change the model definition between steps. For more information, see *About Transferring Results between Abaqus Analyses*.

You can also transfer results and model information from an Abaqus/Standard analysis to a new Abaqus/Standard analysis, where you can specify additional model definitions before continuing the analysis. For example, you might first study the local behavior of a particular component during an assembly process and then study the behavior of the assembled product. You can start by analyzing the local behavior of the component in an Abaqus/Standard analysis. You can then transfer the model information and results from this analysis to a second Abaqus/Standard analysis, where you can specify additional model definitions for the other components and analyze the behavior of the entire product.

Abaqus/CAE always imports the material state along with the deformed mesh. If you want to import only the deformed mesh, you can import a mesh from a selected step and increment of an output database. For more information, see *What kinds of files can be imported and exported from Abaqus/CAE?*.

Abaqus uses the imported information when you submit a job for analysis; however, Abaqus/CAE does not update the shape of the selected instances to reflect the applied deformed mesh. As a result, you should be careful when adding new instances to the assembly and positioning them relative to existing part instances. For example, a new part instance may appear to touch one of the instances associated with the initial state field; however, when the analysis applies the imported deformed mesh, the instances may become separated or overclosed.

To avoid this mismatch between the undeformed state and the imported state, you may want to import the deformed mesh from the analysis instead of working with the undeformed part instance. Even if you import the deformed mesh, you must take care that the frame from which you imported the mesh is the same as the step and increment specified in the initial state field. For more information, see *Importing a part from an output database*. Alternatively, you can create the current model by copying it from the model that generated the previous Abaqus/Standard or Abaqus/Explicit analysis. For more information, see *Manipulating models within a model database*.

The reference configuration is the configuration of the model from which displacements (and associated strains) are calculated. By default, Abaqus/CAE does not use the imported data to update the reference configuration. As a result, displacements and strains are calculated as total values relative to the reference configuration at the start of the original analysis, and the values will be continuous between analyses. You can change the default behavior and configure Abaqus/CAE to update the reference configuration to be the imported configuration. Abaqus/CAE now calculates displacements and strains relative to the new imported reference configuration; for example, for a springback analysis.

Abaqus imposes many restrictions when you try to create an initial state field. For a detailed discussion of these limitations, see *About Transferring Results between Abaqus Analyses*. For example, the mesh of the part instances that you select from the current model must match the mesh of the part instances that you are importing. You can then, for example, change the material definition, add loads and boundary conditions, and change from an Abaqus/Standard to an Abaqus/Explicit step. However, you cannot perform an operation that will change the mesh of a selected part instance; for example, you cannot partition the part instance.

You can transfer results between analyses only if the original analysis used one of the following steps:

- Static stress

- Dynamic stress
- Steady-state transport

In addition, if you are importing data from one Abaqus/Standard analysis to another, the original analysis can use a coupled temperature-displacement step. You cannot import data from a linear perturbation step.

In addition, Abaqus/CAE applies the following limitations:

- The selected part instances and the instances from the previous analysis must have the same name.
- After you define the initial state field, Abaqus/CAE will continue to show the undeformed shape of the model.
- You cannot use the Assembly module position and constraint tools, such as **Translate** and **Face to Face**, to move a part instance associated with an initial field.
- Abaqus/CAE imports only the mesh and the material state from the previous analysis. As a result, you must redefine sets, surfaces, and all of the prescribed conditions (loads, boundary conditions, predefined fields, interactions, connectors, etc.) at the assembly level of the current model. You should not redefine any of these components in the part definitions of the current model.
- Abaqus/CAE checks that the files exist that contain data from the previous Abaqus/Standard or Abaqus/Explicit analysis; however, it does not check that the specified step and increment number have been written to the files. The job submission fails if the data for the specified step or increment do not exist.
- You cannot modify a part instance associated with an initial field (or the part from which you created the instance). In addition, you cannot modify the mesh of a part instance associated with an initial field (or the mesh of the part from which you created the instance).
- You cannot assign new sections, material orientations, normals, or beam orientations to the part from which you created the instance associated with an initial field. Similarly, you cannot assign mass or inertia. However, you can edit the material definition (which Abaqus/CAE imports along with the mesh). The imported material definitions will overwrite any existing material definitions.

Using the Load module toolbox

You can access all the Load module tools through either the main menu bar or the Load module toolbox. *Figure 1* shows the icons for all the load tools in the Load module toolbox.

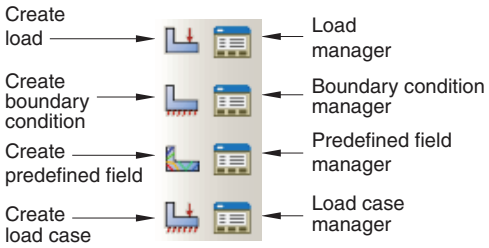


Figure 1: The Load module toolbox.

Using the Load module

This section provides general information on defining loads, boundary conditions, and predefined fields.

For information on other Load module topics, see the following sections:

- *What are step-dependent managers?*
- *Bolt loads*
- *Load cases*

In this section:

- *Creating loads*
- *Creating boundary conditions*
- *Creating predefined fields*
- *Editing the region to which a prescribed condition is applied*

Creating loads

When you create a load, you must specify the name of the load, the step in which to activate the load, the type of load, and the region of the assembly to which you want to apply the load.

1. From the main menu bar, select **Load->Create**.

   A **Create Load** dialog box appears with a default name displayed in the **Name** text field.

    **Tip:** You can also create a load using the  tool in the Load module toolbox.

2. Type a name for the load. For more information on naming objects, see *Using basic dialog box components*.
3. Select the step in which to activate the load. Click the arrow next to the **Step** text field, and select from the list that appears. Loads can be created only in an analysis step; you cannot create a load in the initial step.
4. From the **Category** list on the left side of the dialog box, choose the desired category. The **Category** choices available are dependent upon the type of analysis procedures you are performing.

   The **Types for Selected Step** list on the right side of the dialog box changes to a list of all the available load types.
5. From the **Types for Selected Step** list, select the load type and click **Continue**.
6. If you are creating a gravity load or an inertia relief load, the load editor appears.
7. If you are creating a connector force or connector moment using assembled fasteners, you can click **Done** in the prompt area to select a wire set from the template model.

   The load editor appears.

   a. Click the arrow next to the **Assembled fastener** field, and select from the list that appears.

      The template model name associated with the assembled fastener is displayed in the editor. The **Template set** list is populated with the wire sets that are associated with the referenced template model.

   b. Select a wire set from the **Template set** list. You must ensure that the wire set has a section assignment that has the available components of relative motion for which you want to define forces.

      The appropriate fields for the available components of relative motion are displayed.

8. For all other load types, select the region to which you want to apply the load.

   If you are creating a connector force or connector moment, you must select wires that are associated with a connector section assignment. The best approach for selecting wires is to use the default geometry set name for the wire feature (see *Creating or modifying wire features for multiple connectors*, for more information). If you select multiple wires, you must ensure that the connector sections assigned to the wires in the connector section assignments have the available components of relative motion for which you want to define forces or moments. If there are insufficient available components of relative motion for the connector force or connector moment, a message appears asking you to select different wires or to change the connection type.

   Use one of the following methods to select the region for the load:

   - Select a region in the viewport. You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.



Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

If the model contains a combination of mesh and geometry, click one of the following from the prompt area:

- Click **Geometry** to apply the load to geometry or to a reference point.
- Click **Mesh** to apply the load to a native or orphan mesh selection.

By default, for most load types a set or surface is created that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.

- To select from a list of existing sets or surfaces, do the following:

  1. Click **Sets** or **Surfaces** on the right side of the prompt area. (The name of the button depends on the type of object you are creating. For example, if you are creating a pressure load, a **Surfaces** button appears.)

     Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets or surfaces.

  2. Select the set or surface of interest and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

The load editor appears. The region to which you are applying the load is highlighted in the viewport.

9. Enter all of the data necessary to define the load and click **OK**.

   Note:

   If you create a connector force or connector moment that exceeds the failure criteria for a connector, the connector force or connector moment will still be applied.

   For detailed information on a particular feature of the editor, select **Help->On Context** from the main menu bar and then click the feature of interest or see *Using the load editors*.

   Symbols appear in the viewport that represent the load that you just created. For more information, see *Understanding symbols that represent prescribed conditions*.

Additional information

- *Understanding and using toolboxes and toolbars*
- *What are step-dependent managers?*
- *Selecting objects within the viewport*
- *Using the load editors*
- *Connectors*
- *About assembled fasteners*
- *Creating assembled fasteners*

- *The Set and Surface toolsets*

Creating boundary conditions

When you create a boundary condition, you must specify the name of the boundary condition, the step in which to activate the boundary condition, the type of boundary condition, and the region of the assembly to which you want to apply the boundary condition.

1. From the main menu bar, select **BC->Create**.

   A **Create Boundary Condition** dialog box appears with a default name displayed in the **Name** text field.

   Tip: You can also create a boundary condition using the  tool in the Load module toolbox.

2. Type a name for the boundary condition. For more information on naming objects, see *Using basic dialog box components*.
3. Select the step in which to activate the boundary condition. Click the arrow next to the **Step** text field, and select from the list that appears.
4. From the **Category** list on the left side of the dialog box, choose the desired category. The **Category** choices available are dependent upon the type of analysis procedures you are performing.

   The **Types for Selected Step** list on the right side of the dialog box changes to a list of all the available boundary condition types.
5. From the **Types for Selected Step** list, select the boundary condition type and click **Continue**.
6. If you are creating a connector boundary condition using assembled fasteners, you can click **Done** in the prompt area to select a wire set from the template model.

   The boundary condition editor appears.

   a. Click the arrow next to the **Assembled fastener** field, and select from the list that appears.

      The template model name associated with the assembled fastener is displayed in the editor. The **Template set** list is populated with the wire sets that are associated with the referenced template model.

   b. Select a wire set from the **Template set** list. You must ensure that the wire set has a section assignment that has the available components of relative motion for which you want to define velocity.

      The appropriate fields for the available components of relative motion are displayed.

7. For all other boundary condition types, select the region to which you want to apply the boundary condition.

   If you are creating a connector displacement, connector velocity, or connector acceleration boundary condition, you must select wires that are associated with a connector section assignment. The best approach for selecting wires is to use the default geometry set name for the wire feature (see *Creating or modifying wire features for multiple connectors*, for more information). If you select multiple wires, you must ensure that the connector sections assigned to the wires in the connector section assignments have the available components of relative motion for which you want to define displacement, velocity, or acceleration. If there are insufficient available components of relative motion for the connector boundary condition, a message appears asking you to select different wires or to change the connection type.

   If you are creating a connector material flow boundary condition, you must select endpoints of wires that are associated with a connector section assignment.

If you are creating an Eulerian mesh motion boundary condition, select an Eulerian part instance in the viewport. Otherwise, use one of the following methods to select the region for the boundary condition:

- Select a region in the viewport. You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.

  Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

  If the model contains a combination of mesh and geometry, you must choose the type of region to which you want to apply the boundary condition. From the prompt area, select one of the following:

  - Click **Geometry** to apply the boundary condition to geometry or to a reference point.
  - Click **Mesh** to apply the boundary condition to a native or orphan mesh selection.

  By default, a set or surface is created that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.

- To select from a list of existing sets or surfaces, do the following:

  1. Click **Sets** or **Surfaces** on the right side of the prompt area. (The name of the button depends on the type of object you are creating. For example, if you are creating a pressure load, a **Surfaces** button appears.)

     Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets or surfaces.

  2. Select the set or surface of interest and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

The boundary condition editor appears. The region to which you are applying the boundary condition is highlighted in the viewport.

8. Enter all of the data necessary to define the boundary condition and click **OK**.

   Note:

   If you create a connector displacement boundary condition that exceeds the failure criteria for a connector, the connector displacement will be ignored.

For detailed information on a particular feature of the editor, select **Help->On Context** from the main menu bar and then click the feature of interest or see *Using the boundary condition editors*.

Symbols appear in the viewport that represent the boundary condition that you just created. For more information, see *Understanding symbols that represent prescribed conditions*.

Additional information

- *Understanding and using toolboxes and toolbars*
- *What are step-dependent managers?*
- *Selecting objects within the viewport*
- *Using the boundary condition editors*
- *Connectors*
- *About assembled fasteners*
- *Creating assembled fasteners*
- *The Set and Surface toolsets*

Creating predefined fields

When you create a predefined field, you must specify the name of the field, the step in which to activate the field, the type of field, and the region of the assembly to which you want to apply the field.

Note:

The process for creating temperature fields is described separately; see *Defining a temperature field*.

1. From the main menu bar, select **Predefined Field->Create**.

   A **Create Predefined Field** dialog box appears with a default name displayed in the **Name** text field.

   Tip: You can also create a predefined field using the tool in the Load module toolbox.

2. Type a name for the predefined field. For more information on naming objects, see *Using basic dialog box components*.
3. Select the step in which to activate the predefined field. Click the arrow next to the **Step** text field, and select from the list that appears.
4. From the **Category** list on the left side of the dialog box, choose the desired category. The **Category** choices available are dependent upon the type of analysis procedures you are performing.

   The **Types for Selected Step** list on the right side of the dialog box changes to a list of all the available predefined field types.
5. From the **Types for Selected Step** list, select the predefined field type and click **Continue**.
6. If the model contains a combination of mesh and geometry, you must choose the type of region to which you want to apply the predefined field. From the prompt area, select one of the following:

   - Click **Geometry** to apply the predefined field to geometry or to a reference point.
   - Click **Mesh** to apply the predefined field to a native or orphan mesh selection.

7. Select the region to which you want to apply the predefined field.

   If you are creating a material assignment field or an initial state field, use the mouse to select a part instance in the viewport. For all other predefined fields, select a region using one of the following methods:

   - Use the mouse to select a region in the viewport. You can use the angle method to select a group of faces or edges from geometry or a group of element faces or nodes from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.

     Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

     By default, a set is created that contains the selected objects. You can change this behavior by toggling off the option to create a set in the prompt area. A default name is provided in the prompt area, but you can enter a new name.
   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area.

Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

2. Select the set of interest and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

The predefined field editor appears. The region to which you are applying the predefined field is highlighted in the viewport.

8. Enter all of the data necessary to define the predefined field and click **OK**. For detailed information on a particular feature of the editor, select **Help->On Context** from the main menu bar and then click the feature of interest or see *Using the predefined field editors*.

   Symbols appear in the viewport that represent the predefined field that you just created. For more information, see *Understanding symbols that represent prescribed conditions*.

Additional information

- *Understanding and using toolboxes and toolbars*
- *What are step-dependent managers?*
- *Selecting objects within the viewport*
- *Using the predefined field editors*
- *The Set and Surface toolsets*

Editing the region to which a prescribed condition is applied

You can edit the region to which a prescribed condition is applied in the step in which the load, boundary condition, or predefined field was created.
You cannot edit the region if the definition of the prescribed condition refers to subregions within the original region (a material assignment predefined field, for example).

Note: Gravity loads can be applied to regions in your model, but they cannot be applied to individual point masses in Abaqus/CAE.

1. From the **Load**, **BC**, or **Predefined Field** menu in the main menu bar, select **Manager** to display the **Load Manager**, **Boundary Condition Manager**, or **Predefined Field Manager**.
2. Click the cell located in the row of the prescribed condition that you want to modify and in the column of the step of interest, and click **Edit**. Alternatively, you can just double-click the cell.

   Tip: You can also initiate this procedure by clicking the step in which the prescribed condition was created in the **Step** list located in the context bar. From the **Load**, **BC**, or **Predefined Field** menu in the main menu bar, select **Edit->*prescribed condition***. For example, to edit a load you would select **Load->Edit->*load of your choice***.

   An editor appears.

3. In the top part of the editor, click to edit the region selection.
4. Edit the region using one of the following methods:

   - Select and unselect objects in the viewport. When you have finished editing the region, click mouse button 2. (For more information, see *Selecting objects within the viewport*.)

     Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

   - To select from a list of existing sets or surfaces, do the following:

     1. Click **Sets** or **Surfaces** on the right side of the prompt area. (The name of the button depends on the type of object you are editing. For example, if you are editing a pressure load, a **Surfaces** button appears.)

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets or surfaces.

     2. Select the set or surface of interest and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** or **Surfaces** on the right side of the prompt area.

5. In the editor, finish editing the prescribed condition definition as desired and then click **OK**.

The symbols representing the prescribed condition in the viewport change to appear on the newly edited region.

Additional information

- *What are step-dependent managers?*
- *Understanding symbols that represent prescribed conditions*
- *Using the Load module*

Using the load editors

This section explains how to enter data in the load editor to define specific types of loads.

The following topics are covered in *Modeling techniques*:

- *Bolt loads*
- *Creating the submodel load*

In this section:

- *Defining a concentrated force*
- *Defining a moment*
- *Defining a pressure load*
- *Defining a shell edge load*
- *Defining a surface traction load*
- *Defining a pipe pressure load*
- *Defining a body force*
- *Defining a line load*
- *Defining a gravity load*
- *Defining a generalized plane strain load*
- *Defining a rotational body force*
- *Defining a Coriolis force*
- *Defining a connector force*
- *Defining a connector moment*
- *Defining a substructure load definition to activate a substructure load case*
- *Defining an inertia relief load*
- *Defining a surface heat flux*
- *Defining a body heat flux*
- *Defining a concentrated heat flux*
- *Defining an inward volume acceleration*
- *Defining a concentrated pore fluid flow*
- *Defining a surface pore fluid flow*
- *Defining a concentrated current*
- *Defining a surface current*
- *Defining a body current*
- *Defining a surface current density*
- *Defining a body current density*
- *Defining a concentrated charge*
- *Defining a surface charge*
- *Defining a body charge*
- *Defining a concentrated concentration flux*
- *Defining a surface concentration flux*
- *Defining a body concentration flux*
- *Defining a fluid pressure penetration load*

Defining a concentrated force

You can apply a concentrated force load to a vertex or node.

1. Display the concentrated force load editor using one of the following methods:
   - To create a new concentrated force load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Concentrated force`).
   - To edit an existing concentrated force load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **CF1**, **CF2**, and **CF3** text fields, enter the components of the concentrated force in each direction (units F):

   If you leave a text field blank, a force of zero is assigned to that direction automatically. However, you must enter at least one nonzero component in the editor to define the load.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. If desired, toggle on **Follow nodal rotation** to make the direction of the load rotate with the rotation at this node.

   Follow nodal rotation affects only nodes that have rotational degrees of freedom and steps in which the **Nlgeom** setting is turned on.
6. If you want to change the coordinate system (**CSYS**) for the concentrated force load, click  and use one of the following methods:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the concentrated force load is created. By default, the global coordinate system is used to define the load.
7. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*

- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Concentrated Loads*

Defining a moment

You can create a moment load to define rotation at a vertex or node.

1. Display the moment load editor using one of the following methods:
   - To create a new moment load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Moment`).
   - To edit an existing moment load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **CM1**, **CM2**, and **CM3** text fields, enter the components of the moment about each axis (units FL).

   If you leave a text field blank, a moment of zero is assigned to that direction automatically. However, you must enter at least one nonzero component in the editor to define the load.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. If desired, toggle on **Follow nodal rotation** to make the direction of the load rotate with the rotation at this node.

   Follow nodal rotation affects only nodes that have rotational degrees of freedom and steps in which the **Nlgeom** setting is turned on.
6. If you want to change the coordinate system (**CSYS**) for the moment, click  and use one of the following methods:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.

     Note:

     You should not apply a moment load at the origin of a cylindrical coordinate system; doing so would make the radial and tangential loads indeterminate.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the moment is created. By default, the global coordinate system is used to define the moment.

7. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Concentrated Loads*

Defining a pressure load

You can create a pressure load to define a pressure over a surface.

1. Display the pressure load editor using one of the following methods:
   - To create a new pressure load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Pressure`).
   - To edit an existing pressure load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a pressure that is uniformly distributed over the surface. For this option, the magnitude you provide must be the force per unit area.
   - Select **Total Force** to define a pressure that is uniformly distributed over the surface. For this option, the magnitude you provide must be the total magnitude of the force applied to the surface (instead of force per unit area).
   - Select **Hydrostatic** to define a hydrostatic pressure applied to the surface. (This option is valid only for Abaqus/Standard analyses.)
   - Select **Stagnation** to define a stagnation pressure applied to the surface. (This option is valid only for Abaqus/Explicit analyses.)
   - Select **Viscous** to define a viscous pressure applied to the surface. (This option is valid only for Abaqus/Explicit analyses.)
   - Select **User-defined** to define the magnitude of the load in user subroutine *DLOAD* (for Abaqus/Standard) or *VDLOAD* (for Abaqus/Explicit). See the following sections for more information:
     - *Specifying general job settings*
     - *DLOAD*
     - *VDLOAD*
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying pressure. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, **Total Force**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the pressure magnitude.

   For a **Uniform** distribution, enter the total force magnitude divided by the surface area over which the force is applied (units FL^{-2}).

   For a **Total Force** distribution, enter the total magnitude of the force (units F). Based on the undeformed model geometry, Abaqus/CAE calculates a constant uniform surface pressure from the force magnitude entered. In a large-displacement analysis, however, the actual total force may change during the analysis due to the deformation of the loaded surface.

b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

c. Click **OK** to save your data and to exit the editor.

4. If you selected the **Hydrostatic** distribution option, perform the following steps:

a. In the **Magnitude** text field, enter the pressure magnitude (units FL^{-2}).

b. In the **Zero pressure height** field, enter the *Z*-coordinate (if you are working in three-dimensional or axisymmetric space) or the *Y*-coordinate (if you are working in two-dimensional space) of the height at which the pressure is zero.

c. In the **Reference pressure height** field, enter the *Z*-coordinate (if you are working in three-dimensional or axisymmetric space) or the *Y*-coordinate (if you are working in two-dimensional space) of the height at which the pressure is the magnitude specified in the **Magnitude** field.

(For more information, see *Hydrostatic Pressure Loads on Two-Dimensional, Three-Dimensional, and Axisymmetric Elements in Abaqus/Standard*.)

d. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

5. If you selected the **Stagnation** or **Viscous** distribution option, perform the following steps:

a. In the **Magnitude** text field, enter the pressure magnitude (units FL^{-2}).

b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

c. If desired, toggle on **Determine velocity from reference point** to subtract the velocity of a reference node from the velocity of the surface where the pressure is applied.

d. Click  to select a reference point using one of the following methods:

- Select a point from the viewport.
- Click **Points** in the prompt area, and select a named set.



Note:

The set that you select must contain a single node or vertex.

e. Click **OK** to save your data and to exit the editor.

6. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the pressure magnitude in the **Magnitude** field (units FL^{-2}). Magnitude data that you enter in the editor are passed into the user subroutine in an Abaqus/Standard analysis but are ignored in an Abaqus/Explicit analysis.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Creating discrete fields*
- *Distributed Loads*

Defining a shell edge load

You can create a shell edge load to define a general, shear, normal, or transverse traction or a moment along a shell edge.

1. Display the shell edge load editor using one of the following methods:
   - To create a new shell edge load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Shell edge load`).
   - To edit an existing shell edge load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the shell edge.
   - Select **User-defined** to define the magnitude of the load in user subroutine *UTRACLOAD* (for Abaqus/Standard). See the following sections for more information:
     - *Specifying general job settings*
     - *UTRACLOAD*
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. Click the arrow to the right of the **Traction** field, and select the option of your choice from the list that appears:
   - Select **Normal** to define a normal shell edge traction.
   - Select **Transverse** to define a transverse shell edge traction.
   - Select **Shear** to define a shear shell edge traction.
   - Select **Moment** to define a shell edge moment.
   - Select **General** to define a general shell edge traction.

4. If you selected the **General** traction type, specify the load direction.
   a. Click  next to **Vector** to specify the coordinates of the direction vector.
   b. By default, the traction components are specified with respect to the global axes. To refer to a local coordinate system for the direction components of the traction:
      - Select **CSYS: Picked** and click  to pick a previously defined local coordinate system.
      - Select **CSYS: User-defined** and enter the name of a user subroutine that defines a local coordinate system.
   c. If you selected **CSYS: Picked**, you can define an additional rotation about one of the axes. Click the arrow to the right of the **Additional rotation about axis** field, select the axis about which the other two axes will be rotated, and enter a value for the additional rotation angle.

5. In the **Magnitude** text field, enter the shell edge load magnitude (units FL^{-1}).

6. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
7. If desired, click the arrow to the right of the **Traction is defined per unit** field, and select **deformed area** to define the shell edge load with respect to the current (deformed) area or **undeformed area** to define the shell edge load with respect to the reference (original) area.
8. If you selected the **General** traction type, you can toggle off **Follow rotation** to define a non-follower load (i.e., the load always acts in a fixed global direction rather than rotating with the shell edge in a geometrically nonlinear analysis).
9. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Distributed Loads*

Defining a surface traction load

You can create a surface traction load to define a general or shear traction over a surface.

1. Display the surface traction load editor using one of the following methods:
   - To create a new surface traction load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Surface traction`).
   - To edit an existing surface traction load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select **User-defined** to define the magnitude of the load in user subroutine *UTRACLOAD* (for Abaqus/Standard). See the following sections for more information:
     - *Specifying general job settings*
     - *UTRACLOAD*
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. Click the arrow to the right of the **Traction** field, and select the option of your choice from the list that appears:
   - Select **Shear** to define a shear surface traction.
   - Select **General** to define a general surface traction.
4. Specify the load direction.
   a. Click  next to **Vector** or **Vector before projection** to specify the coordinates of the direction vector.
   b. By default, the traction components are specified with respect to the global axes. To refer to a local coordinate system for the direction components of the traction:
      - Select **CSYS: Picked** and click  to pick a previously defined local coordinate system.
      - Select **CSYS: User-defined** and enter the name of a user subroutine that defines a local coordinate system.
   c. If you selected **CSYS: Picked**, you can define an additional rotation about one of the axes. Click the arrow to the right of the **Additional rotation about axis** field, select the axis about which the other two axes will be rotated, and enter a value for the additional rotation angle.
5. In the **Magnitude** text field, enter the surface traction magnitude (units FL^{-2}).

6. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
7. If desired, click the arrow to the right of the **Traction is defined per unit** field, and select **deformed area** to define the surface traction with respect to the current (deformed) area or **undeformed area** to define the surface traction with respect to the reference (original) area.
8. If you selected the **General** traction type, you can toggle off **Follow rotation** to define a non-follower load in a geometrically nonlinear analysis (i.e., the load always acts in a fixed global direction rather than rotating with the surface).
9. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Distributed Loads*

Defining a pipe pressure load

You can create this type of load to prescribe internal or external pressure in a pipe or elbow.

1. Display the pipe pressure load editor using one of the following methods:
   - To create a new pipe pressure load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Pipe pressure`).
   - To edit an existing pipe pressure load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Select the **Side** option of your choice:
   - Select **Internal** to prescribe an internal pressure within the pipe.
   - Select **External** to prescribe an external pressure on the pipe.
3. In the **Effective diameter** field, enter the appropriate pipe diameter:
   - Enter the inner diameter of the pipe if you selected **Internal** in the previous step.
   - Enter the outer diameter of the pipe if you selected **External** in the previous step.

   Note:

   The effective diameter that you enter remains constant throughout the analysis. It is not scaled as the pipe expands or contracts under the pressure, even when the **Nlgeom** setting is turned on. (For more information on the **Nlgeom** setting, see *Accounting for geometric nonlinearity*.)
4. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the pipe surface.
   - Select **Hydrostatic** to define a hydrostatic pressure on or within the pipe.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DLOAD*. See the following sections for more information:
     - *Specifying general job settings*
     - *DLOAD*
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
5. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the pressure magnitude (units FL^{-2}).
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.
6. If you selected the **Hydrostatic** distribution option, perform the following steps:

a. In the **Magnitude** text field, enter the pressure magnitude (units FL^{-2}).

b. In the **Zero pressure height** field, enter the *Z*-coordinate of the height at which the pressure is zero.

c. In the **Reference pressure height** field, enter the *Z*-coordinate of the height at which the pressure is the magnitude specified in the **Magnitude** field.

(For more information, see *Hydrostatic Pressure Loads on Two-Dimensional, Three-Dimensional, and Axisymmetric Elements in Abaqus/Standard*.)

d. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

7. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the pressure magnitude in the **Magnitude** field (units FL^{-2}). Magnitude data that you enter in the editor are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine *DLOAD*. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Distributed Loads*

Defining a body force

You can define a body force to prescribe loading per unit volume over a body.

1. Display the body force load editor using one of the following methods:
   - To create a new body force load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Body force`).
   - To edit an existing body force load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. If the **Distribution** field appears in the editor, click the arrow to the right of the field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DLOAD* (for Abaqus/Standard) or *VDLOAD* (for Abaqus/Explicit). See the following sections for more information:
     - *Specifying general job settings*
     - *DLOAD*
     - *VDLOAD*
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click **Create** to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. In the **Component 1**, **Component 2**, and (if you are working in three-dimensional space) **Component 3** fields, enter the body force per unit volume in each direction (units FL^{-3}):
      - If you are working in three-dimensional or two-dimensional space, the **Component 1**, **Component 2**, and **Component 3** fields correspond to the 1-, 2-, and (if applicable) 3-directions.
      - If you are working in axisymmetric space, **Component 1** corresponds to the radial direction and **Component 2** corresponds to the axial direction.

   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.
4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, in the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields enter the body force per unit volume in each direction (units FL^{-3}).

      Load magnitude data that you enter in the editor are passed into the user subroutine in an Abaqus/Standard analysis but are ignored in an Abaqus/Explicit analysis.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Distributed Loads*

Defining a line load

You can create a line load to prescribe the force per unit length over a beam.

1. Display the line load editor using one of the following methods:
   - To create a new line load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Line load`).
   - To edit an existing line load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **System** field, and from the list that appears select the coordinate system in which you want to define the load:
   - Select **Global** if you want to specify the load components in the global 1-, 2-, and (if you are working in three-dimensional space) 3-directions.
   - Select **Local** if you want to specify the load components in the beam local 1-direction (if you are working in three-dimensional space) and the beam local 2-direction. (For more information, see *Assigning a beam orientation*.)
3. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DLOAD* (for Abaqus/Standard) or *VDLOAD* (for Abaqus/Explicit). See the following sections for more information:
     - *Specifying general job settings*
     - *DLOAD*
     - *VDLOAD*
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. In the **Component** fields, enter the body force per unit length in each direction (units FL^{-1}):
      - If you selected the **Global** system, the **Component 1**, **Component 2**, and **Component 3** fields correspond to the 1-, 2-, and 3-directions.
      - If you selected the **Local** system, the **Component 1** field corresponds to the beam local 1-direction, and the **Component 2** field corresponds to the beam local 2-direction.

   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.
5. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, in the **Component** fields enter the force per unit length in each direction (units FL^{-1}).

Entering load magnitude data in the editor is optional for user-defined loads. Any data you enter are passed to the user subroutine in an Abaqus/Standard analysis but are ignored in an Abaqus/Explicit analysis.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Distributed Loads*

Defining a gravity load

You can create a gravity load to define a uniform acceleration in a fixed direction. Abaqus calculates the loading using the acceleration magnitude that you enter in the gravity load definition and the density specified in the material definition.

1. Display the gravity load editor using one of the following methods:
   - To create a new gravity load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Gravity`).
   - To edit an existing gravity load using menus or managers, see *Editing step-dependent objects*.
2. By default, a gravity load is applied to the whole model. If desired, you can apply a gravity load to particular regions of a model:
   a. Click .
   b. Select the region to which you want to apply the load, as described in *Creating loads*.
   c. Click **Done** in the prompt area.

   Note:

   You cannot apply gravity loads to individual point masses. To include point masses in a gravity load, apply the load to the whole model.
3. Click the arrow to the right of the **Distribution** field, if available; and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)
4. In the **Component 1**, **Component 2**, and (if you are working with a model in three-dimensional space) **Component 3** text fields, enter the components of the acceleration in each direction:
   - If you are working in three-dimensional or two-dimensional space, the **Component 1**, **Component 2**, and **Component 3** fields correspond to the 1-, 2-, and (if applicable) 3-directions.
   - If you are working in axisymmetric space, only the **Component 2** text field is available. **Component 2** corresponds to the axial direction.

   If you leave a text field blank, a value of zero is assigned to that direction automatically. However, you must enter at least one nonzero component in the editor to define the load.
5. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*

- *Creating expression fields*
- *Distributed Loads*

Defining a generalized plane strain load

You can create a generalized plane strain load to define an axial load applied to the reference point of a region modeled with generalized plane strain elements.
A generalized plane strain load cannot be defined for a coupled temperature-displacement analysis.

1. Display the generalized plane strain load editor using one of the following methods:
   - To create a new generalized plane strain load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Generalized plane strain`).
   - To edit an existing generalized plane strain load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Axial force** text field, enter the axial force (units F).
4. In the **Moment about X** field, enter the moment applied at the reference point about the *X*-axis.
5. In the **Moment about Y** field, enter the moment applied at the reference point about the *Y*-axis.
6. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
7. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Adding unsupported keywords to your Abaqus/CAE model*
- *Generalized Plane Strain Elements*

Defining a rotational body force

You can create a rotational body force load to define loads resulting from the rotation of the model.

You can specify angular velocity, rotary acceleration, or rotordynamic load. Abaqus defines the load by using the square of the angular velocity or by using the acceleration directly. In any case the load definition must include an axis of rotation, which is defined as follows:

- If you are working in three-dimensional space, you define the location and direction of the axis by entering the global coordinates of two points.
- If you are working in two-dimensional space, you specify the location of the axis by entering the coordinates of a point in the plane. The direction of the axis is always out of the plane.
- If you are working in axisymmetric space, the axis is always in the location and direction of the positive global z-axis.



Note: You can define a rotary acceleration force only in two- or three-dimensional space.

1. Display the rotational body force load editor using one of the following methods:

   - To create a new rotational body force load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Rotational body force`).

     If you are working in two- or three-dimensional space, enter in the prompt area the required information concerning the location and, if applicable, the direction of the axis of rotation.
   - To edit an existing rotational body force load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

     If you are editing a load in the step in which it was created, an **Edit** () button appears next to each point you specified in the load editor. Click  if you want to change the coordinates that determine the location and, if applicable, direction of the axis of rotation. (This option applies only if you are working in two- or three-dimensional space.)

2. Select the force type of your choice:

   - Toggle **Centrifugal** if you want to define a centrifugal force.
   - Toggle **Rotary acceleration** if you want to define a rotary acceleration force.
   - Toggle **Rotordynamic load** if you want to define a rotordynamic load.

3. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:

   - Select **Uniform** to define a load that is uniform over the body.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

4. In the text field that appears, enter the appropriate value:

   - If you are defining a centrifugal force, enter the angular velocity in radians/time.
   - If you are defining a rotary acceleration force, enter the rotary acceleration in radians/time^2.
   - If you are defining a rotordynamic load, enter the rotordynamic load in radians/time.

5. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   For centrifugal loading, the amplitude is applied to the calculated load, not the angular velocity.

6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Specifying Loads due to Rotation of the Model in Abaqus/Standard*

Defining a Coriolis force

You can create a Coriolis force load to define loads resulting from rotation of the model. You must specify the force as the product of the material density (mass per unit volume) for solid and shell elements or the mass per unit length for beam elements and the angular velocity in radians per time. The load definition must include an axis of rotation, which is defined as follows:

- If you are working in three-dimensional space, you define the location and direction of the axis by entering the global coordinates of two points.
- If you are working in two-dimensional space, you specify the location of the axis by entering the coordinates of a point in the plane. The direction of the axis is always out of the plane.
- If you are working in axisymmetric space, the axis is always in the location and direction of the positive global *Z*-axis.

1. Display the Coriolis force load editor using one of the following methods:
   - To create a new rotational body force load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Coriolis force`).

     If you are working in two- or three-dimensional space, enter in the prompt area the required information concerning the location and, if applicable, direction of the axis of rotation.
   - To edit an existing Coriolis force load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

     If you are editing a load in the step in which it was created, an **Edit** () button appears next to each point you specified in the load editor. Click  if you want to change the coordinates that determine the location and, if applicable, direction of the axis of rotation. (This option applies only if you are working in two- or three-dimensional space.)
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Coriolis force** field, enter the magnitude of the force.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*

- *Creating expression fields*
- *Specifying Loads due to Rotation of the Model in Abaqus/Standard*

Defining a connector force

You can create a connector force to apply a concentrated force to the available components of relative motion of connectors.

1. Display the connector force load editor using one of the following methods:
   - To create a new connector force, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Connector force`).
   - To edit an existing connector force using menus or managers, see *Editing step-dependent objects*. To edit the region to which the connector force is applied, see *Editing the region to which a prescribed condition is applied*.
2. In the available **F1**, **F2**, and **F3** text fields, enter the components of the concentrated force for each available translational component of relative motion (units F). Only components common to all of the selected wires are available in the editor.

   If you leave a text field blank, a force of zero is assigned to that component of relative motion automatically. However, you must enter at least one nonzero component in the editor to define the load.

   Warning: The connector force will be applied regardless of whether the connector force exceeds the failure criteria for the connector.
3. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
4. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *About Connectors*

Defining a connector moment

You can create a connector moment to apply a concentrated moment to the available components of relative motion of connectors.

1. Display the connector moment load editor using one of the following methods:
   - To create a new connector moment, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Connector moment`).
   - To edit an existing connector moment using menus or managers, see *Editing step-dependent objects*. To edit the region to which the connector moment is applied, see *Editing the region to which a prescribed condition is applied*.
2. In the available **M1**, **M2**, and **M3** text fields, enter the components of the moment for each available rotational component of relative motion (units FL). Only components common to all of the selected wires are available in the editor.

   If you leave a text field blank, a moment of zero is assigned to that component of relative motion automatically. However, you must enter at least one nonzero component in the editor to define the load.

    **Warning:** The connector moment will be applied regardless of whether the connector moment exceeds the failure criteria for the connector.
3. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
4. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *About Connectors*

Defining a substructure load definition to activate a substructure load case

You can create a substructure load definition in a substructure usage model to activate a substructure load case for one or more substructures in your analysis.

The activation process enables you to change the magnitude of substructure load case loads and boundary conditions using two options:

- The magnitude multiplier option applies a scaling value to the specified substructure load case loads and boundary conditions. To reproduce the loading conditions defined during substructure generation exactly, accept the default magnitude of 1.0.
- The amplitude selection enables you to apply a variable multiplying factor to the substructure load case loads and boundary conditions.

For more information about load cases in Abaqus/CAE, see *Load cases*.

1. Display the substructure load editor using one of the following methods:
   - To create a new substructure load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Substructure load`).
   - To edit an existing substructure load using menus or managers, see *Editing step-dependent objects*.
2. Do the following:
   a. Click  to open the **Select Substructure Load Cases** dialog box.
   b. Toggle on check boxes for each substructure load case that you want to activate.
   c. Click **OK** to close the **Select Substructure Load Cases** dialog box.
3. In the **Magnitude multiplier** text field, enter the value by which you want to scale the specified substructure load case loads and boundary conditions. The default value is 1.0, which provides no scaling. You can enter only real values for the magnitude multiplier. To specify complex values, you must use the **Keywords Editor**, as described in *Adding unsupported keywords to your Abaqus/CAE model*.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Defining a load case*
- *Using Substructures*

Defining an inertia relief load

You can create an inertia relief load to balance externally applied forces on a free or partially free body. An inertia relief load is applied to the whole model. You can apply only one active inertia relief load for each general analysis step. For detailed information about inertia relief loads, see *Inertia Relief*.

Note:

You cannot apply inertia relief loads in submodels. For submodels Abaqus ignores the inertia relief effect computed by including an inertia relief load in the global model.

1. Display the inertia relief load editor using one of the following methods:
   - To create a new inertia relief load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Inertia relief`).
   - To edit an existing inertia relief load using menus or managers, see *Editing step-dependent objects*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Compute loading** to continue to compute loading for the specified directions.
   - Select **Fix at current loading** to fix the loading at the magnitude and direction from the previous step.

   The **Method** option is unavailable in the step in which the inertia relief load is created.
3. Toggle on a degree of freedom to define a free direction along which you want to apply the inertia relief load. The degrees of freedom displayed are dependent on the modeling space.
4. If you want to change the coordinate system (**CSYS**) for the inertial relief load, click  and use one of the following methods:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the inertia relief load is created. By default, the global coordinate system is used to define the load.
5. If **X**, **Y**, and **Z** text fields appear at the bottom of the editor, enter the coordinates of the additional point that is required to define the rigid body motion. You must define an additional point for certain combinations of free directions. For more information, see *Inertia Relief*.
6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Inertia Relief*

Defining a surface heat flux

You can create a surface heat flux load to define surface-based heat fluxes.

1. Display the surface heat flux load editor using one of the following methods:
   - To create a new surface heat flux load, follow the procedure outlined in *Creating loads* (**Category**: `Thermal`; **Types for Selected Step**: `Surface heat flux`).
   - To edit an existing surface heat flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface. For this option, the magnitude you provide must be the flux per unit area.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DFLUX*. (This option is valid only in Abaqus/Standard analyses.) See the following sections for more information:
     - *Specifying general job settings*
     - *DFLUX*
   - Select **Total Flux** to define a load that is uniform over the surface. For this option, the magnitude you provide must be the total magnitude of the flux applied to the surface (instead of flux per unit area).
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying surface heat flux. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, **Total Flux**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the surface heat flux magnitude. A positive magnitude indicates heat flow into the surface.

      For a **Uniform** distribution, enter the total flux magnitude divided by the surface area over which the flux is applied (units $JT^{-1}L^{-2}$).

      For a **Total Flux** distribution, enter the total magnitude of the flux (units JT^{-1}). Abaqus/CAE calculates a constant uniform surface flux from the flux magnitude entered.

   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the surface heat flux magnitude in the **Magnitude** field (units $JT^{-1}L^{-2}$). A positive magnitude indicates heat flow into the surface.

      Magnitude data that you enter in the editor are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *Thermal Loads*

Defining a body heat flux

You can create a body heat flux load to define distributed heat fluxes over a volume.

1. Display the body heat flux load editor using one of the following methods:
   - To create a new body heat flux load, follow the procedure outlined in *Creating loads* (**Category**: `Thermal`; **Types for Selected Step**: `Body heat flux`).
   - To edit an existing body heat flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DFLUX*. (This option is valid only in Abaqus/Standard analyses.) See the following sections for more information:
     - *Specifying general job settings*
     - *DFLUX*
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying load. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the body heat flux magnitude (units $JT^{-1}L^{-3}$). A positive magnitude indicates heat flow into the body.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the body heat flux magnitude in the **Magnitude** field (units $JT^{-1}L^{-3}$). A positive magnitude indicates heat flow into the body.

   Magnitude data that you enter in the editor are passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *Thermal Loads*

Defining a concentrated heat flux

You can apply a concentrated heat flux to a vertex or node.

1. Display the concentrated heat flux load editor using one of the following methods:
   - To create a new concentrated heat flux load, follow the procedure outlined in *Creating loads* (**Category**: `Thermal`; **Types for Selected Step**: `Concentrated heat flux`).
   - To edit an existing concentrated heat flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the concentrated heat flux magnitude (units JT^{-1}). A positive magnitude indicates heat flow into the body at the vertex or node.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. If you are working with shell elements and want to apply the heat flux to a degree of freedom other than 11, enter the desired degree of freedom in the **Degree of freedom** field. Otherwise, accept the default degree of freedom.

   You can enter any number between 11 and 31 in the **Degree of freedom** field. If you prefer, you can use the arrows on the right side of the field to scroll through the range of valid degrees of freedom.

   For more information, see "Specifying concentrated heat fluxes," in *Thermal Loads*.
6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Thermal Loads*

Defining an inward volume acceleration

You can create an inward volume acceleration load to specify a volume acceleration at a vertex or node on the boundary of an acoustic medium. (For more information, see "Loads," in *Coupled Acoustic-Structural Analysis*.)

1. Display the inward volume acceleration load editor using one of the following methods:
   - To create a new inward volume acceleration load, follow the procedure outlined in *Creating loads* (**Category**: `Acoustic`; **Types for Selected Step**: `Inward volume acceleration`).
   - To edit an existing inward volume acceleration load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the volume acceleration (units L^3T^{-2}).
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Acoustic-Structural Analysis*

Defining a concentrated pore fluid flow

You can create a concentrated pore fluid flow load to define concentrated pore fluid flow at a vertex or node in soils analyses.

1. Display the concentrated pore fluid flow load editor using one of the following methods:
   - To create a new concentrated pore fluid flow load, follow the procedure outlined in *Creating loads* (**Category**: `Fluid`; **Types for Selected Step**: `Concentrated pore fluid`).
   - To edit an existing concentrated pore fluid flow load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the flow rate (units L^3T^{-1}). A positive magnitude indicates fluid flow out of the body at the vertex or node.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Pore Fluid Flow*

Defining a surface pore fluid flow

You can create a surface pore fluid flow load to define pore fluid flow velocities normal to surfaces in soils analyses.

1. Display the surface pore fluid flow load editor using one of the following methods:
   - To create a new surface pore fluid flow load, follow the procedure outlined in *Creating loads* (**Category**: `Fluid`; **Types for Selected Step**: `Surface pore fluid`).
   - To edit an existing surface pore fluid flow load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DFLOW*. See the following sections for more information:
     - *Specifying general job settings*
     - *DFLOW*
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying load. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the flow velocity (units LT^{-1}). A positive magnitude indicates fluid flow into the surface.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the flow velocity in the **Magnitude** field (units LT^{-1}). A positive magnitude indicates fluid flow into the surface.

      Magnitude data that you enter in the editor are passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *Pore Fluid Flow*

Defining a concentrated current

You can create a concentrated current load to define a concentrated current at a vertex or node in a coupled thermal-electrical analysis.

1. Display the concentrated current load editor using one of the following methods:
   - To create a new concentrated current load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Concentrated current`).
   - To edit an existing concentrated current load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the current magnitude (units CT^{-1}). A positive magnitude indicates current flow into the body at the vertex or node.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Thermal-Electrical Analysis*

Defining a surface current

You can create a surface current load to define current densities over a surface in a coupled thermal-electrical analysis.

1. Display the surface current load editor using one of the following methods:
   - To create a new surface current load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Surface current`).
   - To edit an existing surface current load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** text field, enter the current density (units $C\,L^{-2}T^{-1}$). A positive magnitude indicates current flow into the surface.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Thermal-Electrical Analysis*

Defining a body current

You can create a body current load to define current densities over a body in a coupled thermal-electrical analysis.

1. Display the body current load editor using one of the following methods:
   - To create a new body current load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Body current`).
   - To edit an existing body current load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** text field, enter the current density (units C $L^{-3}T^{-1}$). A positive magnitude indicates current flow into the body.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Thermal-Electrical Analysis*

Defining a surface current density

You can create a surface current density load to define current density over a surface in an eddy current analysis. The surface current density load is available only in an electromagnetic model.

1. Display the surface current density load editor using one of the following methods:
   - To create a new surface current density load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Surface current density`).
   - To edit an existing surface current density load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select **User-defined** to define the magnitude and direction of the load in user subroutine *UDSECURRENT*. See the following sections for more information:
     - *Specifying general job settings*
     - *UDSECURRENT*

3. If you selected the **Uniform** distribution option, perform the following steps:
   a. In the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields, enter the real (in-phase) and imaginary (out-of-phase) parts of the surface current density vector.

      Abaqus/CAE calculates the magnitude and direction of the surface current density vector.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, in the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields enter the real (in-phase) and imaginary (out-of-phase) parts of the surface current density vector.

      Abaqus/CAE calculates the magnitude and direction of the surface current density vector and that information is passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the magnitude and direction of the load. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Eddy Current Analysis*
- *Electromagnetic Loads*

Defining a body current density

You can create a body current density load to define current density over a volume in an eddy current analysis. The body current density load is available only in an electromagnetic model.

1. Display the body current density load editor using one of the following methods:
   - To create a new body current density load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Body current density`).
   - To edit an existing body current density load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the volume.
   - Select **User-defined** to define the magnitude and direction of the load in user subroutine *UDECURRENT*. See the following sections for more information:
     - *Specifying general job settings*
     - *UDECURRENT*

3. If you selected the **Uniform** distribution option, perform the following steps:
   a. In the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields, enter the real (in-phase) and imaginary (out-of-phase) parts of the body current density vector.

      Abaqus/CAE calculates the magnitude and direction of the body current density vector.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, in the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields enter the real (in-phase) and imaginary (out-of-phase) parts of the body current density vector.

      Abaqus/CAE calculates the magnitude and direction of the body current density vector and that information is passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the magnitude and direction of the load. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Eddy Current Analysis*
- *Electromagnetic Loads*

Defining a concentrated charge

You can create a concentrated charge load to apply an electric charge to a vertex or node in a piezoelectric analysis.

1. Display the concentrated charge load editor using one of the following methods:
   - To create a new concentrated charge load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Concentrated charge`).
   - To edit an existing concentrated charge load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the charge magnitude (units C).
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Note:

   The amplitude field appears only if amplitudes are valid for the type of perturbation step in which you are creating the load. For more information, see “Linear perturbation analysis steps,” in *General and Perturbation Procedures*.
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Piezoelectric Analysis*

Defining a surface charge

You can create a surface charge load to apply a distributed electric charge to a surface in a piezoelectric analysis.

1. Display the surface charge load editor using one of the following methods:
   - To create a new surface charge load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Surface charge`).
   - To edit an existing surface charge load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the charge density (units CL^{-2}).
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Note:

   The amplitude field appears only if amplitudes are valid for the type of perturbation step in which you are creating the load. For more information, see “Linear perturbation analysis steps,” in *General and Perturbation Procedures*.
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Piezoelectric Analysis*

Defining a body charge

You can create a body charge load to apply an electric charge to a body in a piezoelectric analysis.

1. Display the body charge load editor using one of the following methods:
   - To create a new body charge load, follow the procedure outlined in *Creating loads* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Body charge`).
   - To edit an existing body charge load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the charge density (units CL^{-3}).
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

    **Note:**

   The amplitude field appears only if amplitudes are valid for the type of perturbation step in which you are creating the load. For more information, see “Linear perturbation analysis steps,” in *General and Perturbation Procedures*.
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Piezoelectric Analysis*

Defining a concentrated concentration flux

You can create a concentrated concentration flux load to define a concentrated concentration flux at a vertex or node in a mass diffusion analysis.

1. Display the concentrated concentration flux load editor using one of the following methods:
   - To create a new concentrated concentration flux load, follow the procedure outlined in *Creating loads* (**Category**: `Mass diffusion`; **Types for Selected Step**: `Concentrated concentration flux`).
   - To edit an existing concentrated concentration flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the region.
   - Select an analytical field to define a spatially varying load. Only analytical fields that are valid for this load type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. In the **Magnitude** field, enter the concentration flux magnitude (units PL^3T^{-1}). A positive magnitude indicates concentration flow into the body at the vertex or node.
4. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Mass Diffusion Analysis*

Defining a surface concentration flux

You can create a surface concentration flux load to define a concentration flux over a surface in a mass diffusion analysis.

1. Display the surface concentration flux load editor using one of the following methods:
   - To create a new surface concentration flux load, follow the procedure outlined in *Creating loads* (**Category**: `Mass diffusion`; **Types for Selected Step**: `Surface concentration flux`).
   - To edit an existing surface concentration flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the surface.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DFLUX*. See the following sections for more information:
     - *Specifying general job settings*
     - *DFLUX*
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying load. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the concentration flux density (units PLT^{-1}). A positive magnitude indicates concentration flow into the surface.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the concentration flux density in the **Magnitude** field (units PLT^{-1}). A positive magnitude indicates concentration flow into the surface.

      Magnitude data that you enter in the editor are passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *Mass Diffusion Analysis*

Defining a body concentration flux

You can create a body concentration flux load to define a concentration flux over a body in a mass diffusion analysis.

1. Display the body concentration flux load editor using one of the following methods:
   - To create a new body concentration flux load, follow the procedure outlined in *Creating loads* (**Category**: `Mass diffusion`; **Types for Selected Step**: `Body concentration flux`).
   - To edit an existing body concentration flux load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a load that is uniform over the body.
   - Select **User-defined** to define the magnitude of the load in user subroutine *DFLUX*. See the following sections for more information:
     - *Specifying general job settings*
     - *DFLUX*
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying load. Only analytical fields and discrete fields that are valid for this load type are displayed in the selection list.

     Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. If you selected the **Uniform**, analytical field, or discrete field distribution option, perform the following steps:
   a. In the **Magnitude** text field, enter the concentration flux density (units PT^{-1}). A positive magnitude indicates concentration flow into the body.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the concentration flux density in the **Magnitude** field (units PT^{-1}). A positive magnitude indicates concentration flow into the body.

      Magnitude data that you enter in the editor are passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the load magnitude. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *The Analytical Field toolset*
- *The Discrete Field toolset*
- *Mass Diffusion Analysis*

Defining a fluid pressure penetration load

You can create a fluid pressure penetration load to define a fluid pressure over a surface.

1. Display the fluid pressure load editor using one of the following methods:
   - To create a new fluid pressure load, follow the procedure outlined in *Creating loads* (**Category**: `Mechanical`; **Types for Selected Step**: `Fluid pressure penetration`).
   - To edit an existing fluid pressure load using menus or managers, see *Editing step-dependent objects*. To edit the region to which the load is applied, see *Editing the region to which a prescribed condition is applied*.
2. In the **Magnitude** text field, enter the fluid pressure magnitude (units FL^{-2}).
3. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears.
4. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
5. In the **Critical pressure** text field, enter the critical pressure.
6. In the **Time period** text field, enter the time period.

   This field is supported only for Abaqus/Standard analyses (linear perturbation analyses are not supported).
7. Click the arrow to the right of the **Wetting type** field, and select the option of your choice from the list that appears:
   - Select **LOCAL** to define the algorithm for controlling the evolution region.
   - Select **WETTING ADVANCE** to define the algorithm for controlling the evolution region in Abaqus/Explicit analyses.
8. If you selected **WETTING ADVANCE** in the **Wetting type** field, perform the following steps to define the evolution regions in the table.
   a. In the **Evolutions region** column, double-click the cell of the table or right-click and select **Edit region**.
   b. Select a region in the viewport.

      A default name is provided in the prompt area, but you can enter a new name.
   c. To select from a list of existing sets, do the following:
      1. Click **Sets** on the right side of the prompt area.
      2. Select the set or surface of interest, and click **Continue**.
9. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Creating discrete fields*
- *Fluid Pressure Penetration Loads*

Using the boundary condition editors

This section explains how to define specific types of boundary conditions.

The following topic is covered in *Modeling techniques*:

- *Creating the submodel boundary condition*

In this section:

- *Defining a symmetry/antisymmetry/encastre boundary condition*
- *Defining a displacement/rotation boundary condition*
- *Defining a velocity/angular velocity boundary condition*
- *Defining an acceleration/angular acceleration boundary condition*
- *Defining a connector displacement boundary condition*
- *Defining a connector velocity boundary condition*
- *Defining a connector acceleration boundary condition*
- *Defining a base motion boundary condition*
- *Defining a secondary base motion boundary condition*
- *Defining a retained nodal DOFs boundary condition*
- *Specifying temperature*
- *Defining a pore pressure boundary condition*
- *Defining a fluid cavity pressure boundary condition*
- *Defining an electric potential boundary condition*
- *Defining a magnetic vector potential boundary condition*
- *Defining a normalized concentration boundary condition*
- *Defining an acoustic pressure boundary condition*
- *Defining a connector material flow boundary condition*
- *Defining an Eulerian boundary condition*
- *Defining an Eulerian mesh motion boundary condition*

Defining a symmetry/antisymmetry/encastre boundary condition

You can define a boundary condition by selecting one of the common types listed in the symmetry/antisymmetry/encastre boundary condition editor.

1. Display the symmetry/antisymmetry/encastre boundary condition editor using one of the following methods:
   - To create a new symmetry/antisymmetry/encastre boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Symmetry/Antisymmetry/Encastre`).
   - To edit an existing symmetry/antisymmetry/encastre boundary condition using menus or managers, see *Editing step-dependent objects*. You can edit the symmetry/antisymmetry/encastre boundary condition only in the step in which it was created.
2. If you are creating the boundary condition in a buckling step, select the **Use BC for** option that specifies the calculations for which you want the boundary condition used. For more information, see "Boundary conditions," in *Eigenvalue Buckling Prediction*.
3. By default, the global coordinate system is used to define the boundary condition. To change the coordinate system in which the boundary condition is applied, click  for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.
4. Select one of the following options:

XSYMM

Symmetry about a plane X = constant (U1 = UR2 = UR3 = 0).

YSYMM

Symmetry about a plane Y = constant (U2 = UR1 = UR3 = 0).

ZSYMM

Symmetry about a plane Z = constant (U3 = UR1 = UR2 = 0).

XASYMM

Antisymmetry about a plane with X = constant (U2 = U3 = UR1 = 0;Abaqus/Standard only).

YASYMM

Antisymmetry about a plane with Y = constant (U1 = U3 = UR2 = 0;Abaqus/Standard only).

ZASYMM

Antisymmetry about a plane with Z = constant (U1 = U2 = UR3 = 0;Abaqus/Standard only).

PINNED

Pinned (U1 = U2 = U3 = 0).

ENCASTRE

Fully built-in (U1 = U2 = U3 = UR1 = UR2 = UR3 = 0).

5. Click **OK** to save your data and to exit the editor.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Boundary Conditions*

Defining a displacement/rotation boundary condition

You can create a displacement/rotation boundary condition to constrain the movement of the selected degrees of freedom to zero or to prescribe the displacement or rotation for each selected degree of freedom.



Note: If you apply the boundary condition to a two-dimensional planar region, only the **U1**, **U2**, and **UR3** degrees of freedom are available. If you model the part with generalized plane strain elements, you must use the **Keywords Editor** to apply boundary conditions on the active degrees of freedom (**U3**, **UR1**, and **UR2**) at the reference node of these elements (see *Adding unsupported keywords to your Abaqus/CAE model*).

1. Display the displacement/rotation boundary condition editor using one of the following methods:
   - To create a new displacement/rotation boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Displacement/Rotation`).
   - To edit an existing displacement/rotation boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If you are creating the boundary condition in a buckling step, select the **Use BC for** option that specifies the calculations for which you want the boundary condition used. For more information, see "Boundary conditions," in *Eigenvalue Buckling Prediction*.
3. By default, the global coordinate system is used to define the boundary condition. To change the coordinate system in which the boundary condition is applied, click  for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the boundary condition is created.
4. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify Constraints** if you want to specify values for particular degrees of freedom.
   - Select **Fixed at Current Position** if you want to fix the degrees of freedom at their final values from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
5. Click the arrow to the right of the **Distribution** field, if available, and do the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*

- *DISP*

- Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying boundary condition. Only analytical fields and discrete fields that are valid for this boundary condition type are displayed in the selection list.

  Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at Current Position** method.

6. If you selected the **Uniform**, analytical field, or discrete field distribution option, perform the following steps:
   a. Use the appropriate method to define the boundary condition:

If no text fields appear next to each degree of freedom:

- Toggle on a degree of freedom to fix the value either at zero (if you are defining a degree of freedom in the initial step) or at its final value at the end of the previous analysis step (if you are defining the degree of freedom in the second analysis step or in a later step.)
- Toggle off a degree of freedom to leave the degree of freedom unconstrained.

If text fields appear next to each degree of freedom:

- Toggle on a degree of freedom to constrain the degree of freedom. The text field becomes available in which you can specify a value for the degree of freedom. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.
- Toggle off a degree of freedom to leave the degree of freedom unconstrained. If you toggle off a degree of freedom after modifying the default or propagated value in the text field, the modified value is lost. If you toggle that degree of freedom back on, the default or propagated value reappears in the text field.
- If an **Amplitude** appears at the bottom of the editor, click the arrow to the right of the field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   b. Click **OK** to save your data and to exit the editor.

7. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, use the following techniques to define displacement:

   - Toggle on a degree of freedom to constrain the degree of freedom. If you are in the initial step, that degree of freedom is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the degree of freedom.
   - Toggle off a degree of freedom to leave the degree of freedom unconstrained.

   Data that you enter in the editor for particular degrees of freedom are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Using the Load module*
- *Using analytical expression fields*
- *Creating expression fields*
- *Creating discrete fields*
- *Boundary Conditions*

Defining a velocity/angular velocity boundary condition

You can create a velocity/angular velocity boundary condition to prescribe a velocity for the selected degrees of freedom of the selected region's nodes.



Note:

To fix the velocity of a region at zero you must define a specific displacement for that region (see *Defining a displacement/rotation boundary condition*).

1. Display the velocity/angular velocity boundary condition editor using one of the following methods:
   - To create a new velocity/angular velocity boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Velocity/Angular velocity`).
   - To edit an existing velocity/angular velocity boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. By default, the global coordinate system is used to define the boundary condition. To change the coordinate system in which the boundary condition is applied, click  for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the boundary condition is created.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. In addition, this option is unavailable if you are performing an Abaqus/Explicit analysis.

4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. Use the appropriate method to define the boundary condition:

If no text fields appear next to each degree of freedom:

You can use the following techniques to define the boundary condition:

- Toggle on a degree of freedom to fix the value at zero.
- Toggle off a degree of freedom to leave the degree of freedom unconstrained.

If text fields appear next to each degree of freedom:

You can use the following techniques to define the boundary condition:

- Toggle on a degree of freedom to constrain the degree of freedom. The text field becomes available in which you can specify a value for the degree of freedom. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.
- Toggle off a degree of freedom to leave the degree of freedom unconstrained. If you toggle off a degree of freedom after modifying the default or propagated value in the text field, the modified value is lost. If you toggle that degree of freedom back on, the default or propagated value reappears in the text field.
- Click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   b. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, use the following techniques to define the velocity:

   - Toggle on a degree of freedom to constrain the degree of freedom. If you are in the initial step, that degree of freedom is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the degree of freedom.
   - Toggle off a degree of freedom to leave the degree of freedom unconstrained.

   Data that you enter in the editor for specific degrees of freedom are passed into the user subroutine.

   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*

Defining an acceleration/angular acceleration boundary condition

You can create an acceleration/angular acceleration boundary condition to prescribe an acceleration for the selected degrees of freedom of the selected region's nodes.

Note:

To fix the acceleration of a region at zero you must define a specific displacement for that region (see *Defining a displacement/rotation boundary condition*).

1. Display the acceleration/angular acceleration boundary condition editor using one of the following methods:
   - To create a new acceleration/angular acceleration boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Acceleration/Angular acceleration`).
   - To edit an existing acceleration/angular acceleration boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. By default, the global coordinate system is used to define the boundary condition. To change the coordinate system in which the boundary condition is applied, click for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   This coordinate system editing option is available only in the step in which the boundary condition is created.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. In addition, this option is unavailable if you are performing an Abaqus/Explicit analysis.

4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. Use the appropriate method to define the boundary condition:

If no text fields appear next to each degree of freedom:

You can use the following techniques to define the boundary condition:

- Toggle on a degree of freedom to fix the value at zero.
- Toggle off a degree of freedom to leave the degree of freedom unconstrained.

If text fields appear next to each degree of freedom:

You can use the following techniques to define the boundary condition:

- Toggle on a degree of freedom to constrain the degree of freedom. The text field becomes available in which you can specify a value for the degree of freedom. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.
- Toggle off a degree of freedom to leave the degree of freedom unconstrained. If you toggle off a degree of freedom after modifying the default or propagated value in the text field, the modified value is lost. If you toggle that degree of freedom back on, the default or propagated value reappears in the text field.
- Click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   b. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, use the following techniques to define the acceleration:

   - Toggle on a degree of freedom to constrain the degree of freedom. If you are in the initial step, that degree of freedom is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the degree of freedom.
   - Toggle off a degree of freedom to leave the degree of freedom unconstrained.

   Data that you enter in the editor for specific degrees of freedom are passed into the user subroutine.

   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*

Defining a connector displacement boundary condition

You can create a connector displacement boundary condition to prescribe a displacement for the available components of relative motion of connectors. If the connector displacement boundary condition exceeds the failure criteria for a connector, the connector displacement will be ignored.

1. Display the connector displacement boundary condition editor using one of the following methods:
   - To create a new connector displacement boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Connector displacement`).
   - To edit an existing connector displacement/rotation boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If you are creating the boundary condition in a buckling step, select the **Use BC for** option that specifies the calculations for which you want the boundary condition used. For more information, see "Boundary conditions," in *Eigenvalue Buckling Prediction*.
3. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify Constraints** if you want to specify values for particular components of relative motion.
   - Select **Fixed at Current Position** if you want to fix the components of relative motion at their final values from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
4. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at Current Position** method or if you are performing an Abaqus/Explicit analysis.
5. If you are defining the boundary condition directly in the editor (and not in user subroutine *DISP*), perform the following steps:
   a. Use the appropriate method to define the boundary condition. Only components common to all of the selected wires are available in the editor.

If no text fields appear next to each component of relative motion:

- Toggle on a component of relative motion to fix the value either at zero (if you are defining a component of relative motion in the initial step) or at its final value at the end of the previous analysis step (if you are defining the component of relative motion in the second analysis step or in a later step.)

- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

If text fields appear next to each component of relative motion:

- Toggle on a component of relative motion to constrain the component of relative motion. The text field becomes available in which you can specify a value for the component of relative motion. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained. If you toggle off a component of relative motion after modifying the default or propagated value in the text field, the modified value is lost. If you toggle that component of relative motion back on, the default or propagated value reappears in the text field.
- Click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

b. Click **OK** to save your data and to exit the editor.

6. If you are defining the boundary condition in user subroutine *DISP*, perform the following steps:

a. If desired, use the following techniques to define displacement. Only components common to all of the selected wires are available in the editor.

- Toggle on a component of relative motion to constrain the component of relative motion. If you are in the initial step, that component of relative motion is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the component of relative motion.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

Data that you enter in the editor for particular degrees of freedom are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Connector Actuation*

Defining a connector velocity boundary condition

You can create a connector velocity boundary condition to prescribe a velocity for the available components of relative motion of connectors.

1. Display the connector velocity boundary condition editor using one of the following methods:
   - To create a new connector velocity boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Connector velocity`).
   - To edit an existing connector velocity boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.

2. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. In addition, this option is unavailable if you are performing an Abaqus/Explicit analysis.

3. If you are defining the boundary condition directly in the editor (and not in user subroutine *DISP*), perform the following steps:
   a. Use the appropriate method to define the boundary condition. Only components common to all of the selected wires are available in the editor.

If no text fields appear next to each component of relative motion:

You can use the following techniques to define the boundary condition:

- Toggle on a component of relative motion to fix the component of relative motion at zero.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

If text fields appear next to each component of relative motion:

You can use the following techniques to define the boundary condition:

- Toggle on a component of relative motion to constrain the component of relative motion. The text field becomes available in which you can specify a value for the component of relative motion. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.

- Toggle off a component of relative motion to leave the component of relative motion unconstrained. If you toggle off a component of relative motion after modifying the default or propagated value in the text field, the modified value is lost. If you toggle that component of relative motion back on, the default or propagated value reappears in the text field.
- Click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

b. Click **OK** to save your data and to exit the editor.

4. If you are defining the boundary condition in user subroutine *DISP*, perform the following steps:

a. If desired, use the following techniques to define the velocity. Only components common to all of the selected wires are available in the editor.

- Toggle on a component of relative motion to constrain the component of relative motion. If you are in the initial step, that component of relative motion is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the component of relative motion.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

Data that you enter in the editor for specific degrees of freedom are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Connector Actuation*

Defining a connector acceleration boundary condition

You can create a connector acceleration boundary condition to prescribe an acceleration for the available components of relative motion of connectors.

1. Display the connector acceleration boundary condition editor using one of the following methods:
   - To create a new connector acceleration boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Connector acceleration`).
   - To edit an existing connector acceleration boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.

2. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. In addition, this option is unavailable if you are performing an Abaqus/Explicit analysis.

3. If you are defining the boundary condition directly in the editor (and not in user subroutine *DISP*), perform the following steps:
   a. Use the appropriate method to define the boundary condition. Only components common to all of the selected wires are available in the editor.

If no text fields appear next to each component of relative motion:

You can use the following techniques to define the boundary condition:

- Toggle on a component of relative motion to fix the value at zero.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

If text fields appear next to each component of relative motion:

You can use the following techniques to define the boundary condition:

- Toggle on a component of relative motion to constrain the component of relative motion. The text field becomes available in which you can specify a value for the component of relative motion. If you are creating the boundary condition in this step, a default value of zero appears in the text field. If you are modifying the boundary condition in this step, the value propagated from the previous step appears in the text field.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained. If you toggle off a component of relative motion after modifying the

default or propagated value in the text field, the modified value is lost. If you toggle that component of relative motion back on, the default or propagated value reappears in the text field.

- Click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

b. Click **OK** to save your data and to exit the editor.

4. If you are defining the boundary condition in user subroutine *DISP*, perform the following steps:

a. If desired, use the following techniques to define the acceleration. Only components common to all of the selected wires are available in the editor.

- Toggle on a component of relative motion to constrain the component of relative motion. If you are in the initial step, that component of relative motion is set to zero. If you are in any step other than the initial step, a text field becomes available in which you can specify a value for the component of relative motion.
- Toggle off a component of relative motion to leave the component of relative motion unconstrained.

Data that you enter in the editor for specific degrees of freedom are passed into the user subroutine.

b. Click **OK** to save your data and to exit the editor.

c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Connector Actuation*

Defining a base motion boundary condition

You can create a base motion boundary condition to specify the motion of nodes in a modal dynamic step, steady-state dynamic (mode-based or subspace-based) step, or random response step. The base motion can be acceleration, velocity, or displacement.

For more information on base motion, see the boundary conditions discussions in the following sections:

- *Transient Modal Dynamic Analysis*
- *Mode-Based Steady-State Dynamic Analysis*
- *Subspace-Based Steady-State Dynamic Analysis*
- *Random Response Analysis*

1. Display the boundary condition editor using one of the following methods:
   - To create a new displacement base motion boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Acceleration base motion` or `Velocity base motion` or `Displacement base motion`).
   - To edit an existing base motion boundary condition using menus or managers, see *Editing step-dependent objects*.
2. In the **Basic** tab, select the **Degree-of-freedom** to specify the direction for which the base motion is being defined. This direction is always a global direction: **U1**, **U2**, **U3**, **UR1**, **UR2**, or **UR3**.
3. For a modal dynamic step or steady-state dynamic (mode-based or subspace-based) step, specify the following options:
   a. If the base motion is to be applied to a secondary base, toggle on **Secondary base** and choose the name of the secondary base boundary condition defined previously in the eigenfrequency extraction step.
   b. Choose an **Amplitude** curve that defines the time history (modal dynamic step) or frequency spectrum (steady-state dynamic step) of the base motion.
   c. Enter a new value for the **Amplitude scale factor**, if desired. The default value is 1.0.
   d. Base motions are always specified in global directions, regardless of the use of nodal transformations. You specify the global direction (**Degree-of-freedom**) for which the base motion is being defined. If a rotation is specified about an origin that is not the origin of the coordinates, you must specify the **Center of rotation** by clicking  and picking a point in the viewport or entering the coordinates.
   e. For a steady-state dynamic step, you can toggle on **Define imaginary (out-of-phase) portion given by amplitude** if the specified amplitude curve defines the imaginary part of the base motion (instead of the real, in-phase portion).
4. For a random response step, click the **Correlation** tab and toggle on **Specify correlation** to define the cross-correlation as part of the definition of random loading. See *Defining the Correlation*, for complete information about correlation options. To define a correlation, fill out the cells in a row of the table as follows:
   a. Click in the **Approach** column, and select one of the following approaches:
      - Select **Correlated** if all terms in the correlation matrix should be included.
      - Select **Uncorrelated** if only diagonal terms should be used.

- Select **User** to indicate that user subroutine *UCORR* will be called to obtain the scaling factors for the correlation matrix.

b. Click in the **PSD** column cell, and select the PSD definition amplitude that you defined previously. This amplitude represents the power spectral density function for the random noise source.

c. In the **Real** and **Imaginary** columns, enter the real and imaginary parts of the scaling factor. These scaling factors specify whether the real and imaginary terms will be included in the spatial correlation matrix.

You can add rows in the table to create additional correlations to define the random loading.

5. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*

Defining a secondary base motion boundary condition

You can create a secondary base motion boundary condition in a frequency extraction step. If a subsequent modal dynamic or steady-state dynamic (mode-based or subspace-based) step has more than one independent base motion, the driven nodes must be grouped together into secondary bases in addition to the primary base. See *Defining Primary and Secondary Bases for Modal Superposition Procedures*, for a complete discussion of base motion boundary conditions.

1. Display the boundary condition editor using one of the following methods:
   - To create a new secondary base boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Secondary base`).
   - To edit an existing boundary condition using menus or managers, see *Editing step-dependent objects*.
2. Select the region to which you want to apply the boundary condition. Double-click in the empty table cell under the **Region** column and use one of the following methods:
   - Select a region in the viewport. You can use the angle method to select a group of faces or edges from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.

     Tip: You can limit the types of objects that you can select in the viewport by specifying filtering options in the Selection toolbar. See *Using the selection options*, for more information.

     If the model contains a combination of mesh and geometry, you must choose the type of region to which you want to apply the boundary condition. From the prompt area, select one of the following:
     - Click **Geometry** to apply the boundary condition to geometry or to a reference point.
     - Click **Mesh** to apply the boundary condition to a native or orphan mesh selection.
   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

        Note:

        The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
3. Select the **Constrained-degrees-of-freedom** in the table by clicking one or more checkboxes.
4. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*

Defining a retained nodal DOFs boundary condition

You can create a retained nodal degrees of freedom (DOFs) boundary condition to specify the degrees of freedom that are to be retained as external to a substructure. This boundary condition can be created only in a **Substructure generation** step.

1. Display the retained nodal DOFs boundary condition editor using one of the following methods:
   - To create a new retained nodal DOFs boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Mechanical`; **Types for Selected Step**: `Retained nodal dofs`).
   - To edit an existing retained nodal DOFs boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. Toggle on any of the following options:

U1

Nodal freedom in the 1-direction.

U2

Nodal freedom in the 2-direction.

U3

Nodal freedom in the 3-direction.

UR1

Rotation about the 1-axis.

UR2

Rotation about the 2-axis.

UR3

Rotation about the 3-axis.

3. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Configuring a substructure generation procedure*
- *Defining the Retained Nodal Degrees of Freedom*

Specifying temperature

You can create a temperature boundary condition to specify a temperature for the selected region.

1. Display the temperature boundary condition editor using one of the following methods:
   - To create a new temperature boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Temperature`).
   - To edit an existing temperature boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the temperature.
   - Select **Fixed at current magnitude** if you want to fix the temperature magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the temperature magnitude in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

      Note:

      The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous analysis step.

5. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the temperature magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

      (The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)
   b. Click **OK** to save any entered data and to exit the editor.
   c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.

      Note:

      You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

6. If a shell region is selected, enter the temperature degrees of freedom that you want to specify in the **Degrees of freedom** field. For information on how shell temperature degrees of freedom are labeled, see either "Heat transfer shell elements" or "Coupled temperature-displacement shell elements" in *Choosing a Shell Element*.

   You can enter multiple degrees of freedom, from 11 to 31, inclusive, by separating each number with a comma. For example, if you want to specify degrees of freedom 11, 12, and 13, you must enter the following:`11,12,13`
7. Click **OK** to save your data and to exit the editor.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Shell Elements*
- *General-Purpose Continuum Elements*
- *Boundary Conditions*

Defining a pore pressure boundary condition

You can create a pore pressure boundary condition to specify the excess or total pore pressure in a porous medium. If no gravity load exists in the model, you must specify excess pore pressure. If the model contains any gravity load, you must specify total pore pressure. (For more information, see "Boundary conditions," in *Coupled Pore Fluid Diffusion and Stress Analysis*.)

1. Display the pore pressure boundary condition editor using one of the following methods:
   - To create a new pore pressure boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Pore pressure`).
   - To edit an existing pore pressure boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the excess or total pore pressure.
   - Select **Fixed at current magnitude** if you want to fix the excess or total pore pressure magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click  to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the magnitude of the excess or total pore pressure in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)



Note:

The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous general step.

c. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the pore pressure magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

(The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)

b. Click **OK** to save any entered data and to exit the editor.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining a fluid cavity pressure boundary condition

You can create a fluid cavity pressure boundary condition to control the pressure in a cavity throughout the analysis. The fluid cavity pressure is associated with a fluid cavity interaction. It is used to determine flow for a fluid exchange interaction. You can change the fluid cavity pressure throughout the analysis by assigning a new value for the step and an amplitude to define the rate of change from the previous value. (For more information, see *Initial Conditions*.)

1. Display the fluid cavity pressure boundary condition editor using one of the following methods:
   - To create a new fluid cavity pressure boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Fluid cavity pressure`).
   - To edit an existing fluid cavity pressure boundary condition using menus or managers, see *Editing step-dependent objects*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify pressure** if you want to specify a value for the fluid cavity pressure.
   - Select **Fixed at current pressure** if you want to fix the fluid cavity pressure magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Fluid cavity interaction** field appears, click the arrow to the right of the field, and select the interaction to which the pressure applies.

   The **Fluid cavity interaction** option appears in the editor only if you are creating a new boundary condition or if you are editing the fluid pressure boundary condition in the step in which it was created. The interaction cannot be changed when editing the boundary condition in a propagated step.
4. If a **Fluid cavity pressure** field appears, enter a value for the current step.
5. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Note:

   The **Amplitude** field is unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous general step (such as when editing a frequency step).
6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Using the Load module*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining an electric potential boundary condition

You can create an electric potential boundary condition to prescribe an electrical potential on a region. (For more information, see "Boundary conditions," in *Coupled Thermal-Electrical Analysis* and *Piezoelectric Analysis*.)

1. Display the electric potential boundary condition editor using one of the following methods:
   - To create a new electric potential boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Electric potential`).
   - To edit an existing electric potential boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the electric potential.
   - Select **Fixed at current magnitude** if you want to fix the electric potential magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the magnitude of the electric potential in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Note:

   The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous general step.

c. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the electric potential magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

(The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)

b. Click **OK** to save any entered data and to exit the editor.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*
- *Coupled Thermal-Electrical Analysis*

Defining a magnetic vector potential boundary condition

You can create a magnetic vector potential boundary condition to prescribe magnetic vector potential on a surface. (For more information, see *Boundary Conditions*.) The magnetic vector potential boundary condition is available only in an electromagnetic model.

1. Display the magnetic vector potential boundary condition editor using one of the following methods:
   - To create a new magnetic vector potential boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Electrical/Magnetic`; **Types for Selected Step**: `Magnetic vector potential`).
   - To edit an existing magnetic vector potential boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a boundary condition that is uniform over the surface.
   - Select **User-defined** to define the magnitude and direction of the boundary condition in user subroutine *UDEMPOTENTIAL*. See the following sections for more information:
     - *Specifying general job settings*
     - *UDEMPOTENTIAL*

3. If you selected the **Uniform** distribution option, perform the following steps:
   a. In the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields, enter the real (in-phase) and imaginary (out-of-phase) parts of the magnetic vector potential.

      Abaqus/CAE calculates the magnitude and direction of the magnetic vector potential.
   b. If desired, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)
   c. Click **OK** to save your data and to exit the editor.

4. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, in the **Component 1**, **Component 2**, and (if applicable) **Component 3** fields enter the real (in-phase) and imaginary (out-of-phase) parts of the magnetic vector potential.

      Abaqus/CAE calculates the magnitude and direction of the magnetic vector potential and that information is passed into the user subroutine.
   b. Click **OK** to save your data and to exit the editor.
   c. Enter the Job module and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the magnitude and direction of the boundary condition. For more information, see *Specifying general job settings*.



Note: You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Eddy Current Analysis*

Defining a normalized concentration boundary condition

You can create a normalized concentration boundary condition to prescribe values of normalized concentration for a region in a mass diffusion analysis. (For more information, see Boundary conditions in *Mass Diffusion Analysis*.)

1. Display the normalized concentration boundary condition editor using one of the following methods:
   - To create a new normalized concentration boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Normalized concentration`).
   - To edit an existing normalized concentration boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the normalized concentration.
   - Select **Fixed at current magnitude** if you want to fix the normalized concentration magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the normalized concentration magnitude in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

   Note:

   The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at the final value from the previous general step.

c. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the normalized concentration magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

(The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)

b. Click **OK** to save any entered data and to exit the editor.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Using the Load module*
- *Creating and modifying prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*
- *Mass Diffusion Analysis*

Defining an acoustic pressure boundary condition

You can create an acoustic pressure boundary condition to specify the pressure for the selected region of an acoustic medium. (For more information, see Boundary Conditions in *Coupled Acoustic-Structural Analysis*.)

1. Display the acoustic pressure boundary condition editor using one of the following methods:
   - To create a new acoustic pressure boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Acoustic pressure`).
   - To edit an existing acoustic pressure boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.
2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the acoustic pressure.
   - Select **Fixed at current magnitude** if you want to fix the acoustic pressure magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.
3. If a **Distribution** field appears in the editor, click the arrow to the right of the field, and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP*. See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.
4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the magnitude of the acoustic pressure in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)



Note:

The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous analysis step.

c. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:

a. If desired, enter the acoustic pressure magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

(The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)

b. Click **OK** to save any entered data and to exit the editor.

c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Using the Load module*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*
- *Coupled Acoustic-Structural Analysis*

Defining a connector material flow boundary condition

You can create a connector material flow boundary condition to constrain the material flow at the endpoints of a connector.

1. Display the connector material flow boundary condition editor using one of the following methods:
   - To create a new connector material flow boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Connector material flow`).
   - To edit an existing connector material flow boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.

2. If a **Method** field appears toward the top of the editor, click the arrow to the right of the field and select one of the following:
   - Select **Specify magnitude** if you want to specify a value for the connector material flow.
   - Select **Fixed at current magnitude** if you want to fix the connector material flow magnitude at its final value from the previous general step.

   The **Method** option appears in the editor only if both methods are valid for the selected step.

3. If a **Distribution** field appears in the editor, click the arrow to the right of the field and select one of the following:
   - Select **Uniform** to define a uniform boundary condition.
   - Select **User-defined** to define the boundary condition in user subroutine *DISP* (for Abaqus/Standard). See the following sections for more information:
     - *Specifying general job settings*
     - *DISP*
   - Select an analytical field to define a spatially varying boundary condition. Only analytical fields that are valid for this boundary condition type are displayed in the selection list. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

   The **Distribution** option appears in the editor only if you are creating the boundary condition or editing the boundary condition in the step in which it was created. This option is unavailable if you select the **Fixed at current magnitude** method.

4. If you selected the **Uniform** or analytical field distribution option, perform the following steps:
   a. If applicable, enter the magnitude of the connector material flow in the **Magnitude** field.
   b. If applicable, click the arrow to the right of the **Amplitude** field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.)

      Note:

      The **Magnitude** and **Amplitude** fields are unavailable if you are creating the boundary condition in the initial step (in which case the magnitude is always zero) or are fixing the magnitude at its final value from the previous general step.

c. Click **OK** to save your data and to exit the editor.

5. If you selected the **User-defined** distribution option, perform the following steps:
   a. If desired, enter the connector material flow magnitude in the **Magnitude** field. Magnitude data that you enter in the editor are passed into the user subroutine.

      (The **Magnitude** field is unavailable if you are creating the boundary condition in the initial step.)
   b. Click **OK** to save any entered data and to exit the editor.
   c. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   d. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the boundary condition. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

Additional information

- *Creating and modifying prescribed conditions*
- *Using the Load module*
- *Using analytical expression fields*
- *Creating expression fields*
- *Boundary Conditions*

Defining an Eulerian boundary condition

You can create an Eulerian boundary condition to control the flow of materials into and out of an Eulerian region. Eulerian boundary conditions should be applied only to parts with an Eulerian section assignment, and they cannot be applied in the initial step of an analysis.

1. Display the Eulerian boundary condition editor using one of the following methods:
   - To create a new Eulerian boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Eulerian boundary`).
   - To edit an existing Eulerian boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the region to which the boundary condition is applied, see *Editing the region to which a prescribed condition is applied*.

2. Specify the **Flow type** that the boundary condition will control:
   - **Inflow** controls the flow of Eulerian material into the region across the selected surface.
   - **Outflow** controls the flow of Eulerian material out of the region across the selected surface.
   - **Both** allows you to specify controls for inflow and outflow conditions in the same boundary condition definition.

3. If you selected **Inflow** (or **Both**) as the **Flow type**, select the **Inflow** condition:
   - Select **Free** to allow any new material to flow freely into the Eulerian region. The material that flows into the region is based on the material that currently occupies the region: as Eulerian material inside the region moves away from the boundary surface, the same material flows across the boundary to replace it.
   - Select **None** to prevent inflow of any new Eulerian material (including void) across the boundary surface. Eulerian material inside the region can move tangentially along the boundary surface, but it cannot separate from the boundary surface.
   - Select **Void** to allow only void to flow into the Eulerian region. As Eulerian material inside the region moves away from the boundary surface, void flows across the boundary to replace it.

4. If you selected **Outflow** (or **Both**) as the **Flow type**, select the **Outflow** condition:
   - Select **Zero pressure** to allow material to flow out of the Eulerian region while also enforcing a condition of zero net pressure across the boundary surface using artificial traction forces.
   - Select **Free** to allow material to flow freely out of the Eulerian region without any artificial traction forces.
   - Select **Nonreflecting** to allow material to flow out of the Eulerian region while also minimizing the reflection of dilatational and shear wave energy back into the region using artificial traction forces.
   - Select **Equilibrium** to allow material to flow out of the Eulerian region while also enforcing a condition of zero-order continuous stress across the boundary surface using artificial traction forces.

Additional information

- *Creating and modifying prescribed conditions*
- *Using the Load module*
- *Defining Eulerian Boundaries*

Defining an Eulerian mesh motion boundary condition

Eulerian mesh motion allows an Eulerian mesh—which is usually rigid and undeformable—to expand, contract, and translate during an analysis. As part of the mesh motion definition you identify an object to follow, which can be either a Lagrangian surface or an Eulerian material instance; the Eulerian mesh resizes and moves during the analysis so that it always surrounds this object.

Eulerian mesh motion is calculated based on a rectangular "bounding box" that is constructed by Abaqus. The bounding box cannot be visualized in Abaqus/CAE, but it typically corresponds relatively closely to the Eulerian mesh. The creation and behavior of the bounding box is described in *Eulerian Mesh Motion*. You can set restraints and restrictions on the contraction, expansion, and translation of this bounding box as part of the mesh motion boundary condition definition. These restrictions may prevent the Eulerian mesh from completely encompassing the specified object to follow at all times during the analysis.

Eulerian mesh motion boundary conditions can be applied only to parts with an Eulerian section assignment, and they cannot be applied in the initial step of an analysis.

1. Display the Eulerian mesh motion boundary condition editor using one of the following methods:
   - To create a new Eulerian mesh motion boundary condition, follow the procedure outlined in *Creating boundary conditions* (**Category**: `Other`; **Types for Selected Step**: `Eulerian mesh motion`).
   - To edit an existing Eulerian mesh motion boundary condition using menus or managers, see *Editing step-dependent objects*. To edit the part instance to which the mesh motion definition applies, see *Editing the region to which a prescribed condition is applied*.

2. Specify the **Object to Follow**:
   - To follow a Lagrangian surface, choose **Surface region** and click  to select a surface:
     - Select a surface in the viewport. You can use the angle method to select a group of faces from geometry or a group of element faces from a mesh. For more information, see *Using the angle and feature edge method to select multiple objects*. When you have finished selecting, click mouse button 2.
     - To select from a list of existing surfaces, do the following:
       1. Click **Surfaces** on the right side of the prompt area.

          Abaqus/CAE displays the **Region Selection** dialog box containing a list of available surfaces.
       2. Select the surface of interest, and click **Continue**.

           **Note:**

          The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

   - To follow an Eulerian material, choose **Eulerian material instance** and select an instance.

3. By default, the global coordinate system is used to construct the mesh bounding box. To change the coordinate system in which the mesh bounding box is constructed, click  for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.

- Select an existing datum coordinate system by name.

  1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
  2. Select a name from the list, and click **OK**.

- Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

Alternatively, click  to define a new datum coordinate system.

This coordinate system editing option is available only in the step in which the boundary condition is created.

4. On the **Controls** tabbed page, specify general constraints and limitations on the motion of the mesh bounding box:
   a. To prevent the mesh from contracting during the analysis, toggle off **Allow mesh contraction**. This option is toggled on by default.
   b. Specify an **Aspect ratio limit** for the maximum change allowed in the length to width ratio in any of the three bounding box directions. The value must be greater than or equal to one.
   c. Specify the **Mesh velocity factor**. This factor defines the maximum velocity of the Eulerian mesh as a fraction of the velocity of the object being followed. The value must be greater than or equal to zero.
   d. If the object being followed is an Eulerian material, specify a **Volume fraction threshold**. Elements whose Eulerian material volume fraction are less than this threshold are considered empty for the purposes of mesh motion, and the mesh does not expand or move to account for the material in these elements. The value must be greater than zero and less than or equal to one.
   e. Specify the **Buffer size**, which preserves some distance between the object being tracked and the Eulerian mesh boundary:

      - Select **Specify**, and enter a value to define a buffer equal to the given value times the maximum Eulerian element size. The value must be greater than or equal to zero.
      - Select **Initial** to define a buffer equal to the initial distance between the object and the mesh boundary.

5. On each of the **Axis** tabbed pages, specify limits on the motion of the bounding box in each of its three directions:
   a. Choose whether the original **Center position** is **Free** or **Fixed** along the selected axis.
   b. Choose whether the original **Positive plane position** is **Free** or **Fixed**. The positive plane is the face of the bounding box whose normal is aligned with the positive direction of the selected axis.
   c. Choose whether the original **Negative plane position** is **Free** or **Fixed**. The negative plane is the face of the bounding box whose normal is aligned with the negative direction of the selected axis.
   d. Specify the **Expansion ratio** along the selected axis:

      - Choose **Unlimited** to allow infinite scaling of the bounding box if necessary.
      - Choose **Specify maximum**, and enter a value to define a maximum scaling ratio. The value must be greater than or equal to one.

   e. Specify the minimum **Contraction ratio** along the selected axis. The value must be greater than or equal to zero and less than or equal to one.

- *Creating and modifying prescribed conditions*
- *Using the Load module*
- *Eulerian analyses*”

- *Eulerian Mesh Motion*

Using the predefined field editors

This section explains how to define or initialize specific types of predefined fields.

In this section:

- *Defining an initial velocity field*
- *Defining a hardening field*
- *Defining an initial stress*
- *Defining a geostatic initial stress field*
- *Defining a temperature field*
- *Defining values for predefined field variables*
- *Defining a material assignment field*
- *Defining an initial state field*
- *Defining a saturation field*
- *Defining an initial void ratio field*
- *Defining a pore pressure field in a porous medium*
- *Defining a fluid cavity pressure field*

Defining an initial velocity field

You can create this type of predefined field to define the initial velocity of the selected region. For more information, see *Initial Conditions*.

1. Display the velocity field editor using one of the following methods:
   - To create a new velocity field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Mechanical`; **Types for Selected Step**: `Velocity`).
   - To edit an existing velocity field using menus or managers, see *Editing step-dependent objects*.

   Note:

   You can create or edit an initial velocity only in the initial step.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define an initial velocity that is uniform over the region.
   - Select an analytical field to define a spatially varying initial velocity. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. Click the arrow to the right of the **Definition** field, and select the option of your choice from the list that appears:
   - Select **Translational only** to define an initial translational velocity only.
   - Select **Rotational only** to define an initial rotational velocity only.
   - Select **Translational & rotational** to define both an initial translational velocity and an initial rotational velocity.

4. If you selected the **Translational only** definition option, enter the magnitudes for the initial translational velocities in the 1-, 2-, and (if you are working in three-dimensional space) 3-directions in the **V1**, **V2**, and (if applicable) **V3** text fields. (If you leave a text field blank, an initial translational velocity of zero is assigned to that degree of freedom by default; the velocity must be defined in at least one direction.)
5. If you selected the **Rotational only** definition option:
   a. Enter the magnitude of the angular velocity (in radians/time) in the **Angular velocity** text field.
   b. Define the axis of rotation.
      - For three-dimensional models, enter the *x*-, *y*-, and *z*-coordinates, separated by commas, of the points in the **Axis point 1** and **Axis point 2** text fields.
      - For two-dimensional models, enter the *x*- and *y*-coordinates, separated by a comma, of the point in the **Axis point** text field. Rotation is assumed to be about the positive *z*-axis at the specified point.
      - For axisymmetric models, enter the radius in the **Axis radius** text field. Rotation is assumed to be about the positive *z*-axis at the specified radius.

6. If you selected the **Translational & rotational** definition option, follow the instructions in both Steps 3 and 4.

7. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating predefined fields*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Initial Conditions*

Defining a hardening field

You can create a hardening field to define initial values of state variables for plastic hardening. For more information, see *Initial Conditions*.

1. Display the hardening field editor using one of the following methods:
   - To create a new hardening field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Mechanical`; **Types for Selected Step**: `Hardening`).
   - To edit an existing hardening field using menus or managers, see *Editing step-dependent objects*.

   Note:

   You can create or edit a hardening field only in the initial step.

2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define initial values that are uniform over the region.
   - Select an analytical field to define spatially varying initial values. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. Click the arrow to the right of the **Definition** field, and select the option of your choice from the list that appears:
   - Select **Kinematic hardening** to specify an initial equivalent plastic strain and the initial backstress tensors for kinematic hardening material models.
   - Select **Crushable foam** to specify an initial volumetric compacting plastic strain for the crushable foam material model.
   - Select **Rebar** to specify an initial equivalent plastic strain and the initial backstress tensors for rebar layers. (This option is valid only for Abaqus/Standard analyses.)
   - Select **Section points** to specify an initial equivalent plastic strain and the initial backstress tensors at section points through the thickness of shell elements.
   - Select **User-defined** to specify the initial equivalent plastic strain and, if relevant, the initial backstress tensors in user subroutine *HARDINI*. (This option is valid only in Abaqus/Standard analyses.) See the following sections for more information:
     - *Specifying general job settings*
     - *HARDINI*

4. If applicable, click the arrows to the right of the **Number of backstresses** field to specify the number of backstresses to include in the model. The default number of backstresses is 1. The maximum number of backstresses allowed is 10.

   Additional columns appear in the **Data** table to specify multiple backstress tensors.

5. In the **Data** table, enter the data relevant to your **Definition** selection (not all of the following parameters will apply):

Equiv Plast Strain

Initial equivalent plastic strain, $\bar{\varepsilon}^{pl}\big|_0$.

Vol Plast Strain

Initial volumetric compacting plastic strain, $-\varepsilon_{\text{vol}}^{pl}$.

Rebar Layer Name

Name of rebar layer.

Section Point Number

Section point number.

Alpha 1_11, Alpha 1_22, etc.

Values for each component of the initial first backstress tensor, $\boldsymbol{\alpha}\mathbf{1}^0$.

Alpha k_11, Alpha k_22, etc.

Values for each component of the initial k^{th} backstress tensor, $\boldsymbol{\alpha}\boldsymbol{k}^0$.

6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating predefined fields*
- *Understanding symbols that represent prescribed conditions*
- *Using analytical expression fields*
- *Creating expression fields*
- *Initial Conditions*

Defining an initial stress

You can create an initial stress field for a region of your model. You can create or modify an initial stress field only in the initial step. For more information, see *Initial Conditions*.

1. Display the stress field editor using one of the following methods:
   - To create a new stress field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Mechanical`; **Types for Selected Step**: `Stress`).
   - To edit an existing stress field using menus or managers, see *Editing step-dependent objects*.

   Note:

   You can create or edit a stress field only in the initial step.

2. Click the arrow to the right of the **Specification** field, and select the option of your choice from the list that appears:
   - Select **Direct specification**, then specify stress components in the data table. You can specify up to six component values.
   - Select **From output database file** to specify stress values from a user-specified output database file.

3. If you are specifying initial stress from an output database file, do the following:
   a. Specify the file name from which you want to import stress values.
   b. From the **Step** and **Increment** fields, specify the step name and increment number from which you want to import stress values.
4. Click **OK** to create the initial stress field and to close the dialog box.

   Blue circles appear on the model representing the initial stress field that you just created.

Additional information

- *Creating predefined fields*
- *Understanding symbols that represent prescribed conditions*
- *Initial Conditions*

Defining a geostatic initial stress field

When a geostatic stress field is prescribed for a particular region, the stress in the vertical direction (assumed to be the *z*-direction in three-dimensional and axisymmetric models and the *y*-direction in two-dimensional models) is assumed to vary (piecewise) linearly with this vertical coordinate. Geostatic stress fields are applicable only to continuum elements. For more information, see *Initial Conditions*. For Abaqus/Explicit analyses, elevation-dependent initial stresses cannot be specified for beams and shells.

1. Display the geostatic stress field editor using one of the following methods:
   - To create a new geostatic stress field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Mechanical`; **Types for Selected Step**: `Geostatic stress`).
   - To edit an existing geostatic stress field using menus or managers, see *Editing step-dependent objects*.

   Note:

   You can create or edit a geostatic stress field only in the initial step.

2. In the **Stress magnitude 1** field, specify the vertical component of (effective) stress at the first location in the region.
3. In the **Vertical coordinate 1** field, specify the vertical coordinate corresponding to the first location in the region.
4. In the **Stress magnitude 2** field, specify the vertical component of (effective) stress at the second location in the region.
5. In the **Vertical coordinate 2** field, specify the vertical coordinate corresponding to the second location in the region.
6. In the **Lateral coefficient 1** field, specify the first coefficient of lateral stress. This coefficient defines the *x*-direction stress component.
7. If desired, toggle on **Specify** for the **Lateral coefficient 2** setting and specify the second coefficient of lateral stress. This coefficient defines the *y*-direction stress component in three-dimensional cases and the thickness or hoop direction component in plane or axisymmetric cases. If this value is omitted, it is assumed to be the same as the first lateral stress coefficient.
8. Click **OK** to save your data and to exit the editor.

   Light blue circles appear on the model representing the initial geostatic stress field that you just created.

Additional information

- *Creating predefined fields*
- *Understanding symbols that represent prescribed conditions*
- *Initial Conditions*

Defining a temperature field

You can define the magnitude and time variation of the temperature of the selected region for each analysis step; Abaqus will interpolate the temperatures to the material points. Alternatively, you can use the temperature values that Abaqus computed during a previous analysis with thermal components to define the temperatures in the current model.

1. Display the temperature field editor using one of the following methods:
   - To create a new temperature field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Temperature`).

     Note:

     If you want to specify the temperature variation directly in the temperature field editor, in user subroutine *UTEMP*, or using an analytical field or discrete field, you must select a region for the temperature field. If you want to specify any other temperature distribution, you may click **Done** in the prompt area to use a calculated temperature.

   - To edit an existing temperature field using menus or managers, see *Editing step-dependent objects*.

2. If you are editing a temperature field in any general analysis step other than the one in which the field was created, click the arrow to the right of the **Status** field, and select the option of your choice from the list that appears:
   - Select **Propagated** to indicate that the temperature field is active in the current step with temperatures values held constant from the previous step. No data can be specified; click **OK** to exit the editor. If you change the status of the current step from **Modified** to **Propagated** and click **OK**, all modifications to the current step are removed.
   - Select **Modified** to edit the field definition for the current step. Continue with Step 3.
   - Select **Reset to initial** to reset the field definition to the one that was prescribed in the initial step. No data can be specified; click **OK** to reset the field and to exit the editor.

3. If you are creating a new temperature field, click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears.

   Note:

   If you are creating a temperature field in the initial step, only the **Direct specification**, **From results or output database file**, analytical field, and discrete field distribution options are available.

   - Select **Direct specification** to specify the temperature variation over the selected region in the temperature field editor.
   - Select **From results or output database file** to read temperature values from the results or output database file of a previous Abaqus analysis with thermal components.
   - Select **User-defined** to define temperature values in user subroutine *UTEMP*. This option is available only for Abaqus/Standard analyses. See the following sections for more information:
     - *Specifying general job settings*
     - *UTEMP*

- Select **From results or output database file and user-defined** to read temperature values from the results or output database file of a previous Abaqus analysis with thermal components and modify them in user subroutine *UTEMP*. This option is available only for Abaqus/Standard analyses.
- Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying temperature. The selection list contains all analytical fields and only discrete fields that are valid for temperature fields.

  Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

4. If you selected the **Direct specification**, analytical field, or discrete field distribution option, perform the following steps:
   a. Click the arrow to the right of the **Section variation** field, and select an option from the list that appears. Only options that are valid for the selected region appear in the list. Beam and shell gradient values are not affected by an analytical field or a discrete field. You should select an option that is consistent with the temperature variation method defined in the section definition associated with the region selected for this field as follows:
      - Select **Constant through region** to define a constant temperature over a section.

        In the **Magnitude** text field, enter the magnitude of the temperature across the section.
      - Select **Gradient through shell section** to define a temperature variation through a shell section. The **Temperature variation** method in the shell section editor must specify **Linear through thickness**.

        In the **Reference magnitude** text field, enter the magnitude of the temperature at the reference surface. In the **Thickness gradient** text field, enter the temperature gradient through the section.
      - Select **Gradient through beam section** to define a temperature variation through a beam section. (This method is not available for axisymmetric models.) The **Temperature variation** method in the beam section editor must specify **Linear by gradients**.

        In the **Reference magnitude** text field, enter the magnitude of the temperature at the cross-section origin. In the **N1 gradient** and **N2 gradient** text fields, enter the temperature gradients through the section in the $\mathbf{n_1}$- and $\mathbf{n_2}$-directions of the beam, respectively. For beams in a plane only the temperature magnitude and the **N2 gradient** must be specified.
      - Select **Defined at shell/beam temperature points** to define a piecewise linear temperature variation through a shell or beam section. The **Temperature variation** method in the shell section editor must specify **Piecewise linear over *n* values**. The **Temperature variation** method in the beam section editor must specify **Interpolated from temperature points**.

        Select the number of temperature points to be specified by either typing an integer in the **Temperature points** text field or using the arrows to the right of the text field to select a number, and enter the magnitude of the temperature at each point in the **Section Data** table. The number of values in the section definition should be less than or equal to the number of temperature data points given for this field. If the number of values is less, the last value will be repeated to match the number of temperature data points.

   b. If desired (for all steps other than the initial step), click the arrow to the right of the **Amplitude** text field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.) Beam and shell gradient values will be modified by the amplitude definition.
   c. Click **OK** to save your data and to exit the editor.

5. If you selected the **From results or output database file** distribution option, perform the following steps:
   a. In the **File name** text field, enter the name of the results or output database file from which temperature data are to be read; or click  to display the **Select Results or Output Database File** dialog box and select the file of your choice. (See *Using file selection dialog boxes* for more information.)
   b. If you are creating a temperature field in the initial step:
      - In the **Step** text field, enter the step number of the analysis whose results or output database file is being used as input from which the field data should be read.
      - In the **Increment** text field, enter the increment number of the analysis whose results or output database file is being used as input from which the field data should be read.
   c. Optionally, if you are creating a temperature field in an analysis step other than the initial step:
      - In the **Begin step** text field, enter the step number of the analysis whose results or output database file is being used as input that begins the field data to be read. By default, the first step available in the results or output database file will be used.
      - In the **Begin increment** text field, enter the increment number of the analysis whose results or output database file is being used as input that begins the field data to be read. By default, the first increment available in the results or output database file will be used.
      - In the **End step** text field, enter the step number of the analysis whose results or output database file is being used as input that ends the field data to be read. By default, the **Begin step** value will be used.
      - In the **End increment** text field, enter the increment number of the analysis whose results or output database file is being used as input that ends the field data to be read. By default, the last increment available for the **End step** value in the results or output database file will be used.
   d. Choose the **Mesh compatibility**.
      - Select **Compatible** if the meshes in the original analysis and the current analysis are the same or differ only in the element order.

        If the original and current meshes differ only in the element order, you must indicate that midside node temperatures in second-order elements are to be interpolated from corner node temperatures by toggling on **Interpolate midside nodes**.
      - Select **Incompatible** if the meshes in the original analysis and the current analysis are dissimilar. This option is valid only if the temperature values are being read from an output database file.
   e. If the original and current meshes are dissimilar, you can specify a value for the **Exterior tolerance**.
      - Specify the **absolute** value by which nodes of the current model may lie outside the region of the elements of the original model. If this value is not specified or is 0.0, the **relative** tolerance will apply.
      - Specify the **relative** fraction of the average element size by which nodes of the current model may lie outside the region of the elements of the original model.

      If both tolerances are specified, Abaqus uses the tighter tolerance.
   f. Click **OK** to save your data and to exit the editor.
6. If you selected the **User-defined** distribution option, perform the following steps:

a. Click **OK** to exit the editor.

b. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

c. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the temperature field. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

7. If you selected the **From results or output database file and user-defined** distribution option, follow the procedures outlined in Steps 5 and 6 for the **From results or output database file** and **User-defined** distribution options.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*
- *Creating discrete fields*
- *Predefined Temperature*

Defining values for predefined field variables

You can define the initial value and the magnitude and time variation of a field variable over a selected region for each analysis step; Abaqus interpolates the values to the material points.

1. Display the predefined field variable editor using one of the following methods:
   - To create a new predefined field variable, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Field`).

     Note:

     If you want to specify the field variable variation directly in the editor, in user subroutine *UFIELD* or *VUFIELD*, or using an analytical field or discrete field, you must select a region for the field variable. If you want to specify any other field distribution, you can click **Done** in the prompt area to use calculated fields.

   - To edit an existing field variable using menus or managers, see *Editing step-dependent objects*.

2. If you are editing a field variable in any general analysis step other than the one in which the field variable was created, click the arrow to the right of the **Status** field, and select the option of your choice from the list that appears:
   - Select **Propagated** to indicate that the field variable is active in the current step with field variable values held constant from the previous step. No data can be specified; click **OK** to exit the editor. If you change the status of the current step from **Modified** to **Propagated** and click **OK**, all modifications to the current step are removed.
   - Select **Modified** to edit the field variable definition for the current step. Continue with Step 3.
   - Select **Reset to initial** to reset the field variable definition to the one that was prescribed in the initial step. No data can be specified; click **OK** to reset the field and to exit the editor.

3. If you are creating a new field variable, click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears.

   Note:

   If you are creating a field variable in the initial step, only the **Direct specification**, **From results or output database file**, analytical field, and discrete field distribution options are available.

   - Select **Direct specification** to specify the field variable variation over the selected region in the field editor.
   - Select **From results or output database file** to read field variable values from the results or output database file of a previous Abaqus analysis.
   - Select **User-defined** to define field variable values in user subroutine *UFIELD* for Abaqus/Standard analyses or user subroutine *VUFIELD* for Abaqus/Explicit analyses. See *Specifying general job settings* for more information.
   - Select **From results or output database file and user-defined** to read field variable values from the results or output database file of a previous Abaqus analysis and modify them in user subroutine *UFIELD* or *VUFIELD*.
   - Select an analytical field, labeled with an (A), or a discrete field, labeled with a (D), to define a spatially varying field variable. The selection list contains all analytical fields and only discrete fields that are valid for the field variable.

Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

4. If you selected the **Direct specification**, analytical field, or discrete field distribution option, perform the following steps:
   a. Click the arrows to right of the **Field variable number** field to indicate the number of the field variable you are defining.
   b. Click the arrow to the right of the **Section variation** field, and select an option from the list that appears. Only options that are valid for the selected region appear in the list. Beam and shell gradient values are not affected by an analytical field or a discrete field. You should select an option that is consistent with the field variable variation method defined in the section definition associated with the region selected for this field variable as follows:
      - Select **Constant through region** to define a constant field variable value over a section.

        In the **Magnitude** text field, enter the magnitude of the field variable across the section.
      - Select **Gradient through beam section** to define the field variable variation through a beam section. (This method is not available for axisymmetric models.)

        In the **Reference magnitude** text field, enter the magnitude of the field variable at the cross-section origin. In the **N2 gradient** text field, enter the field variable gradients through the section in the $\mathbf{n_2}$-direction of the beam.
      - Select **Defined at shell/beam field points** to define a piecewise linear field variable variation through a shell or beam section.

        Select the number of field points to be specified by either typing an integer in the **Field points** text field or using the arrows to the right of the text field to select a number, and enter the magnitude of the field variable at each point in the **Section Data** table. The number of values in the section definition should be less than or equal to the number of field variable data points given for this field. If the number of values is less, the last value will be repeated to match the number of field variable data points.

   c. If desired (for all steps other than the initial step), click the arrow to the right of the **Amplitude** text field, and select the amplitude of your choice from the list that appears. Alternatively, you can click  to create a new amplitude. (See *The Amplitude toolset* for more information.) Beam and shell gradient values will be modified by the amplitude definition.
   d. Click **OK** to save your data and to exit the editor.
5. If you selected the **From results or output database file** distribution option, perform the following steps:
   a. Click the arrows to the right of the **Field variable number** field to indicate the number of the field variable you are defining. The default is field variable number 1. To define more than one field variable, you can create additional predefined fields. You must number the field variables consecutively starting from 1.
   b. In the **File name** text field, enter the name of the results or output database file from which field variable data are to be read; or click  to display the **Select Results or Output Database File** dialog box and select the file of your choice. (See *Using file selection dialog boxes* for more information.)
   c. If you are reading results from an output database in an Abaqus/Standard analysis, you can specify a scalar nodal output variable to be read and used to initialize the field variable.
   d. If you are creating a field variable in the initial step:

- In the **Step** text field, enter the step number of the analysis whose results or output database file is being used as input from which the field variable data should be read.
- In the **Increment** text field, enter the increment number of the analysis whose results or output database file is being used as input from which the field variable data should be read.

e. Optionally, if you are creating a field variable in an analysis step other than the initial step:

- In the **Begin step** text field, enter the step number of the analysis whose results or output database file is being used as input that begins the field variable data to be read. By default, the first step available in the results or output database file is used.
- In the **Begin increment** text field, enter the increment number of the analysis whose results or output database file is being used as input that begins the field variable data to be read. By default, the first increment available in the results or output database file is used.
- In the **End step** text field, enter the step number of the analysis whose results or output database file is being used as input that ends the field variable data to be read. By default, the **Begin step** value is used.
- In the **End increment** text field, enter the increment number of the analysis whose results or output database file is being used as input that ends the field variable data to be read. By default, the last increment available for the **End step** value in the results or output database file is used.

f. Choose the **Mesh compatibility**.

- Select **Compatible** if the meshes in the original analysis and the current analysis are the same or differ only in the element order.
- Select **Incompatible** if the meshes in the original analysis and the current analysis are dissimilar. This option is valid only if the field variable values are being read from an output database file.

g. Click **OK** to save your data and to exit the editor.

6. If you selected the **User-defined** distribution option, perform the following steps:

a. Click **OK** to exit the editor.

b. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

c. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the field variable values. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

7. If you selected the **From results or output database file and user-defined** distribution option, follow the procedures outlined in Steps 5 and 6 for the **From results or output database file** and **User-defined** distribution options.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*

- *Creating discrete fields*
- *Predefined Field Variables*

Defining a material assignment field

A material assignment field is used to specify the geometric composition of different materials in the initial state of an Eulerian part instance.
The available materials that can be assigned depend on the material instances defined in the Eulerian section assigned to the part. For a general overview of material assignments for Eulerian parts, see *Assigning materials to Eulerian part instances*.

Each Eulerian part instance in a model can have only one material assignment field applied to it. Conversely, a material assignment field can be applied to only one Eulerian part instance; if you have multiple Eulerian part instances in your model, you must create a material assignment field for each instance.

Materials are assigned by selecting a region of the Eulerian part instance, then specifying the material instances that are present in that region. Each material instance is assigned a percentage indicating the volume fraction occupied by the material instance within the selected region. If the material instance percentages within a region do not add up to 100%, the remaining volume fraction in the region is occupied by a void with no material properties. A void is also assigned by default to any regions within the part instance for which a material assignment is not explicitly specified. For a detailed discussion of how Abaqus interprets material volume fractions, refer to *Eulerian Analysis*.

1. Display the material assignment field editor using one of the following methods:
   - To create a new material assignment field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Material assignment`). You can create a material assignment field only during the initial step.
   - To edit an existing material assignment field using menus or managers, see *Editing step-dependent objects*. You can modify a material assignment field only during the initial step, and you cannot edit the region to which the material assignment field is applied.

2. Specify the method you are using to define volume fractions for each material instance:
   - Choose **Uniform** to define volume fractions that are uniform across each region.
   - Choose **Discrete fields** to define the volume fractions using discrete fields. Discrete fields can be used to define volume fractions that are spatially varying across a region.

3. Select a region for the material assignment. The region must be contained within the part instance that was selected in Step 1. If the Eulerian part consists of a single cell (there are no interior partitions), Abaqus/CAE automatically selects the whole part.
   a. Click mouse button 3 in the **Region** column of the **Volume Fractions** table, and select **Edit Region** from the menu that appears. Alternatively, you can double-click mouse button 1 in the **Region** column.
   b. Select cells in the part instance using one of the following methods:
      - Select and unselect cells in the viewport. When you have finished selecting the cells, click mouse button 2. (For more information, see *Selecting objects within the viewport*.)
      - To select from a list of existing sets, click **Sets** on the right side of the prompt area, select the set of interest in the **Region Selection** dialog box, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. The remaining columns in the **Volume Fractions** table correspond to the material instances in the Eulerian section. The material instance name is appended to the part instance name in the column headings (for example, a column labeled **Bottle-1.Water** corresponds to a material instance named `Water` in a part instance named `Bottle-1`). If you are defining uniform volume fractions, a **Void** column also appears; you cannot edit values in the **Void** column. In the columns corresponding to each material instance, enter the appropriate volume fraction for the region:

   - If you are defining uniform volume fractions, enter the volume fraction for each material instance as a percentage in decimal form. For example, enter `0.45` for a volume fraction of 45%. Values greater than `1` are automatically reset to `1`. To assign all material instances equally to a region, click mouse button 3 anywhere in the **Volume Fractions** table, and select **Equalize Rows** from the menu that appears.

     Note:

     As you enter material volume fractions, Abaqus/CAE automatically adjusts the value in the **Void** column so that all values in the table row sum to one. If the sum of the entered values is greater than one, Abaqus/CAE automatically sets **Void** to zero and normalizes the remaining values so that they sum to one.

   - If you are defining the volume fractions using discrete fields, click in the material instance columns; then click the arrow that appears and select a name from the list of available discrete fields. If the discrete field has not been defined, click mouse button 3 anywhere in the **Volume Fractions** table, and select **Create Discrete Field** from the menu that appears to access the Discrete Field toolset (see *Creating discrete fields*). Volume fractions in the discrete field definition should be specified as percentages in decimal form; for example, enter `0.45` for a volume fraction of 45%. Data in the discrete field definition for elements that are not within the selected region are ignored.

     Warning:

     Because discrete fields are mesh-dependent, subsequent modifications to the Eulerian part mesh will invalidate a material assignment based on discrete fields.

5. To clear all material volume fraction values from a row, click mouse button 3 anywhere in the row, and select **Clear Rows** from the menu that appears.
6. To define material volume fractions in another region of the part instance, click mouse button 3 in the **Volume Fractions** table, and select **Add Row** from the menu that appears.

   Abaqus/CAE adds a new row to the **Volume Fractions** table. Repeat the above steps to define a new region and its associated material volume fractions.
7. Once material instances have been appropriately assigned to all regions of the part instance, click **OK** to save your data and to exit the material assignment field editor. If material instances are not assigned to some regions of the part instance, those regions are initially assigned a void with no material properties.

Additional information

- *Creating Eulerian sections*
- *Creating predefined fields*
- *Eulerian analyses*

Defining an initial state field

You can select part instances from your model and associate an initial state field with the instances. An initial state field defines a deformed mesh and its associated material state using data imported from a previous Abaqus/Standard or Abaqus/Explicit analysis. You can create an initial state field only during the initial step. For more information, see *Transferring results between Abaqus analyses*, and *About Transferring Results between Abaqus Analyses*.

1. Display the initial state field editor using one of the following methods:
   - To create a new initial field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Initial state`). You can create an initial state field only during the initial step.
   - To edit an existing initial state field using menus or managers, see *Editing step-dependent objects*. You can modify an initial state field only during the initial step.
2. In the **Job name** field of the editor, enter the name of the job from which Abaqus/CAE will import the deformed mesh and material state. Abaqus/CAE imports data from several of the files created by the original analysis. As a result, the files from the analysis must reside in the directory from which you started the current Abaqus/CAE session.
3. Choose the step from which to import. Do one of the following:
   - Choose **Last step** to import data from the last step of the previous analysis.
   - Choose **Specify**, and enter an integer specifying the step from which to import data from the previous analysis. A value of 1 specifies the first step.
4. Choose the frame from which to import. Do one of the following:
   - Choose **Last** to import data from the last frame of the specified step.
   - Choose **Specify**, and enter an integer specifying the frame of the specified step. A value of 1 specifies the first frame.
5. By default, Abaqus/CAE does not use the imported data to update the reference configuration. As a result, displacements and strains are calculated as total values relative to the reference configuration at the start of the analysis and will be continuous from the start of the analysis. Toggle on **Update reference configuration** to reset the reference configuration to be the imported configuration. Displacements and strains are now calculated relative to the new imported configuration.
6. Click **OK** to create the initial state field and to close the dialog box.

   Yellow circles appear on the model representing the initial field that you just created. For more information, see *Understanding symbols that represent prescribed conditions*. Although Abaqus/CAE checks that the files from the previous analysis exist when you create the initial state field, it imports the deformed mesh and the material state from the specified step and increment only when you analyze the model in the Job module. As a result, you will not see any other changes in the current model after you create the initial state field; for example, Abaqus/CAE does not update the mesh of the current model.

Additional information

- *Importing a part from an output database*
- *Creating predefined fields*
- *Importing and Transferring Results*

Defining a saturation field

You can define the initial saturation, *s*, for a region in a coupled pore fluid diffusion/stress analysis (see *Coupled Pore Fluid Diffusion and Stress Analysis*). You can create an initial saturation field only during the initial step.

1. Display the saturation field editor using one of the following methods:
   - To create a new saturation field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Saturation`).
   - To edit an existing saturation field using menus or managers, see *Editing step-dependent objects*.
2. Click the arrow to the right of the **Distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define an initial saturation that is uniform over the region.
   - Select an analytical field to define a spatially varying initial saturation. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)
3. From the **Saturation** field, specify an initial saturation value for the region between `0.0` (for no saturation) and `1.0` (for full saturation).
4. Click **OK** to create the saturation field and to close the dialog box.

   Green squares appear on the model representing the initial saturation field that you just created.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining an initial void ratio field

For Abaqus/Standard analyses you can specify the initial values of the void ratio, *e*, at the nodes of a porous medium (see *Coupled Pore Fluid Diffusion and Stress Analysis*). The initial void ratio can be defined directly as an elevation-dependent function, by interpolation from the output database file of a previous Abaqus analysis, or by user subroutine *VOIDRI*. For more information, see *Initial Conditions*. You can create or modify an initial void ratio field only during the initial step.

1. Display the initial void ratio field editor using one of the following methods:
   - To create a new initial void ratio field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Void ratio`).
   - To edit an existing initial void ratio field using menus or managers, see *Editing step-dependent objects*.

2. Click the arrow to the right of the **Point 1 distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a void ratio that is uniform for the selected region.
   - Select **From output database file** to read void ratio values from the output database file of a previous Abaqus analysis with void ratio components.
   - Select **User-defined** to specify void ratio using options in user subroutine *VOIDRI*. See the following sections for more information:
     - *Specifying general job settings*
     - *VOIDRI*
   - Select an analytical field to define a spatially varying initial void ratio. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. For uniform distributions, do the following:
   a. From the **Elevation distribution** options, select either **Linear** or **Constant**.
   b. In the **Void ratio 1** field, specify the initial void ratio in the first region in your model.
   c. If you selected an elevation distribution that varies linearly, specify the following options:
      - In the **Vertical coordinate 1** field, specify the vertical position of the first location in your model for which you are specifying initial void ratio.
      - In the **Point 2 distribution** field, specify either a **Uniform** distribution for void ratio at the second elevation or select an analytical field to define a spatially varying initial void ratio at the second elevation. If you select **Uniform**, specify the void ratio and vertical position of the second location in your model.

4. If you are specifying void ratio output from an output database file, do the following:
   a. Specify the file name from which you want to import void ratio values.
   b. From the **Step** options, either select **Last** to read void ratio output from the last step of the output database or select **Specify** and enter the number of the step from which you want to read void ratio data.
   c. From the **Increment** options, either select **Last** to read void ratio output from the last increment of the selected step or select **Specify** and enter the number of the increment from which you want to read void ratio data.
   d. Toggle on **Interpolate midside nodes** to interpolate void ratio values between dissimilar meshes.

5. For user-defined void ratio distributions, do the following:
   a. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)
   b. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the predefined field. For more information, see *Specifying general job settings*.

   Note:

   You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

6. Click **OK** to create the initial void ratio field and to close the dialog box.

   Sandy brown squares appear on the model representing the initial void ratio field that you just created.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining a pore pressure field in a porous medium

For Abaqus/Standard analyses you can define the initial pore pressure field, u_w, for nodes in a coupled pore fluid diffusion/stress analysis (see *Coupled Pore Fluid Diffusion and Stress Analysis*). The initial pore pressure can be defined directly as an elevation-dependent function, by interpolation from the output database file of a previous Abaqus analysis, or by user subroutine *UPOREP*. You can create or modify a pore pressure field only during the initial step. For more information, see *Initial Conditions*.

1. Display the pore pressure field editor using one of the following methods:
   - To create a new pore pressure field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Pore pressure`). You can create a pore pressure field only during the initial step.
   - To edit an existing pore pressure field using menus or managers, see *Editing step-dependent objects*. You can modify a pore pressure field only during the initial step.

2. Click the arrow to the right of the **Point 1 distribution** field, and select the option of your choice from the list that appears:
   - Select **Uniform** to define a pore pressure that is uniform for the selected region.
   - Select **From output database file** to read pore pressure values from the output database file of a previous Abaqus analysis with pore pressure components.
   - Select **User-defined** to specify pore pressure output using options in user subroutine *UPOREP*. See the following sections for more information:
     - *Specifying general job settings*
     - *UPOREP*
   - Select an analytical field to define a spatially varying initial pore pressure. Alternatively, you can click $f(x)$ to create a new analytical field. (See *The Analytical Field toolset* for more information.)

3. For uniform distributions, do the following:
   a. From the **Elevation distribution** options, select either **Linear** or **Constant**.
   b. In the **Pore pressure 1** field, specify the initial pore pressure in the first region in your model.
   c. If you selected an elevation distribution that varies linearly, specify the following options:
      - In the **Vertical coordinate 1** field, specify the vertical position of the first location in your model for which you are specifying initial pore pressure.
      - In the **Point 2 distribution** field, specify either a **Uniform** distribution for pore pressure at the second elevation or select an analytical field to define a spatially varying initial pore pressure at the second elevation. If you select **Uniform**, specify the pore pressure and vertical position of the second location in your model.

4. If you are specifying pore pressure output from an output database file, do the following:
   a. Specify the file name from which you want to import pore pressure values.
   b. From the **Step** options, either select **Last** to read pore pressure output from the last step of the output database or select **Specify** and enter the number of the step from which you want to read pore pressure data
   c. From the **Increment** options, either select **Last** to read pore pressure output from the last increment of the selected step or select **Specify** and enter the number of the increment from which you want to read pore pressure data.

d. Toggle on **Interpolate midside nodes** to interpolate pore pressure values between dissimilar meshes.

5. For user-defined pore pressure distributions, do the following:

a. Enter the Job module, and display the job editor for the analysis job of interest. (For more information, see *Creating, editing, and manipulating jobs*.)

b. In the job editor, click the **General** tab, and specify the file containing the user subroutine that defines the predefined field. For more information, see *Specifying general job settings*.



Note:

You can specify only one user subroutine file in the job editor; if your analysis involves more than one user subroutine, you must combine the user subroutines into one file and then specify that file.

6. Click **OK** to create the pore pressure field and to close the dialog box.

Magenta squares appear on the model representing the initial pore pressure field that you just created.

Additional information

- *Understanding symbols that represent prescribed conditions*
- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*
- *Coupled Pore Fluid Diffusion and Stress Analysis*

Defining a fluid cavity pressure field

You can define the initial fluid cavity pressure for fluid-filled cavities (see *About Surface-Based Fluid Cavities*). You can create or modify a fluid cavity pressure field only during the initial step. For more information, see *Initial Conditions*. To modify fluid cavity pressure in a step other than the initial step, you must define a fluid cavity pressure boundary condition (for more information, see *Defining a fluid cavity pressure boundary condition*).

1. Display the fluid cavity pressure field editor using one of the following methods:
   - To create a new fluid cavity pressure field, follow the procedure outlined in *Creating predefined fields* (**Category**: `Other`; **Types for Selected Step**: `Fluid cavity pressure`). You can create a fluid cavity pressure field only during the initial step.
   - To edit an existing fluid cavity pressure field using menus or managers, see *Editing step-dependent objects*. You can modify a fluid cavity pressure field only during the initial step.
2. Click the arrow to the right of the **Fluid cavity interaction** field, and select the interaction to which the fluid cavity pressure applies.
3. Enter the **Fluid cavity pressure**.
4. Click **OK** to create the fluid cavity pressure field and to close the dialog box.

   A green diamond appears at the fluid cavity reference point representing the initial fluid cavity pressure field that you just created.

Additional information

- *Creating predefined fields*
- *Using analytical expression fields*
- *Creating expression fields*
- *About Surface-Based Fluid Cavities*

The Mesh module

The Mesh module contains tools that allow you to generate meshes on parts and assemblies created within Abaqus/CAE. In addition, the Mesh module contains functions that verify an existing mesh.

For information on editing orphan nodes and elements, see *What can I do with the Edit Mesh toolset?*.

In this section:

- *Understanding the role of the Mesh module*
- *Entering and exiting the Mesh module*
- *Mesh module basics*
- *Understanding seeding*
- *Assigning Abaqus element types*
- *Verifying and improving meshes*
- *Understanding mesh generation*
- *Structured meshing and mapped meshing*
- *Swept meshing*
- *Free meshing*
- *Bottom-up meshing*
- *Mesh-geometry association*
- *Understanding adaptive remeshing*
- *Advanced meshing techniques*
- *Using the Mesh module toolbox*
- *Seeding a model*
- *Creating and deleting meshes*
- *Controlling mesh characteristics*
- *Obtaining mesh information and statistics*
- *Creating a mesh part*
- *Controlling adaptive remeshing*

Understanding the role of the Mesh module

The Mesh module allows you to generate meshes on parts and assemblies created within Abaqus/CAE. Various levels of automation and control are available so that you can create a mesh that meets the needs of your analysis. As with creating parts and assemblies, the process of assigning mesh attributes to the model—such as seeds, mesh techniques, and element types—is feature based. As a result you can modify the parameters that define a part or an assembly, and the mesh attributes that you specified within the Mesh module are regenerated automatically.

The Mesh module provides the following features:

- Tools for prescribing mesh density at local and global levels.
- Model coloring that indicates the meshing technique assigned to each region in the model.
- A variety of mesh controls, such as:
  - Element shape
  - Meshing technique
  - Meshing algorithm
  - Adaptive remeshing rule
- A tool for assigning Abaqus/Standard or Abaqus/Explicit element types to mesh elements. The elements can belong either to a model that you created or to an orphan mesh.
- A tool for verifying mesh quality.
- Tools for refining the mesh and for improving the mesh quality.
- A tool for saving the meshed assembly or selected part instances as a mesh part.

Entering and exiting the Mesh module

You can enter the Mesh module at any time during an Abaqus/CAE session by clicking **Mesh** in the **Module** list located in the context bar. Upon entering the **Mesh** module, the Abaqus/CAE interface changes in the following ways:

- The **Seed**, **Mesh**, **Feature**, and **Tools** menus appear on the main menu bar.
- The **Object** field that appears in the context bar allows you to display either a part or the assembly.
- Abaqus/CAE may change the color of the part instances in the assembly displayed in the viewport. These color cues describe the meshability and dependence of each instance. Independent instances appear in a color that describes their meshability, while dependent intances appear blue in the Assembly context and white in the Part context. See *Meshing independent and dependent part instances*.



Note:

Part instances are color coded according to their meshability and dependence only when the **Mesh defaults** color mapping is selected. If you displayed the default color mapping in a different module, Abaqus/CAE applies the **Mesh defaults** color mapping automatically upon your entry to the Mesh module. If you selected a non-default color mapping such as **Materials** in a different module, Abaqus/CAE continues color coding according to that color mapping (in this case, by material type) when you enter the Mesh module.

To exit the Mesh module, select any other module from the **Module** list. You need not save your mesh before exiting the module; it will be saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

Mesh module basics

This section provides brief explanations of terms and concepts that you must understand to use the Mesh module effectively. It gives you an overview of the functions available and describes the role that each function plays in the mesh creation process.

In this section:

- *The meshing process*
- *Mesh attributes and controls*
- *Mesh generation*
- *Top-down meshing*
- *Bottom-up meshing*
- *Mesh technique color coding*
- *Mesh refinement*
- *Mesh optimization*
- *Mesh verification*
- *Meshing independent and dependent part instances*
- *Displaying a native mesh*

The meshing process

To create an acceptable mesh, you use the following process:

Assign mesh attributes and set mesh controls

The Mesh module provides a variety of tools that allow you to specify different mesh characteristics, such as mesh density, element shape, and element type.

Generate the mesh

The Mesh module uses a variety of techniques to generate meshes. The different mesh techniques provide you with different levels of control over the mesh.

Refine the mesh

The Mesh module provides a variety of tools that allow you to refine the mesh:

- The seeding tools allow you to adjust the mesh density in selected regions.
- The Partition toolset allows you to partition complex models into simpler subregions.
- The Virtual Topology toolset allows you to simplify your model by combining small faces and edges with adjacent faces and edges.
- The Edit Mesh toolset allows you to make minor adjustments to your mesh.

Optimize the mesh

You can assign remeshing rules to regions of your model. Remeshing rules enable successive refinement of your mesh where each refinement is based on the results of an analysis.

Verify the mesh

The verification tools provide you with information concerning the quality of the elements used in a mesh.

Mesh attributes and controls

Abaqus/CAE provides you with a variety of tools for controlling mesh characteristics:

- You can specify the density of a mesh by creating seeds along the edges of the model to indicate where the corner nodes of the elements should be located. For example, *Figure 1* displays a model with biased seeding along the top and left edges.

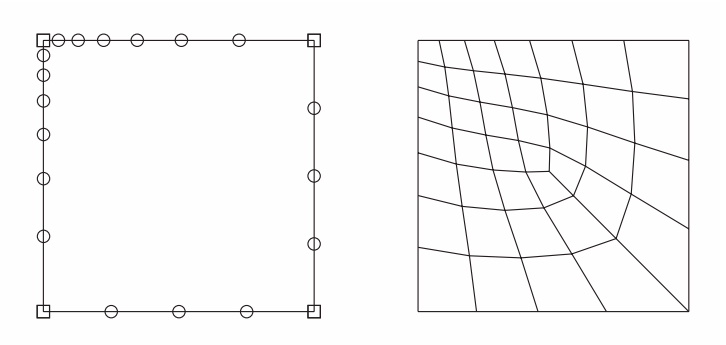


Figure 1: A model with biased seeding.

For more information, see *Understanding seeding*.

- You can select the shape of the mesh elements. For example, *Figure 2* shows a model that has been meshed first with quadrilateral elements and then with triangular elements.

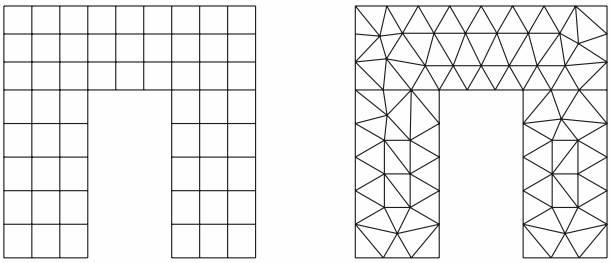


Figure 2: Two meshes with different element shapes.

For more information, see *Assigning Abaqus element types*.

- You can choose the meshing technique—free, structured, or swept—and, where applicable, you can choose the meshing algorithm—medial axis or advancing front. For more information, see *Mesh generation*.
- You can select the element type that is assigned to the mesh by choosing the element family, geometric order, and shape along with specific element controls, such as hourglassing. For more information, see *Understanding mesh generation*.

Additional information

- *Understanding seeding*
- *Assigning Abaqus element types*
- *Verifying and improving meshes*

Mesh generation

Abaqus/CAE can use a variety of meshing techniques to mesh models of different topologies. In some cases you can choose the technique used to mesh a model or model region. In other cases only one technique is valid. The different meshing techniques provide varying levels of automation and user control. There are two meshing methodologies available in Abaqus/CAE: top-down and bottom-up.

Top-down meshing generates a mesh by working down from the geometry of a part or region to the individual mesh nodes and elements. You can use top-down meshing techniques to mesh one-, two-, or three-dimensional geometry using any available element type. The resulting mesh exactly conforms to the original geometry. The rigid conformance to geometry makes top-down meshing predominantly an automated process but may make it difficult to produce a high-quality mesh on regions with complex shapes.

Bottom-up meshing generates a mesh by working up from two-dimensional entities (geometric faces, element faces, or two-dimensional elements) to create a three-dimensional mesh. You can use bottom-up meshing techniques to mesh only solid three-dimensional geometry using all—or nearly all—hexahedral elements. Generating a mesh using the bottom-up meshing technique is a manual process, and the resulting mesh may vary significantly from the original geometry. However, allowing the mesh to vary from geometry may allow you to produce a high quality hexahedral mesh on regions with complex shapes.

Additional information

- *Top-down meshing*
- *Bottom-up meshing*

Top-down meshing

Top-down meshing relies on the geometry of a part to define the outer bounds of the mesh. The top-down mesh matches the geometry; you may need to simplify and/or partition complex geometry so that Abaqus/CAE recognizes basic shapes that it can use to generate a high-quality mesh. In some cases top-down methods may not allow you to mesh portions of a complex part with the desired type of elements. The top-down techniques—structured, swept, and free meshing—and their geometry requirements are well-defined, and loads and boundary conditions applied to a part are associated automatically with the resulting mesh.

Structured meshing

Structured meshing is the top-down technique that gives you the most control over your mesh because it applies preestablished mesh patterns to particular model topologies. Most unpartitioned solid models are too complex to be meshed using preestablished mesh patterns. However, you can often partition complex models into simple regions with topologies for which structured meshing patterns exist. *Figure 1* shows an example of a structured mesh. For more information, see *Structured meshing and mapped meshing*.

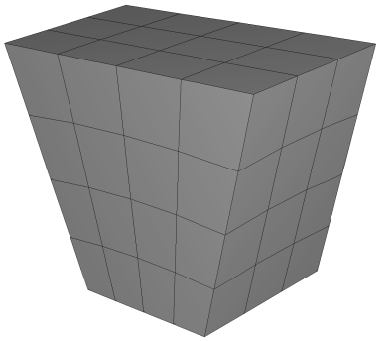


Figure 1: A structured mesh.

Swept meshing

Abaqus/CAE creates swept meshes by internally generating the mesh on an edge or face and then sweeping that mesh along a sweep path. The result can be either a two-dimensional mesh created from an edge or a three-dimensional mesh created from a face. Like structured meshing, swept meshing is a top-down technique limited to models with specific topologies and geometries. *Figure 2* shows an example of a swept mesh. For more information, see *Swept meshing*.

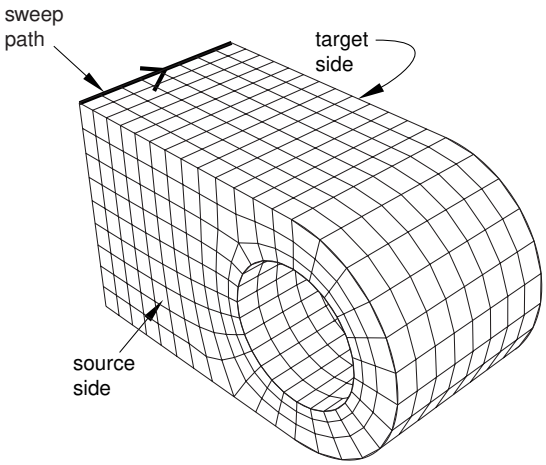


Figure 2: A swept mesh.

Free meshing

The free meshing technique is the most flexible top-down meshing technique. It uses no preestablished mesh patterns and can be applied to almost any model shape. However, free meshing provides you with the least control over the mesh since there is no way to predict the mesh pattern. *Figure 3* shows an example of a free mesh. For more information, see *Free meshing*.

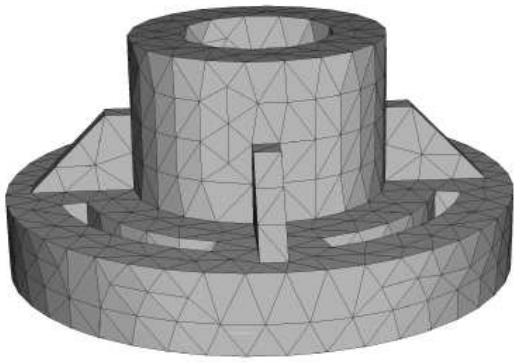


Figure 3: A free mesh generated with tetrahedral elements.

Additional information

- *Bottom-up meshing*
- *Understanding mesh generation*
- *Assigning Abaqus element types*
- *Verifying and improving meshes*

Bottom-up meshing

Bottom-up meshing uses the part geometry as a guideline for the outer bounds of the mesh, but the mesh is not required to conform to the geometry. Removing this restriction gives you greater control over the mesh and allows you to create a hexahedral or hexahedral-dominated mesh on geometry that is too complex for the structured or swept meshing techniques. Bottom-up meshing can be applied to any solid model shape. It provides you with the most control over the mesh, since you select the method and the parameters that drive the mesh. However, you must also decide whether the resulting mesh is a suitable approximation of the geometry. If it is not, you can delete the mesh and try a different bottom-up meshing method or partition the region and mesh the resulting smaller regions with either bottom-up or top-down meshing techniques.

To mesh a single bottom-up region, you may have to apply several successive bottom-up meshes. For example, you may use an extruded bottom-up mesh to generate part of a region, then use the element faces of the extruded mesh as a starting point to generate a swept mesh for features that the extruded mesh did not include.

Loads and boundary conditions are applied to geometry. Unlike a top-down mesh, a bottom-up mesh may not be fully associated with geometry. Therefore, you should check that the mesh is correctly associated with the geometry in areas where loads or boundary conditions are applied. Proper mesh-geometry association will ensure that the loads and boundary conditions are correctly transferred to the mesh during the analysis. (For more information, see *Mesh-geometry association*.) Because of the extra effort required by the user to create a satisfactory mesh compared to the automated top-down meshing processes, bottom-up meshing is recommended for use only when top-down meshing cannot generate a suitable mesh.

Figure 1 shows an example of a bottom-up meshed part. Although this part is relatively simple, it requires two regions and four bottom-up meshes to completely mesh the part. Abaqus/CAE displays bottom-up meshed regions using a mixture of the region geometry color (light tan) and the mesh color (light blue) to emphasize that the geometry and mesh may not be associated. Displaying both the geometry and the mesh allows you to view and edit the mesh-geometry associativity.

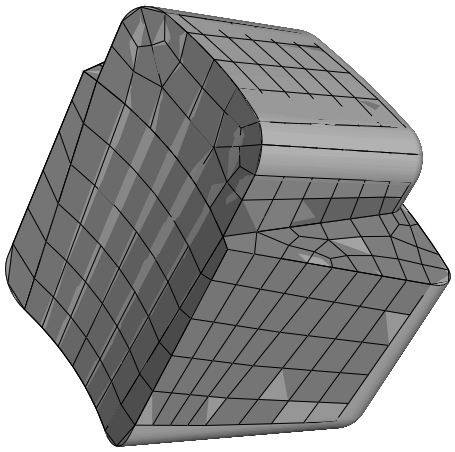


Figure 1: A bottom-up hexahedral meshed part.

Additional information

- *Top-down meshing*
- *Understanding mesh generation*
- *Assigning Abaqus element types*
- *Verifying and improving meshes*

• *Bottom-up meshing*

Mesh technique color coding

When the **Mesh defaults** color mapping is selected, Abaqus/CAE uses different colors to indicate which meshing technique, if any, is currently assigned to a region. For example, if a solid region is meshable using the structured meshing technique, the region turns green when you enter the Mesh module; the green color indicates that the structured meshing technique is assigned to that region. Yellow indicates that the sweep meshing technique is applied to a region. If a region is unmeshable using the currently assigned element shape, the region turns orange when you enter the Mesh module. Regions that are pink or light tan have been assigned the free and bottom-up meshing techniques, respectively.



Note:

You must use the **Mesh Controls** dialog box to assign the bottom-up meshing technique to a region. Abaqus/CAE does not automatically assign the bottom-up meshing technique and will not indicate whether a region that is assigned the bottom-up technique can also be meshed using a top-down technique. (For more information, see *Assigning mesh controls*.)

You can change the applicable meshing techniques by partitioning the region into smaller regions with simpler topology, by changing the element shape assigned to the region, or by using the Virtual Topology toolset.

Additional information

- *Determining which regions are meshable*
- *Color coding geometry and mesh elements*

Mesh refinement

The Mesh module provides a set of tools that allow you to refine a mesh.

- You can use the Partition toolset to divide geometric regions into smaller regions. The resulting partitions introduce new edges that you can seed; therefore, you can combine partitioning and seeding to gain additional control over the mesh generation process. You can also use partitioning to create regions to which you can assign different element types. For example, you might want to assign reduced-integration elements to some portions of your model and fully integrated elements to others. For more information, see *The Partition toolset*.
- In some cases the geometry contains details such as very small faces and edges. The Virtual Topology toolset allows you to remove these small details by combining a small face with an adjacent face or by combining a small edge with an adjacent edge. Introducing virtual topology is a convenient method for creating a clean, well-formed mesh. For more information, see *The Virtual Topology toolset*.
- You can use the Edit Mesh toolset to make minor adjustments to your mesh. For more information, see *What can I do with the Edit Mesh toolset?*.

Mesh optimization

You can assign remeshing rules to regions of your model. Remeshing rules enable successive refinement of your mesh based on solution results. After each analysis, the Mesh module adjusts your mesh with the aim of reducing selected error indicators in the solution results. For more information, see *Understanding adaptive remeshing*, *Advanced meshing techniques*, and *Creating, editing, and manipulating adaptivity processes*.

Mesh verification

The Mesh module provides a set of tools that allow you to verify a mesh and to obtain mesh statistics and mesh information. The Mesh module also provides geometry diagnostic tools that will help you determine why Abaqus/CAE cannot mesh a region. For more information, see *Verifying your mesh*, *Querying your mesh*, and *Using the geometry diagnostic tools*.

Meshing independent and dependent part instances

The approach to meshing independent and dependent instances is different. For more information, see *What is the difference between a dependent and an independent part instance?*.

Independent

To mesh an independent instance, use the context bar to change the **Object** to **Assembly** and mesh the instance directly. You cannot mesh a part that you have used to create an independent instance.

Dependent

To mesh a dependent instance, use the context bar to change the **Object** to **Part** and select the part with which the dependent instance is associated. You can then mesh the part, and Abaqus/CAE applies the same mesh to each dependent instance in the assembly. Dependent instances are convenient when you have a linear or radial pattern of part instances. You can mesh the original part, and Abaqus/CAE applies the same mesh to each instance of the part in the pattern.

Displaying a native mesh

You can switch between displaying the geometry of the part instance and the meshed representation of the same instance.

Click the **Show native mesh**  icon located in the **Visible Objects** toolbar.

You can use the **Show native mesh** tool in any of the assembly-related modules to switch between displaying the geometry of the assembly and a meshed representation of the assembly. Abaqus/CAE displays the meshed representation of both independent and dependent part instances in the assembly (assuming that you have created the appropriate meshes).

Toggling between the geometry of a part and its meshed representation using the **Show native mesh** tool allows you to see how closely the mesh follows the geometry. The tool also allows you to see how Abaqus/CAE incorporated virtual topology into the mesh. In addition, you may find it useful to click on the **Show native mesh** tool in the Job module. You can then confirm that the entire assembly has been meshed correctly before you submit a job for analysis.

The display of any orphan elements in the model is unaffected by this tool; orphan elements are displayed regardless of whether you display the geometry or elements for the native portion of part instances.

Understanding seeding

This section explains the concept of seeding and how to use seeding to improve meshes.

In this section:

- *What are mesh seeds?*
- *Can I seed a face or a cell?*
- *Controlling the seed density*
- *Applying curvature control to your seeding*
- *Constraining seeds*
- *Minimizing seed repositioning*
- *What is the relationship between vertices and nodes?*

What are mesh seeds?

Seeds are markers that you place along the edges of a region to specify the target mesh density in that region. Both the mesh density along the boundary of the region and the mesh density in the interior of the region are determined by the seeds along the edges of the region.

You can create and control seeds using the **Seed** menu in the Mesh module main menu bar. Abaqus/CAE generates meshes that match your seeds as closely as possible. Abaqus/CAE can use the following methods to control the distribution of the seeds:

- Position the seeds uniformly along all the edges of a part or part instance
- Position the seeds uniformly along an edge
- Position the seeds with a bias such that the mesh density increases toward one end of the edge
- Position the seeds with a bias such that the mesh density increases from each end toward the center of the edge
- Position the seeds with a bias such that the mesh density increases from the center toward each end of the edge

Figure 1 shows a combination of uniform seeding and bias seeding.

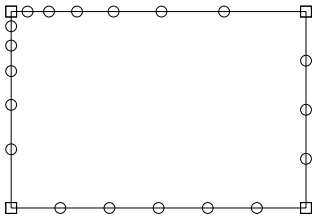


Figure 1: A model with uniform and bias seeding.

You should apply seeds to all edges. If a uniform seed distribution is sufficient, the recommended approach is to seed the entire part or part instance. If you want more control over the mesh, you can partition the region and then provide seeds along the partitions you have created. This technique is described in greater detail in *Verifying and improving meshes*.

Mesh seeds specify only a target mesh density. If you are using hexahedral or quadrilateral elements, Abaqus/CAE often changes the element distribution so that the mesh can be generated successfully. You can prevent such adjustments by constraining the number of seeds along an edge. When you constrain seeds, you are prescribing mainly the number of elements along the edge, and, to a lesser extent, the precise locations of the nodes; if necessary, Abaqus/CAE adjusts the locations of the nodes to reduce element distortion. In addition, you should use such constraints with care, since they can make it more difficult for the mesh generator to obtain a mesh.

By default, Abaqus/CAE displays seeds on a part or assembly only when you are defining or modifying seed placement. You can enable persistent seed display if you want the seeds to appear while you perform other operations in the Mesh module; toggle on  in the **Visible Objects** toolbar to maintain seed display.

Can I seed a face or a cell?

You can select edges, faces, or cells to seed; however, Abaqus/CAE creates seeds only along edges. When you select faces or cells to seed, Abaqus/CAE creates seeds only along the edges of the faces or cells. In addition, you can select a set or surface to seed; as a result, Abaqus/CAE creates seeds along the edges of the geometry contained in the set or surface.

When you are applying uniform seeds, you can use a combination of the following methods to select the region to which Abaqus/CAE will apply the seeds:

Individually/By angle

You can select edges, faces, or cells individually; or you can use the angle method to select a group of edges or faces. For example, if you choose the angle method and select an edge, Abaqus/CAE then selects every adjacent edge until the angle between the edges is equal to or exceeds the angle that you entered. For more information, see *Using the angle and feature edge method to select multiple objects*.

Selection filters

You can use filters to choose the type of object to select—**Edges**, **Faces**, **Cells**, or **All**. By default, Abaqus/CAE allows you to select all types of object. The option to select by angle becomes available only after you select **Edges** or **Faces** from the selection filters. For more information, see *Filtering your selection based on the type of object*.

Sets or surfaces

By default, Abaqus/CAE allows you to apply seeds to edges, faces, and cells that you select from the viewport. Alternatively, you can click **Sets/Surfaces** on the right of the prompt area and select from eligible sets (or surfaces). When you select a set (or surface), Abaqus/CAE applies seeds to every edge in the set (or surface), including every edge of all the cells and faces. Abaqus/CAE ignores any vertices in the set (or surface).

Controlling the seed density

You can use the following methods to control the seed density along selected edges:

- Specify the average element size for every edge of the entire part or part instance.
- Specify the number of elements desired along an edge.
- Specify the average element size along an edge. (If the edge length is not an integer multiple of the element length, Abaqus/CAE will change the element length slightly to obtain an integer number of elements along the edge.)
- Specify a nonuniform distribution of elements along an edge. The element density can increase from one end of the edge to the other (single bias), or the element density can vary from the center of the edge to each end (double bias). For a nonuniform distribution you can specify either of the following:
  - The number of elements desired along an edge and a bias ratio. The bias ratio is the ratio of the largest element to the smallest element.
  - The size of the smallest element and the size of the largest element.

If you select edges that you previously seeded using a combination of these methods, Abaqus/CAE provides an **As Is** option that allows you to retain the seeding method. Abaqus/CAE provides a similar option if you select edges with a mixture of curvature controls or seed constraints.

For detailed instructions on prescribing seed density, see the following sections:

- *Defining seed density for an entire part or part instance*
- *Seeding an edge by prescribing the number of elements*
- *Seeding an edge by prescribing element size*
- *Prescribing biased seeding along an edge*
- *Applying constraints to edge seeds*
- *Seeding previously meshed parts, part instances, or regions*
- *Deleting part or instance seeds*
- *Deleting edge seeds*

Seeds created by specifying an average element size for the entire part or part instance are called part seeds or instance seeds, respectively, and appear in white; seeds created using the other methods are called edge seeds and appear in magenta. Edge seeds always override part or instance seeds; therefore, when you specify the average element size for the entire part or part instance, part or instance seeds appear only on edges of the region that do not already have edge seeds. New edges created by partitioning are given part or instance seeds by default.

When you seed an edge of a region that is assigned the swept or revolved mesh technique, the edge seeding tools automatically propagate seeds from the selected edge to the matching edges in the region. In other words, the seeds on the face or edge at the beginning of the sweep path are propagated automatically to the face or edge at the end of the sweep path. Likewise, the seeds created on one edge along the sweep path are propagated automatically to the other edges along the sweep path. As a result, even though you select a single edge of a face to seed, Abaqus/CAE will propagate the seeds to additional edges and faces. For more information, see *What is swept meshing?*.

Applying curvature control to your seeding

The part seeding tool allows you to specify a target element size when you are seeding a part, a part instance, or multiple edges. If the geometry is relatively regular, specifying a single target element size can result in an acceptable mesh. However, if you specify a single target element size and the geometric features that make up the part or edges vary in size, the resulting mesh may be too coarse to adequately represent any small features, as shown in *Figure 1*.

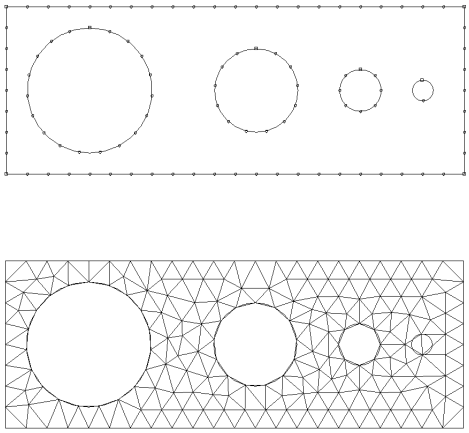


Figure 1: Seeding and the resulting mesh with no curvature control.

To avoid the problem of inadequate seeding around small curved features, Abaqus/CAE applies curvature control when it seeds a part, a part instance, or edges. Curvature control allows Abaqus/CAE to calculate the seed distribution based on the curvature of the edge along with the target element size. *Figure 2* shows the same part seeded and meshed with curvature control enabled.

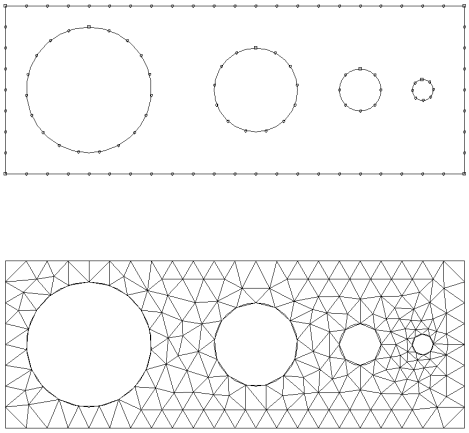


Figure 2: Seeding and the resulting mesh with curvature control enabled.

You can configure the following to specify how curvature control will influence the seeding:

Deviation factor

The deviation factor is a measure of how much the element edges deviate from the original geometry, as shown in *Figure 3*.

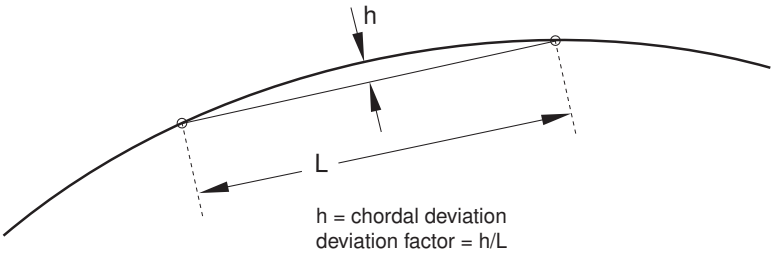


Figure 3: Deviation factor.

To help you visualize the deviation factor, Abaqus/CAE displays the approximate number of elements it would create around a circle corresponding to the setting that you enter. As you reduce the deviation factor, the number of elements that Abaqus/CAE would create around a circle increases. This number is only a visual aid; for example, if you are seeding a spline or an ellipse, Abaqus/CAE creates a different number of elements, depending on the local curvature along the edge.

Specify minimum size factor

Specifying a minimum size factor prevents Abaqus/CAE from creating very fine meshes in areas of high curvature that you have no interest in modeling; for example, kinks in spline curves or fillets with a very small radius. The number that you enter representing the minimum size is the fraction of the global seed size. As a result, if you change the global seed size, you do not have to change the minimum size factor.

For detailed instructions on applying curvature control, see *Defining seed density for an entire part or part instance*.

Constraining seeds

By default, mesh seeds prescribe only a target mesh density. Abaqus/CAE generally matches the mesh seeds exactly when you are using the free meshing technique to generate triangular- or tetrahedral-shaped elements. However, in other cases Abaqus/CAE may alter the element distribution so that it can successfully generate the mesh. If you want to prevent Abaqus/CAE from altering the element distribution, you can fix a specific number of elements along an edge by constraining the seeds along that edge. You can constrain only edge seeds, not part or instance seeds.

You can assign any one of the following three states to a group of seeds along an edge:

Unconstrained

This is the default setting. The number of elements along an edge can either increase or decrease so that the mesh can become denser or coarser than is specified by the seeds. Unconstrained seeds appear as open circles.

Partially constrained

The number of elements along an edge may be increased during mesh generation but cannot be decreased. This constraint allows the mesh to become denser than is specified by the seeds but no coarser. Partially constrained seeds appear as upward-pointing triangles.

Fully constrained

The number of elements specified by constrained seeds along an edge cannot be altered by the mesh generation process. When the seeds are fully constrained, the mesh generation will attempt to allow the location of the nodes to correspond exactly to the location of the seeds. However, an exact match between the seeds and the nodal positions is not guaranteed. Fully constrained seeds appear as squares.

Abaqus/CAE always creates a fully constrained seed at each geometric vertex of a region to indicate that a finite element node will be positioned at each vertex.

In many cases the mesh generator must redistribute elements (and deviate from the number and location of the seeds) to generate a mesh successfully. For the greatest likelihood of meshing success, leave seeds unconstrained or at least avoid fully constraining large numbers of seeds in a given part or part instance so that the mesh generator has as much freedom to redistribute seeds as possible.

For detailed instructions on constraining edge seeds, see the following sections :

- *Applying constraints to edge seeds*
- *Relaxing constraints using the error dialog box*

Minimizing seed repositioning

During the mesh generation process Abaqus/CAE uses the seeds that you create as target locations for nodes along the edges of the mesh. However, if you are using quadrilateral- or hexahedral-shaped elements, a close match between your seeds and nodes depends heavily on the following:

The element shapes you allow in transition regions

You will obtain a better match between your seeds and the nodes of the mesh if you allow triangular elements in transition regions. The seeds and the nodes are less likely to match if you restrict your mesh to including only quadrilateral elements.

The mesh transition setting

You will obtain a better match between your seeds and the nodes of the mesh if you allow for mesh transition.

The meshing technique

The mesh generated using the advancing front meshing algorithm matches your seeding better than the mesh generated using the medial axis algorithm.

The seed constraints

Fully constrained seeds closely match the generated nodes in both number and position. However, you must fully constrain only a few edges of a part or part instance; otherwise, Abaqus/CAE will not be able to generate a mesh.

How neighboring regions are seeded

When meshing multiple regions, Abaqus/CAE often redistributes the elements so that the mesh is compatible between regions. Even though a single region's seed arrangement may be adequate for generating a mesh on that one region, the seed arrangement may need to be changed since the number of elements must be compatible with neighboring regions along shared edges.



Note:

Mesh compatibility between part instances is not guaranteed. In some simple cases seeding can help achieve part-to-part mesh compatibility. Techniques for obtaining compatible meshes are described in *Compatible meshes between part instances*.

Abaqus/CAE tries to adhere as closely as possible to the number and location of seeds that you specified when balancing the element redistribution for the entire model. If given a choice between making a large change along a single seeded edge and making a small change to many edges, Abaqus/CAE will make many small changes.

What is the relationship between vertices and nodes?

When you seed a model, Abaqus/CAE automatically places fully constrained seeds wherever vertices appear along the model's edges. Fully constrained seeds that appear at vertices always indicate that nodes will appear at those vertices. (Fully constrained seeds that appear at other locations along an edge of a region do not indicate the exact location of nodes; they indicate only the number of nodes along that edge.) Therefore, when you sketch a part, you should keep in mind that the location of vertices in the part influences the quality of the mesh that Abaqus/CAE can generate. (For information about altering vertex locations, see *Dragging Sketcher objects*.)

For example, *Figure 1* shows a sketch of a two-dimensional part.

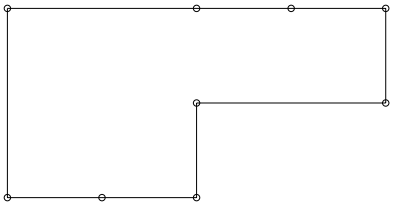


Figure 1: Vertices on a two-dimensional part.

Note the locations of the nine vertices. These vertices were created by sketching several line segments along the top and bottom edges rather than one continuous line segment along each edge.

When that part or an instance of the part is seeded, square-shaped, fully constrained seeds appear at each vertex, as shown in *Figure 2*.

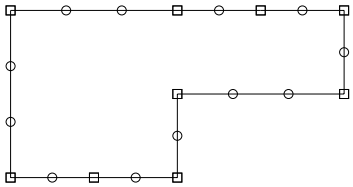


Figure 2: Fully constrained seeds appear at each vertex.

When the model is meshed, Abaqus/CAE always places nodes at the location of the fully constrained seeds that are located at vertices, as shown in *Figure 3*.

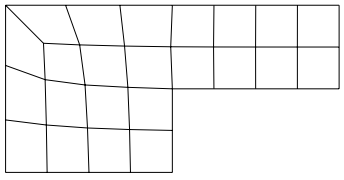


Figure 3: Nodes appear at the vertices.

Likewise, *Figure 4* shows the sketch of two concentric circles that will be extruded to form a hollow cylinder.

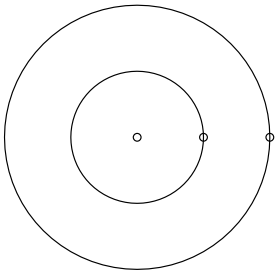


Figure 4: Concentric circles with aligned vertices.

Note the location of the vertices, which the Sketcher creates at the locations you click to define the circles' perimeters.

When the cylinder is seeded, square-shaped, fully constrained seeds appear at each vertex, as shown in *Figure 5*.

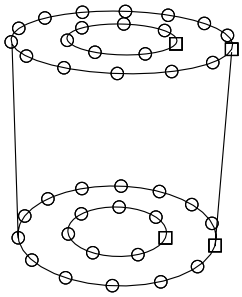


Figure 5: Fully constrained seeds appear at each vertex.

When the model is meshed, nodes always appear at the location of the fully constrained seeds that are located at vertices, as shown in *Figure 6*.

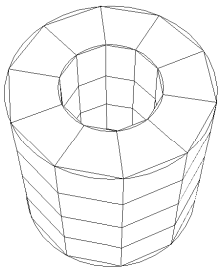


Figure 6: Nodes appear at the vertices.

If you do not align the two vertices when you sketch the cylinder, you risk generating a distorted mesh. For example, the vertices of the two concentric circles are not aligned in *Figure 7*.

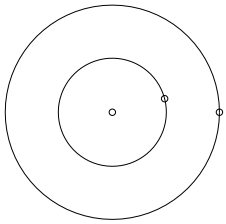


Figure 7: Concentric circles whose vertices are not aligned.

As a result, the mesh is slightly distorted on the right side, as shown in *Figure 8*.

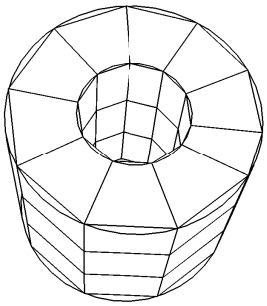


Figure 8: A distorted mesh.

Assigning Abaqus element types

This section explains how to assign Abaqus/Standard and Abaqus/Explicit element types to geometric regions and to orphan elements.

In this section:

- *How do mesh elements correspond to Abaqus elements?*
- *What kinds of elements must be generated outside the Mesh module?*
- *Element type assignment*
- *Preferred element types list*

How do mesh elements correspond to Abaqus elements?

The Mesh module can generate meshes containing the following element shapes.

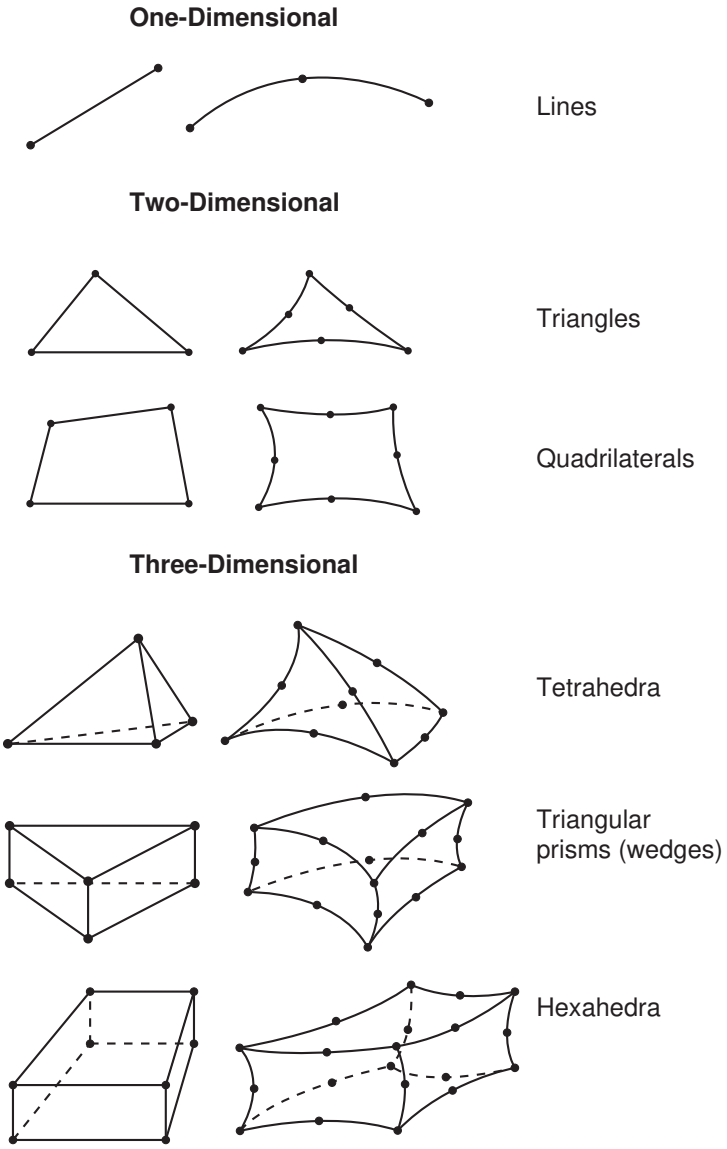


Figure 1: Element shapes.

Most elements in Abaqus/Standard and Abaqus/Explicit correspond to one of the shapes shown; that is, they are topologically equivalent to these shapes. For example, although the elements CPE4, CAX4R, and S4R are used for stress analysis, DC2D4 is used for heat transfer analysis, and AC2D4 is used for acoustic analysis, all five elements are topologically equivalent to a linear quadrilateral.

Every mesh region has one or more Abaqus element types assigned to it by default. Each element type corresponds to an element shape that can be used in the region. For example, a shell mesh region typically has a quadrilateral and a triangular element type assigned to it by default. However, you can change the element assignment for any Abaqus element that is topologically equivalent to the element shape assigned to the region. As a result, you can choose to mesh a shell region with only all triangular elements, and Abaqus/CAE ignores the quadrilateral element assignment.

To change the element assignment to an Abaqus element that is topologically equivalent to the element shape assigned to the region, select **Mesh->Element Type** from the main menu bar. Similarly, you can select **Mesh->Controls** to select the element shape for meshing.

However, since no element type checking is done until you submit the analysis, it is possible to choose an element that is inappropriate for the analysis you will be conducting. For example, Abaqus/CAE does not prevent you from specifying heat transfer elements such as DC2D4, even though you may be conducting a stress analysis.

What kinds of elements must be generated outside the Mesh module?

Abaqus/CAE provides support for most of the elements that are used by Abaqus/Standard and Abaqus/Explicit. However, some elements are not supported and must be generated outside the Mesh module.

The following list describes the elements that are not supported by Abaqus/CAE. If you want to assign these element types to a model, you must use a text editor to add them to the input file generated in the Job module. For information on generating the input file, see *Basic steps for analyzing a model*.

- Acoustic interface element (ASI1)
- Coupled thermal-electrical-structural elements (Q3D4, Q3D6, etc.)
- Distributing coupling elements (DCOUP2D and DCOUP3D)
- Drag chain elements (DRAG2D and DRAG3D)
- Elastic-plastic joint elements (JOINT2D and JOINT3D)
- Frame elements (FRAME2D and FRAME3D)
- Gap elements, coupled temperature-displacement and heat transfer (GAPUNIT and DGAP)
- Infinite elements (CIN3D8, CINAX4, etc.)
- Line spring elements (LS3S and LS6)
- Membrane elements, 9-node quadrilateral (M3D9 and M3D9R)
- Membrane elements, cylindrical (MCL6 and MCL9)
- Particle element (PC3D)
- Pipe-soil interaction elements (PSI24, PSI34, etc.)
- Slide line elements (ISL21A and ISL22A)
- Stress/displacement variable node elements (C3D15V, C3D27, etc.)

  Note:

  After you submit the analysis for execution, Abaqus/Standard automatically converts any C3D20*(R)(H)* element that is adjacent to a secondary surface in a contact pair into the corresponding C3D27*(R)(H)* element. (Neither element is available in Abaqus/Explicit.) Otherwise, there is no way to generate variable node hexahedra with Abaqus/CAE.

- Surface elements, cylindrical (SFMCL6 and SFMCL9)
- Thin shell element, 9-node doubly curved (S9R5)
- Tube-to-tube contact elements (ITT21 and ITT31)
- Poroelastic acoustic elements (C3D4A, C3D6A, and C3D8A)

Some pyramid elements, such as C3D5 and AC3D5, are supported by Abaqus/CAE in that they can be imported from and written to input files. However, Abaqus/CAE has no meshing algorithms that generate a pyramid shape. You can assign a different pyramid element type to existing pyramid elements in the Mesh module.

You cannot assign some elements, such as CONN2D2 and SPRING1 in the Mesh module; however, you can create equivalent connectors in the Interaction module or engineering features in the Property module or Interaction module as shown in *Table 1*. These elements are written to the input file.

Table 1: Abaqus/CAE support for connectors and engineering features.

Elements	Abaqus/CAE support
CONN2D2, CONN3D2	Equivalent connector in Interaction module
DASHPOTA, DASHPOT1, DASHPOT2	Engineering feature in Property module or Interaction module (linear behavior independent of field variables)
	Equivalent connector in Interaction module
GAPCYL, GAPSPHER, GAPUNI	Equivalent connector in Interaction module
HEATCAP	Engineering feature in Property module or Interaction module
ITSCYL, ITSUNI	Equivalent connector in Interaction module
JOINTC	Equivalent connector in Interaction module
MASS	Engineering feature in Property module or Interaction module
ROTARYI	Engineering feature in Property module or Interaction module
SPRINGA, SPRING1, SPRING2	Engineering feature in Property module or Interaction module (linear behavior independent of field variables)
	Equivalent connector in Interaction module

For more information, see *Understanding connectors*, *Inertia*, and *Springs and dashpots*.

Element type assignment

You can assign element types to geometric regions and to orphan mesh elements.

Element types can be assigned to the following:

- A region selected from geometry-based parts or part instances. The part instances must have come from parts that you created in the Part module or from parts that you imported.
- A set that refers to a region selected from geometry-based parts or part instances. The set can also refer to a skin reinforcement.
- An orphan element or element set.

All regions from geometry-based parts or part instances have default element type assignments. These assignments depend on the kind of part to which the region or element belongs. You can also specify a list of preferred element types for element type assignment (see *Preferred element types list*, for more information).

You can view and change the Abaqus element types that are assigned using the **Element Type** dialog box, which you can display by selecting **Mesh->Element Type**. For example, the **Element Type** dialog box for a two-dimensional structural region is shown in *Figure 1*.

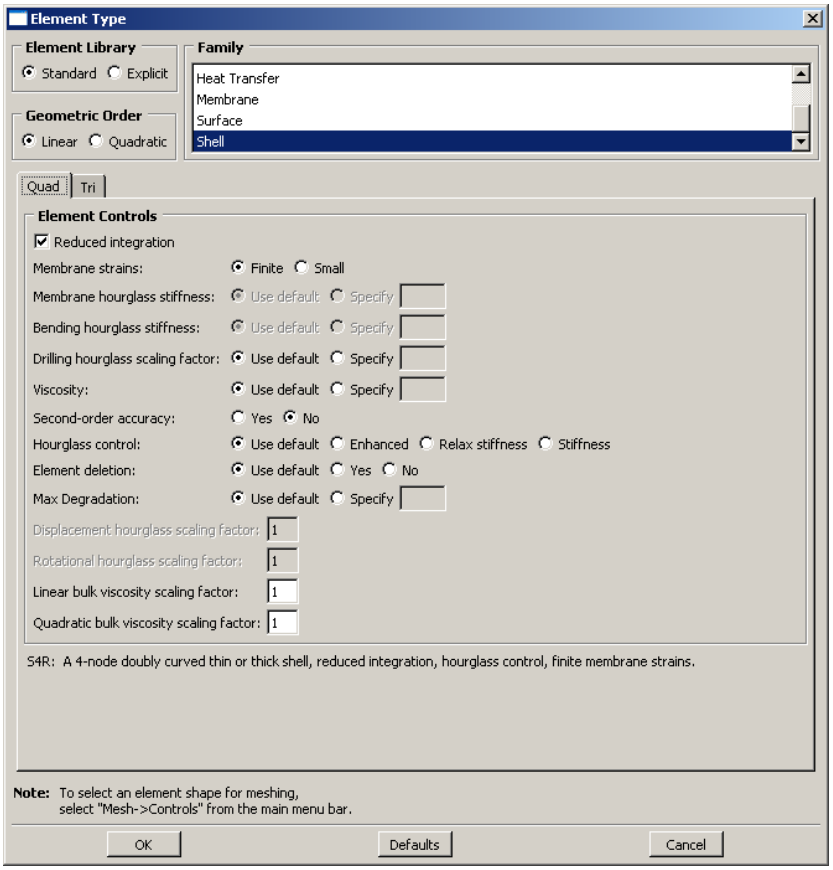


Figure 1: The Element Type dialog box for a two-dimensional structural region in an Abaqus/Standard model.

At the top of the dialog box, you enter your preferences for element library, geometric order, and family. Then, you select a specific element type by clicking the tabs in the bottom half of the dialog box and choosing from the options that appear. For more information on the element control options, see *Section Controls*.

The dialog box can contain from one to three tabs depending on the dimensionality of the selected region or regions:

- The **Line** tab allows you to choose an applicable element type and assign it to one-dimensional mesh elements in the region.
- The **Quad** and **Tri** tabs allow you to choose an applicable element type and assign it to two-dimensional mesh elements in the region.
- The **Hex**, **Wedge**, and **Tet** tabs allow you to assign three-dimensional element types to the three-dimensional mesh elements in the region.

For example, in *Figure 1* the options for a linear shell element from the Abaqus/Standard element library are selected. After clicking the **Quad** tab, reduced integration and finite membrane strains are selected. The name and a brief description of the quadrilateral shell element that meets all of these criteria appear at the bottom of the tabbed page.

The **Tri** tab in this dialog box is shown in *Figure 2*. The name and a brief description of the triangular shell element that meets all of the criteria specified in the dialog box appear at the bottom of the **Tri** tabbed page in *Figure 2*. If the selected region in this example happens to contain a combination of triangular and quadrilateral mesh elements:

- The quadrilateral mesh elements are assigned the S4R element type.
- The triangular mesh elements are assigned the S3 element type.

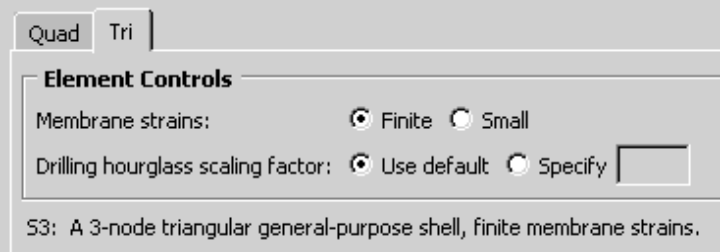


Figure 2: The Tri tab.

If the region contains only quadrilateral elements, all of the elements are assigned the S4R element type.

For detailed, step-by-step instructions for assigning element types to a mesh region, see *Associating Abaqus elements with mesh regions*. For lists of the element types that are available, see the *Abaqus/Standard Element Index* and the *Abaqus/Explicit Element Index*. You can select most of these elements through the **Element Type** dialog box. *What kinds of elements must be generated outside the Mesh module?* describes the elements that cannot be selected.

Preferred element types list

You can specify a list of preferred element types for element type assignment.

Abaqus/CAE must be run interactively to use preferred element types. The list must be created prior to loading the Mesh module in Abaqus/CAE.

When a part or part instance that has never been assigned an element type is meshed, the preferred element type list is consulted. If an element type appropriate to the geometry is found in the list, it is assigned to the geometry. Multiple element types representing different shapes (for example, triangles and quadrilaterals) can be assigned in combination, but only element types that are compatible with each other are used. When more than one appropriate element type is found in the list, the first element type encountered takes precedence.

The list is also consulted when populating the **Element Type** dialog box such that preferred element types are selected by default for a region not previously assigned any element types. If you click **Defaults** in the dialog box, the default element types (not the preferred element types) are displayed.

You can specify the preferred element types list in either the `abaqus_v6.env` or the `custom_v6.env` environment file. It is recommended that you specify the list of preferred element types in the `onCaeStartup()` function in the environment file. For example:

```
def onCaeStartup():
    import mesh
    prefElems=(mesh.C3D8T, mesh.C3D10MT, mesh.S8R)
    session.defaultMesherOptions.setValues(guiPreferredElements=prefElems)
```

• *Using the Abaqus environment files*

Verifying and improving meshes

This section explains how you use the tools in the Mesh module to verify your mesh quality, to control mesh generation, and to improve the mesh quality.

In this section:

- *Verifying your mesh*
- *Querying your mesh*
- *Why partition in the Mesh module?*
- *How are seeds and other attributes affected by partitioning?*
- *Regenerating partitions after modifying geometry*
- *Using virtual topology to improve your mesh*
- *Using adaptive remeshing to improve your mesh*

Verifying your mesh

Upon completion of a meshing operation, Abaqus/CAE highlights any bad elements in the mesh. Abaqus/CAE also provides a set of tools in the Mesh module that allow you to verify the quality of your mesh and to obtain information about the nodes and elements in the mesh. You can use these tools to isolate regions where the mesh quality is poor and to guide you if you need to refine your mesh. To verify the quality of the mesh, choose the **Object** from the context bar, and select **Mesh->Verify** from the main menu bar. You can then select the part, part instances, geometric regions, or element to verify. Abaqus/CAE allows you to choose between checking that your mesh will pass the quality tests in the analysis products and checking that your mesh passes individual quality checks, such as checking for elements with a large aspect ratio. Any elements that do not pass the specified criteria are highlighted in the viewport, and you can choose to create and save a set containing the highlighted elements or, if applicable, the cells, faces, or edges related to those elements. For detailed information on using the mesh verify tools, see *Verifying element quality*.

You can use the **Analysis checks** to verify that the elements in your mesh will pass the element quality checks that are included with the input file processor in Abaqus/Standard or Abaqus/Explicit. Abaqus/CAE highlights any elements that fail the mesh quality tests and displays the number of elements tested along with the number of errors and warnings in the message area. The mesh quality tests in the input file processor are extensive and specific to each element type. At a minimum, the mesh quality tests issue a warning for elements that seem inappropriately distorted, and the tests issue an error if the distortion is severe. Abaqus/CAE does not support analysis checks for beam, gasket, or cohesive elements.

You can use the **Shape metrics** to highlight elements of a selected shape that do not meet one of the following selection criteria:

Shape factor

Abaqus/CAE highlights elements with a normalized shape factor smaller than a specified value. The shape factor criterion is available only for triangular and tetrahedral elements. The shape factor ranges from 0 to 1, with 1 indicating the optimal element shape and 0 indicating a degenerate element.

- For triangular elements the normalized shape factor is defined as

$$shape\ factor = \frac{element\ area}{optimal\ element\ area}.$$

  Optimal element area is the area of an equilateral triangle with the same circumradius as the element. (The circumradius is the radius of the circle passing through the three vertices of the triangle.)

- For tetrahedral elements the normalized shape factor is defined as

$$shape\ factor = \frac{element\ volume}{optimal\ element\ volume}.$$

  Optimal element volume is the volume of an equilateral tetrahedron with the same circumradius as the element. (The circumradius is the radius of the sphere passing through the four vertices of the tetrahedron.)

Small face corner angle

Abaqus/CAE highlights elements containing faces where two edges meet at an angle smaller than a specified angle.

Large face corner angle

Abaqus/CAE highlights elements containing faces where two edges meet at an angle larger than a specified angle.

Aspect ratio

Abaqus/CAE highlights elements with an aspect ratio larger than a specified value. The aspect ratio is the ratio between the longest and shortest edge of an element.

Table 1 shows the default limits for the selection criteria based on the element shape.

Table 1: Element shape selection criteria limits.

Selection criterion	Quadrilateral	Triangle	Hexahedra	Tetrahedra	Wedge
Shape factor	N/A	0.01	N/A	0.0001	N/A
Smaller face corner angle	10	5	10	5	10
Larger face corner angle	160	170	160	170	160
Aspect ratio	10	10	10	10	10

You can use the **Size metrics** to highlight elements that do not meet one of the following selection criteria:

Geometric deviation factor

Abaqus/CAE highlights elements with an edge whose geometric deviation factor is greater than the specified value. The geometric deviation factor is a measure of how much an element edge deviates from the original geometry, and Abaqus/CAE calculates this value by dividing the maximum gap between an element edge and its parent geometric face or edge by the length of the element edge. By default, Abaqus/CAE highlights elements whose geometric deviation factor is greater than 0.2.

Abaqus/CAE calculates the geometric deviation factor only for elements in a native mesh. If you select a part that contains no geometry, Abaqus/CAE disables this option in the **Verify Mesh** dialog box. If your selection includes both native and orphan elements, Abaqus/CAE considers only the native elements for calculations of geometric deviation factor.

Short edge

Abaqus/CAE highlights elements with an edge shorter than a specified value.

Long edge

Abaqus/CAE highlights elements with an edge longer than a specified value.

Stable time increment

Abaqus/CAE highlights elements with a calculated stable time increment less than the specified value. The stable time increment calculation requires a suitable material definition and section assignment and is meaningful only for Abaqus/Explicit analyses.

The stable time increment calculation in Abaqus/CAE is an approximation of the initial stable time increment calculation made by Abaqus/Explicit for an element-by-element formulation. It does not account for any of the following conditions:

- Mass scaling
- Point mass
- Rotary inertia

- Nonstructural mass
- Reinforcement (rebar)

Material behaviors supported for the stable time increment calculation in Abaqus/CAE include elastic, hyperelastic, hyperfoam (without user-defined test data), and acoustic medium. Composite sections with multiple materials are not supported. For more information, see *Stability*.

Maximum allowable frequency for acoustic elements

Abaqus/CAE highlights finite acoustic elements that may not be valid for modal or steady-state dynamic analyses in Abaqus/Standard above the specified frequency value. The maximum allowable frequency calculation requires a suitable material definition and section assignment. The calculation is a guideline based on approximately 10 elements per wavelength:

$$f_{max} = \frac{PC_o}{10h},$$

where P is the interpolation order (1 or 2), h is the size of the element bounding box, and C_o is the speed of sound $\left(\sqrt{\frac{bulk\ modulus}{density}}\right)$.

In addition, for both shape and size metrics Abaqus/CAE displays the following information in the message area for each selected part, part instance, or region:

- The name of the part or part instance.
- The total number of elements of the selected shape in the part instance or in the selected regions.
- The number of highlighted elements and the percentage of the elements being verified that these elements comprise.
- The average value of the selection criterion. For the geometric deviation factor, Abaqus/CAE calculates the average value by considering only elements along a curve or face; solid elements in the center of a volume are excluded from this value.
- The “worst” value of the selection criterion—the value closest to the criterion if it is not exceeded or the value farthest beyond the criterion if it is exceeded.

Querying your mesh

The Query toolset in the Mesh module allows you to obtain information about the nodes and elements in the mesh. In addition, you can select **Tools->Query** from the main menu bar to request the following information about the mesh:

- The total number of nodes and elements in a selected part, part instance, or region along with the number of elements of each element shape.
- The type and connectivity of a selected element.
- The positive and negative sides of shell and membrane faces.
- The direction of beam and truss tangents.
- The mesh stack orientation.
- Whether any edges of boundary faces have incompatible interfaces, cracks, or gaps and whether any edges intersect other faces.
- The location of free or non-manifold edges—exterior shell or solid element edges that are not shared by exactly two exterior elements.
- The location of any unmeshed regions.

For detailed information on using the Query toolset, see *Obtaining mesh information*.

Why partition in the Mesh module?

You can use the Partition toolset to divide parts or independent part instances into smaller regions. There are three reasons to create partitions in the Mesh module:

- To divide a complex, three-dimensional part or instance into simpler regions that Abaqus/CAE can mesh using primarily hexahedral elements with the structured or swept meshing techniques. (Almost all three-dimensional parts are meshable using the free meshing technique, but three-dimensional free meshes can include only tetrahedral elements.)
- To gain more control over mesh generation.
- To obtain regions to which you can assign different element types.

See *The Partition toolset* for detailed information on how to use each tool in the Partition toolset.

You can partition only parts or independent part instances. If you need to partition a dependent instance, you can partition the original part from which the dependent instance was created. Alternatively, you can create a copy of the original part and then create an independent instance of the copy. You can then replace the dependent instance with the new independent instance and partition the independent instance. For more information, see *What is the difference between a dependent and an independent part instance?*.

By default, the free meshing technique with quadrilateral elements is applied to all two-dimensional parts and part instances. When you create the mesh using this default technique, Abaqus/CAE implicitly creates partitions that divide the part into regions that can be meshed using the structured meshing technique. (For more information, see *Free meshing with quadrilateral and quadrilateral-dominated elements*.) Therefore, all two-dimensional parts are meshable without any manual partitioning.

However, when a three-dimensional part or instance is unmeshable using hexahedral elements, you must take one of the following steps:

- Change the element shape from hexahedra to tetrahedra so that the free meshing technique can be applied.
- Partition into structured- or swept-meshable regions.

When the **Mesh defaults** color mapping is selected, Abaqus/CAE uses the color orange to indicate that a three-dimensional region is unmeshable using the currently assigned element shape. For example, *Figure 1* illustrates a part that cannot be meshed with hexahedral elements.

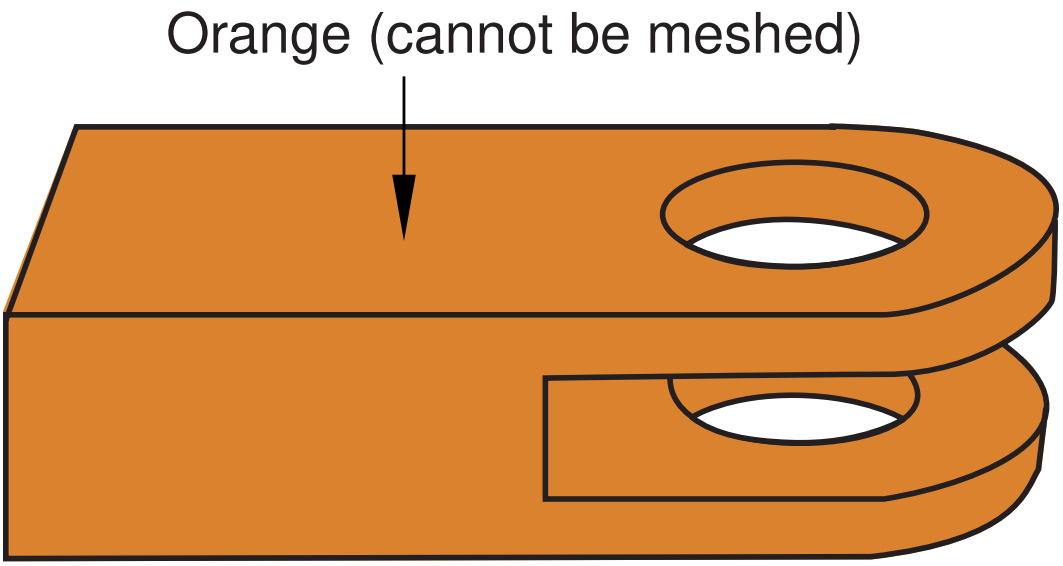


Figure 1: Unmeshable three-dimensional region.

With the addition of a partition, the part can be meshed with hexahedral elements, as shown in *Figure 2*; the green region can be meshed using the structured meshing technique, and the yellow regions can be meshed using the swept meshing technique.

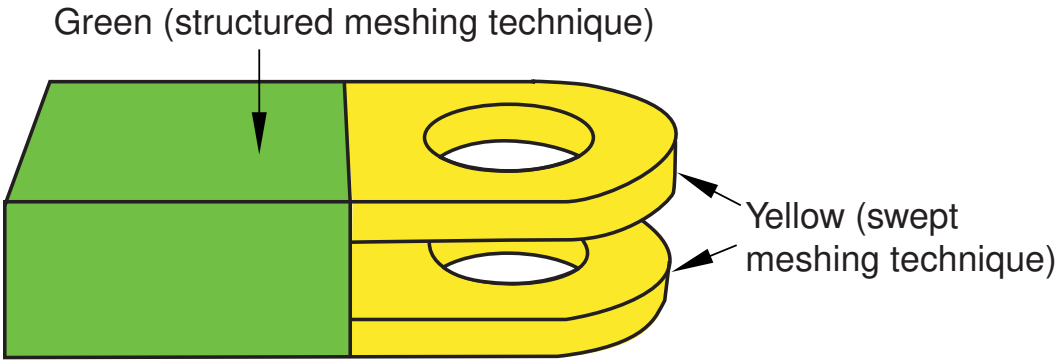


Figure 2: The model is partitioned into three regions.

Even when a part or instance can be meshed without partitioning, you may still want to partition to gain more control over mesh generation. Without partitions, the mesh is aligned only along the exterior edges; with partitions, the resulting mesh will have rows or grids of elements aligned along the partitions. That is, the mesh "flows" along the partitions. For example, in *Figure 3* the partition that divides the rectangle in two causes the mesh to flow at an angle along the partition.

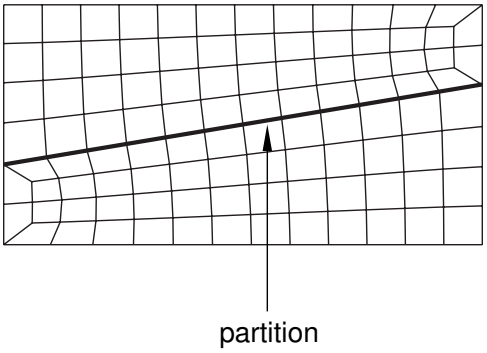


Figure 3: The mesh flows along the partition.

You can use the additional edges created by partitioning a face to control the mesh characteristics. For example, *Figure 4* illustrates how a partition and local mesh seeds allow you to control the mesh flow and density.

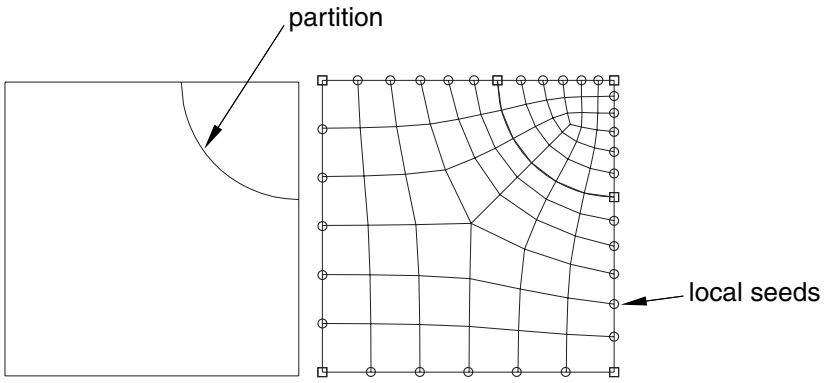


Figure 4: A partition and local mesh seeds allow you to control the mesh flow and density.

Similarly, *Figure 5* shows how partitioning and local mesh seeds allow you to refine the mesh in the area of a stress concentration.

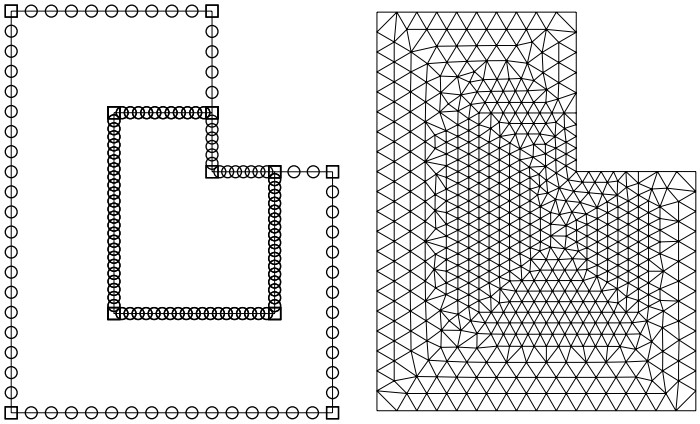


Figure 5: Partitioning and local mesh seeds allow you to refine the mesh in the area of a stress concentration.

In addition, you can apply different mesh controls, such as element shape, to the regions created by a partition.

When partitioning, remember that partitions will become element boundaries. Therefore, try to ensure that partitions make angles as close to 90° as possible with other partitions or edges. In addition, you should avoid creating unwanted short edges that will distort the mesh.

How are seeds and other attributes affected by partitioning?

Seed distributions along edges you have seeded may change during the partitioning process; Abaqus/CAE redistributes the seeds to accommodate any new vertices created by partitioning. For example, the left and right edges of the part instance in *Figure 1* are seeded with seven elements per edge.

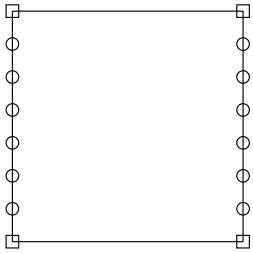


Figure 1: The left and right edges each have seven elements.

If you create a partition that splits the part instance into two regions, new vertices are created at the midpoints of both edges. In *Figure 2* you can see how Abaqus/CAE added seeds at the new vertices so that nodes will exist at the corners of each region.

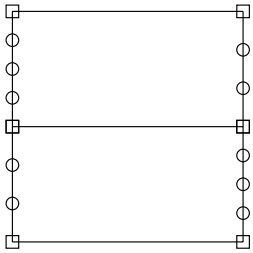


Figure 2: Redistribution of seeds.

Abaqus/CAE also redistributed the existing seeds to eliminate any overly small elements created by the new partition. However, this redistribution can result in seeds that are not aligned. The top region has one seed more on the left side than it does on the right, and the reverse is true for the bottom region. In this example you could change the number of elements along the right and left edges to an even number to ensure that the seeds align after partitioning.

Any other mesh attributes, such as element shape or element type, that you have applied are applied automatically to each new region that you create with a partition. However, once you have created the different regions, you can assign different mesh attributes to each region.

Regenerating partitions after modifying geometry

Partitions are features associated with the part or part instance; therefore, you can modify and regenerate them like any other feature.

For example, consider the partition on the right side of the part instance shown in *Figure 1*.

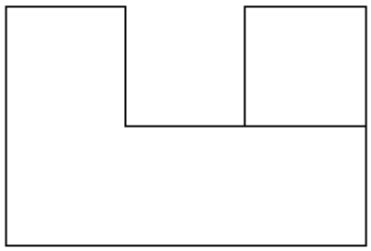


Figure 1: A partitioned part instance.

If you return to the Part module and widen the right side of the model, the partition also expands and continues to divide the face into two regions, as shown in *Figure 2*.

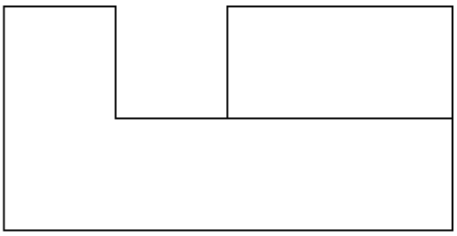


Figure 2: The partition is regenerated.

Sometimes regeneration of a partition creates unmeshable regions. In this situation simply add, modify, or delete partitions until the part instance becomes meshable again.

Using virtual topology to improve your mesh

In some cases parts or independent part instances contain details such as very small faces and edges. The Virtual Topology toolset allows you to remove these small details by combining a small face with an adjacent face or by combining a small edge with an adjacent edge. You can also ignore selected edges and vertices, which has the same effect as combining faces and edges. Introducing virtual topology is a convenient method for creating a clean, well-formed mesh. The Virtual Topology toolset is available only in the Mesh module.

However, adding virtual topology to a part instance can restrict your ability to subsequently mesh the part instance. For example, you cannot mesh regions that contain virtual topology using the following techniques:

- Two-dimensional free meshing with quadrilateral or quadrilateral-dominated elements using the medial axis algorithm.
- Three-dimensional swept meshing using the medial axis algorithm.
- Two-dimensional structured meshing if the region to be meshed is not bounded by four corners.
- Three-dimensional structured meshing if the region to be meshed is not bounded by six sides.

For more information, see *The Virtual Topology toolset*.

In addition, you can apply virtual topology only to independent instances. If you need to apply virtual topology to a dependent instance, you can create a copy of the original part and then create an independent instance of the copy. You can then replace the dependent instance with the new independent instance and apply virtual topology to the independent instance. For more information, see *What is the difference between a dependent and an independent part instance?*.

Using adaptive remeshing to improve your mesh

In many cases you will not know the adequacy of your mesh refinement for your particular solution goal until you have executed a number of analyses and evaluated the solution results. Mesh refinement studies are typically performed in these cases, where a mesh is successively refined and key solution results are confirmed to converge. You can automate this process by applying remeshing rules to regions of interest in your model and using the Abaqus/CAE adaptive remeshing process to automatically perform successive mesh refinement based on a series of executed analyses.

With a remeshing rule you can specify:

- The region where you want the mesh refined.
- The solution quality criteria (for example error indicators in the Mises stress) that mesh refinement is based on.
- The analysis step or steps that refinement is based on.
- Minimum and maximum element size constraints.
- A sizing algorithm and parameters appropriate to your simulation.

For more information, see *Advanced meshing techniques* and *Creating, editing, and manipulating adaptivity processes*.

Understanding mesh generation

This section explains basic concepts and terminology related to meshes and mesh generation.

In this section:

- *Generating a mesh*
- *Preserving the precision of nodal coordinates*
- *Determining which regions are meshable*
- *What should I do if a region fails to mesh?*
- *What is a mesh transition?*
- *What is the difference between the medial axis algorithm and the advancing front algorithm?*
- *What kinds of meshes cannot be generated automatically?*
- *When will Abaqus/CAE delete a mesh?*
- *Do I have to mesh the entire model in one operation?*
- *Can I change the geometric order of the elements in a mesh?*

Generating a mesh

Most meshing in Abaqus/CAE is completed in a "top-down" fashion.
This means that the mesh is created to conform exactly with the geometry of a region and works down to the element and node positions.

1. Abaqus/CAE follows these basic steps to generate a mesh:

   Generate a mesh on each top-down region using the meshing technique currently assigned to that region. By default, Abaqus/CAE generates meshes with first-order line, quadrilateral, or hexahedral elements throughout.

2. Merge the meshes of all regions into a single mesh. Typically, Abaqus/CAE merges the nodes along the common boundaries of neighboring regions into a single set of nodes. However, in certain cases Abaqus/CAE creates tied surface interactions instead of merging these nodes; for example, along the common interface between hexahedral and tetrahedral meshes. For more information, see *Meshing multiple three-dimensional solid regions*.

Top-down meshes generated by Abaqus/CAE conform to the geometry of the part or part instance they discretize, as shown in *Figure 1*:

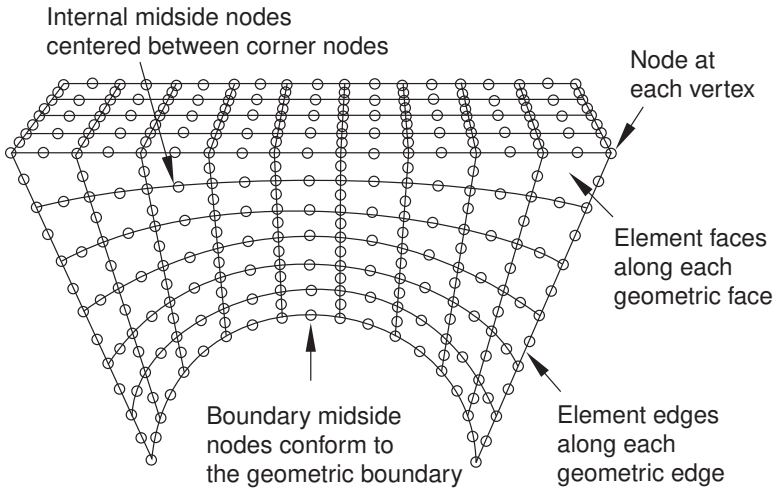


Figure 1: The mesh conforms to the geometry of the part instance.

- A node is generated at each geometric vertex.
- A connected set of element edges is generated along each geometric edge.
- A connected set of element faces is generated along each geometric face.
- Nodes that are on the boundary of the mesh (including the midside nodes of second-order elements) are also on the boundary of the geometry.
- Midside nodes of internal second-order elements are centered between the end nodes of the element edges.

For detailed, step-by-step instructions on creating a top-down mesh, see *Creating a mesh*.

Relying directly on the geometry to form the outer mesh boundaries can impact mesh quality as Abaqus/CAE creates elements to fill small details. In some cases you may not be able to implement a partitioning strategy that allows you to apply a top-down swept or structured meshing technique on a complex region. For solid regions, you can use the "bottom-up" meshing technique in place of the automated top-down meshing techniques to generate a hexahedral mesh. Bottom-up meshing is a manual, incremental meshing process that builds up a three-dimensional mesh from two-dimensional entities. You define the regions that will be meshed using the bottom-up technique, control the meshing process, decide whether the resulting mesh meets your needs, and—since the mesh is not required to conform to the geometry— control the associativity of the geometry to the mesh. For more information on bottom-up meshing, see *Bottom-up meshing*.

Preserving the precision of nodal coordinates

When you create a part in the Part module, it exists in its own coordinate system, independent of other parts in the model. In contrast, when you create an instance of the part in the Assembly module and position it relative to other part instances, you are working in the assembly's global coordinate system.

To preserve precision, the Mesh module separates the positioning information of a part instance from the geometry of the instance. As a result, when you generate a mesh, the nodal coordinates for the part instance are computed relative to the coordinate system of the original part. (When the Job module generates an input file, Abaqus/CAE writes the nodal coordinates for each instance relative to its own coordinate system and passes the instance positioning and orientation information to the analysis product via the **INSTANCE* keyword.)

The Mesh module stores these nodal coordinates in single precision. If the geometry of the part lies far from the origin of its coordinate system, some precision of the nodal coordinates will be lost. To prevent this loss of precision, you should try to position a part close to the origin of its coordinate system. For example, the origin of the coordinate system of an Abaqus/CAE native part is located at the origin of the sketch that defined the base feature. Therefore, if possible, you should position the sketch of the base feature over the origin of the sketcher grid.

Determining which regions are meshable

When the **Mesh defaults** color mapping is selected, the color of a region in the Mesh module indicates the meshing technique currently assigned to that region. The color coding is as follows:

- Structured meshing technique: green
- Free meshing technique: pink
- Swept meshing technique: yellow
- Unmeshable: orange
- Bottom-up meshing technique: light tan

See *Structured meshing and mapped meshing*, *Free meshing*, *Swept meshing*, and *Bottom-up meshing* for information about each meshing technique. See *Color coding geometry and mesh elements* for more information about color mappings.

In many cases Abaqus/CAE can use more than one technique to mesh a region; in these cases you can either accept the default technique, or you can use the **Mesh Controls** dialog box to select an alternative technique. In addition, you can change the meshing techniques that are valid for a region by adding partitions to the region or by assigning a different element shape to the region. For example, if you change the element shape assignment of an unmeshable three-dimensional part instance from hexahedra to tetrahedra, the part instance becomes meshable using the free-meshing technique. For more information, see *Why partition in the Mesh module?*.



Note:

You must use the **Mesh Controls** dialog box to assign the bottom-up meshing technique to a region. To unassign the bottom-up technique, you may select another technique or click **Defaults** in the **Mesh Controls** dialog box to allow Abaqus/CAE to use the default element shape and meshing technique for the region.

The default meshing technique for two-dimensional models is the free meshing technique. If you are not satisfied with the quality of the mesh generated by the free meshing technique, or if you prefer a more regular grid-like mesh pattern, you can assign structured meshing to the simpler regions of your model. However, if your model is large and complex, identifying the simple regions where structured meshing is applicable can be a time-consuming process. To make the process faster, you can apply the structured meshing technique to the entire model, and Abaqus/CAE will do the following:

- Determine if any faces are too complex to be structured meshed and ask if you wish to remove them from your selection.
- Determine if any faces are poorly shaped and will result in unacceptable mesh quality and ask if you wish to remove them from your selection.

If Abaqus/CAE removed any faces from your selection, they are colored pink to indicate that they will be meshed using free meshing. Remaining faces are colored green to indicate that Abaqus/CAE will mesh them using structured meshing.

For example, *Figure 1* shows a shell model of an electrical connector. The user attempted to assign structured meshing to the entire assembly, and Abaqus/CAE removed the indicated faces from their selection.

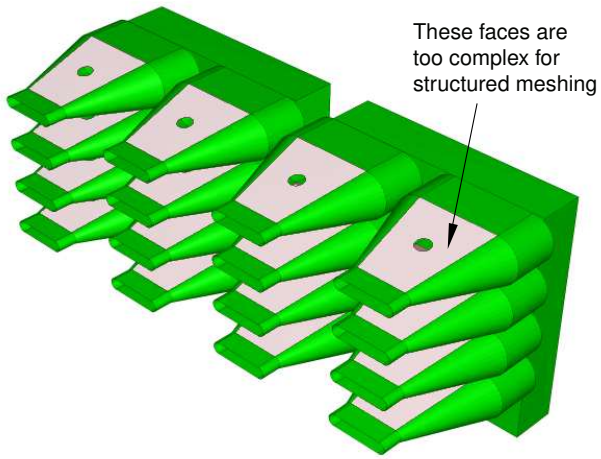


Figure 1: Faces that cannot be structured meshed are removed from the selection.

If you are meshing a solid model, you must select one or more cells and use the **Mesh Controls** dialog box to determine whether the structured technique can be applied to those cells. If you have a region that will be meshed using free tetrahedral meshing, you can select boundary faces and use the **Mesh Controls** dialog box to determine whether the structured technique can be applied to create a triangular boundary mesh prior to tetrahedral meshing of the solid.

For detailed information on controlling the mesh technique and element shape assigned to a region, see the following sections:

- *Bottom-up meshing*
- *Assigning mesh controls*
- *Choosing an element shape*
- *Selecting a meshing technique*
- *Changing mesh controls for previously meshed regions*

What should I do if a region fails to mesh?

If a region fails to mesh, Abaqus/CAE displays an **Error** dialog box that explains why the meshing failed. In most cases Abaqus/CAE highlights the region and allows you to save it in a set. You can create a display group from the set and use the display group to study the region that failed to mesh.

Some of the more common reasons why a region fails to mesh and the associated solutions are as follows:

Inadequate seeding

The region contains some small edges or the seed density is too coarse. You can use the Virtual Topology toolset to merge small edges. Alternatively, if you save the region that failed to mesh in a set, you can apply local seeds of a finer density to the saved set.

When you are creating a hexahedral mesh, a quadrilateral mesh, or a quadrilateral-dominated mesh using the medial axis algorithm, Abaqus may need to alter seeds to generate the mesh. In some cases, mesh generation may fail because the modified seeds density is too coarse. Meshing may succeed if you incrementally mesh the regions of the part in a different order, or, as described above, you can apply local seeds of a finer density and remesh the part.

Bad geometry

Bad geometry refers to small edges or faces or to part instances that are imprecise. You can use the Query toolset to check the geometry. For more information, see *Using the geometry diagnostic tools*.

Poor boundary triangles

When you are creating a free mesh with tetrahedral elements, Abaqus/CAE first creates a triangular mesh on the faces of the region and then uses those triangles as faces of the boundary tetrahedral elements. You can choose to preview the triangular mesh on the faces and decide if it is acceptable before continuing with the time-consuming process of generating tetrahedral elements through the interior of the region. For more information, see *What is a tetrahedral boundary mesh?*.

In some cases Abaqus/CAE cannot complete the conversion from triangles to tetrahedra and highlights the nodes on the boundary mesh that cannot be inserted into the tetrahedral mesh. The highlighted nodes serve as indicators of regions that require attention, and you can try the following:

- Use the seeding tools to increase the mesh density.
- Use the Virtual Topology toolset to combine small faces and edges with adjacent faces and edges.
- Use the Partition toolset to partition regions into simpler subregions.
- Use the Edit Mesh toolset to improve the tetrahedral boundary mesh.
- Use mesh controls to change the mesh technique applied to faces of the solid region.

Gasket regions

You can generate a gasket reinforcement mesh only on a region that contains a gasket mesh.

What is a mesh transition?

A mesh transition is an area where a mesh transitions from coarse (large elements) to fine (small elements), as shown in *Figure 1*.

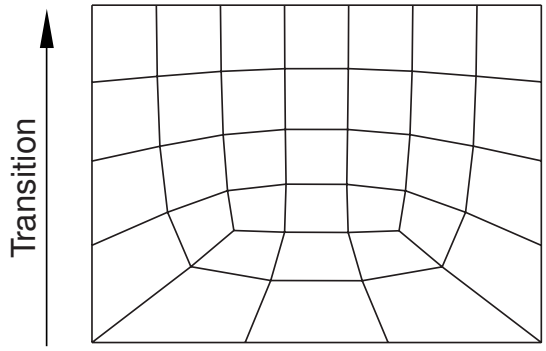


Figure 1: A mesh with a transition from coarse to fine elements.

Abaqus/CAE provides mesh transition controls for the following types of meshes:

- A two-dimensional, quadrilateral-only mesh that is created using the structured meshing technique or the free meshing technique with the medial axis algorithm.
- A three-dimensional, hexahedral-only mesh that is created by sweeping a two-dimensional mesh. For more information, see *What is swept meshing?*.

When transition controls are applicable to the type of mesh you are creating, a toggle button appears on the right side of the **Mesh Controls** dialog box that allows you to minimize the mesh transition. By default, Abaqus/CAE minimizes the mesh transition, which in some cases will reduce mesh distortion. Conversely, if you toggle off the option to minimize mesh transition, the mesh may move closer to the specified mesh seeds. To display the **Mesh Controls** dialog box, select **Mesh->Controls** from the main menu bar. For more information, see *Setting the mesh algorithm*.

What is the difference between the medial axis algorithm and the advancing front algorithm?

The medial axis algorithm and the advancing front algorithm are two meshing schemes that Abaqus/CAE can use to generate a mesh when you are doing the following:

- Meshing a surface with quadrilateral or quadrilateral-dominated elements using the free meshing technique.
- Meshing a solid region with hexahedral or hexahedral-dominated elements using the swept meshing technique. (Abaqus/CAE generates hexahedral and hexahedral-dominated meshes by sweeping the quadrilateral and quadrilateral-dominated elements generated by the two algorithms from the source side to the target side.)

The two algorithms are described as follows:

Medial axis

The medial axis algorithm first decomposes the region to be meshed into a group of simpler regions. The algorithm then uses structured meshing techniques to fill each simple region with elements. If the region being meshed is relatively simple and contains a large number of elements, the medial axis algorithm generates a mesh faster than the advancing front algorithm. Minimizing the mesh transition may improve the mesh quality. The mesh transition option is available only for quadrilateral and hexahedral meshing. For more information, see *What is a mesh transition?*.

Advancing front

The advancing front algorithm generates quadrilateral elements at the boundary of the region and continues to generate quadrilateral elements as it moves systematically to the interior of the region.

The elements generated by the advancing front algorithm will always follow the seeding exactly for quadrilateral-dominated and hexahedral-dominated meshes (except when you are creating a three-dimensional revolved mesh, and the profile being revolved touches the axis of revolution). For other meshes the elements generated by the advancing front algorithm will always follow the seeding more closely than those generated by the medial axis algorithm. If the region to be meshed contains virtual topology, you can use only the advancing front algorithm to generate the mesh.

If you select the advancing front algorithm, you can allow Abaqus/CAE to use mapped meshing where appropriate. (Mapped meshing is the same as structured meshing but applies to only four-sided regions.) For more information, see *What is mapped meshing?*, and *When can Abaqus/CAE apply mapped meshing?*.

You may have to experiment with the two algorithms to obtain the optimal mesh. *Figure 1* illustrates a simple shell region that was meshed with quadrilateral-dominated elements using the two meshing algorithms. In this example both algorithms generate an acceptable mesh.

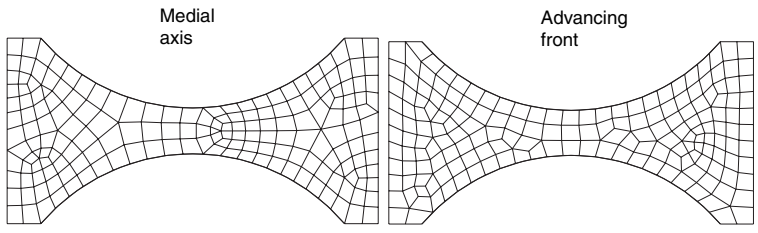


Figure 1: Both algorithms generate acceptable meshes.

Because the elements produced by the advancing front algorithm follow your seeds, the resulting mesh may include some skew in the elements in narrow regions. Element skew is illustrated in *Figure 2*.

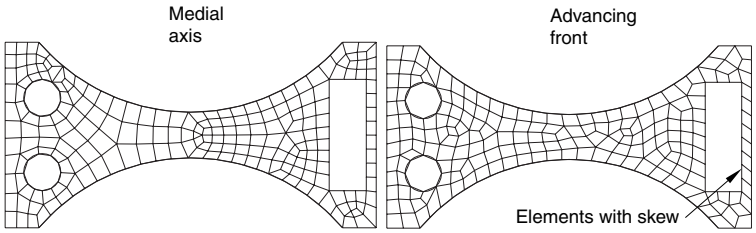


Figure 2: In some cases the advancing front algorithm generates elements with some skew.

In contrast, the advancing front algorithm may generate elements of a more uniform size with a more consistent aspect ratio, as shown in *Figure 3*. Uniform element size can play an important role in the analysis; for example, if you are creating a mesh for an Abaqus/Explicit analysis, small elements in the mesh can unduly control the size of the time step. In addition, if it is important that the elements follow your seeds, the advancing front algorithm is preferable.

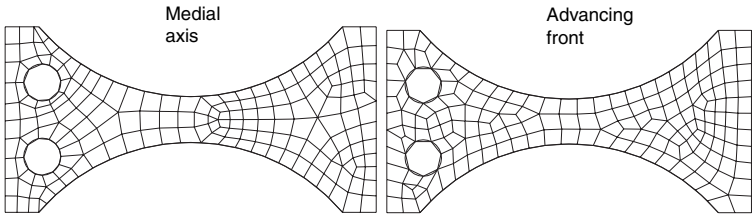


Figure 3: In some cases the advancing front algorithm produces a more uniform mesh.

In some cases, when you mesh multiple regions, Abaqus/CAE generates a mesh with sheared elements at the interface between regions. Nodes in one region may be positioned differently than nodes in an adjacent region, which results in shear at the common boundary when Abaqus/CAE merges the adjacent meshes. *Figure 4* shows multiple swept regions and the resulting mesh generated by the medial axis algorithm.

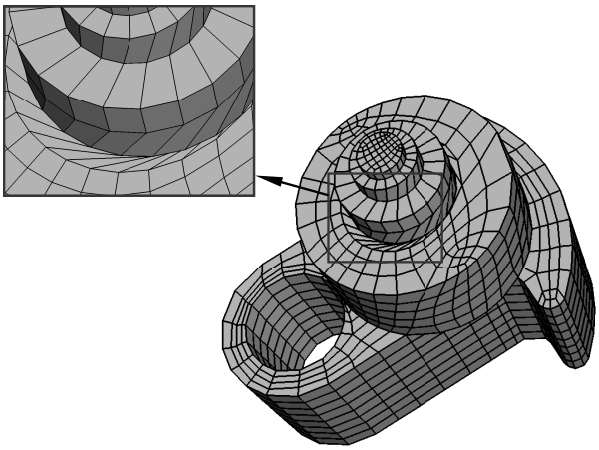


Figure 4: Mesh shear is significant between adjacent regions using the medial axis algorithm.

The advancing front algorithm positions the nodes on the source side at the same location as your seeds; as a result, the mesh shear will be reduced. *Figure 5* shows the same part meshed with the same seeding using the advancing front algorithm. However, as stated earlier, you may have to experiment with the two algorithms to obtain the optimal mesh.

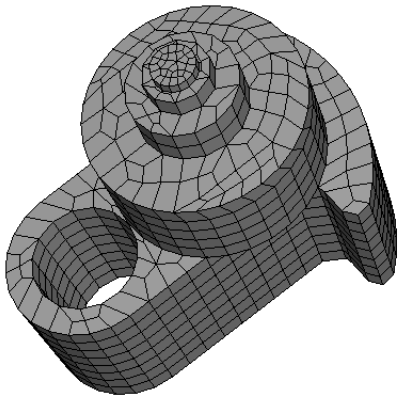


Figure 5: Mesh shear is reduced between adjacent regions using the advancing front algorithm.

Additional information

• *Setting the mesh algorithm*

What kinds of meshes cannot be generated automatically?

There are a few types of meshes that you cannot create using the mesh generator in the Mesh module:

Compatible meshes between part instances of the same assembly

Compatibility means that the element faces or element edges of the meshes of adjacent part instances share the same nodes and have the same topology at the common interface. You cannot prescribe mesh compatibility between instances. However, you can use the **Merge/Cut** tool in the Assembly module to merge multiple instances into a single part instance. You can then create a single compatible mesh from the single part instance. See *Compatible meshes between part instances*, for more information.

Symmetric meshes

You cannot ensure that Abaqus/CAE will generate a symmetric mesh for a symmetric part or part instance.

Bottom-up meshes

Bottom-up meshing is a manual process. You must set and apply the parameters to create the mesh for each region to which you have assigned the bottom-up meshing technique.

When will Abaqus/CAE delete a mesh?

The following attributes of a part, part instance, or region affect how the mesh will be generated:

- Seeding
- Element shape
- Meshing technique
- Meshing algorithm
- Logical corners of a two-dimensional structured region
- Transition control
- Sweep path of a swept region

If you change any of the attributes listed, the existing mesh will no longer be consistent with its attributes. As a result, Abaqus/CAE deletes the mesh, and you can recreate a new mesh that matches the new attributes. (Element order and element family are the only attributes that, when changed, do not require the mesh to be deleted and recreated.)

Whenever you make a change that will affect any of these attributes for a top-down meshed region, Abaqus/CAE displays a dialog box similar to the one shown in *Figure 1*.

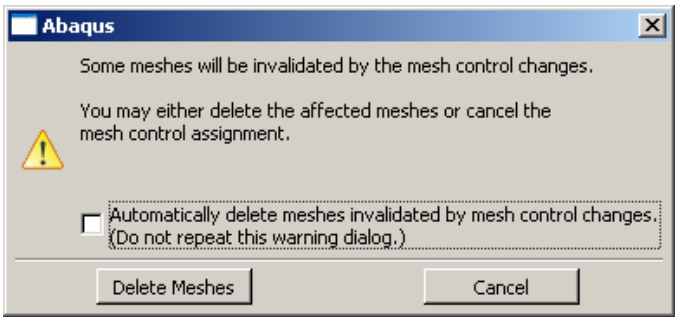


Figure 1: The warning dialog box.

You can delete the mesh by clicking the **Delete Meshes** button, or you can keep your mesh and exit the procedure by pressing the **Cancel** button.

You can also avoid this warning message for the remainder of the current session by toggling **Automatically delete meshes invalidated by mesh control changes**. The next time you attempt to change the attributes of a part, part instance, or region that already contains a mesh, the mesh will be deleted immediately without any warning being displayed.

If you save your model to a model database before you delete the mesh, you can revert back to that mesh if you are dissatisfied with later meshing attempts.

Since bottom-up meshes can be very time consuming to create, Abaqus/CAE attempts to preserve them in many circumstances that would cause a top-down mesh to be deleted. Bottom-up meshes are retained during changes to seeding, partitioning, and virtual topology. When you partition a region or create a virtual topology feature the mesh will be retained, but associativity with the geometry effected by the partition or virtual topology operation will be lost.

Warning:

There is no way to avoid deleting a top-down mesh when you change one of the meshing attributes listed above. Likewise, if you change the part geometry, Abaqus/CAE always deletes both top-down and bottom-up meshes without warning. Since remeshing can be time consuming for large or complex models, you should use caution when changing these attributes.

For detailed, step-by-step instructions on deleting a mesh, see *Deleting a mesh*.

Additional information

- *Mesh-geometry association*
- *Seeding previously meshed parts, part instances, or regions*
- *Changing mesh controls for previously meshed regions*

Do I have to mesh the entire model in one operation?

Abaqus/CAE allows you to mesh the model in an incremental fashion, where each meshing operation meshes a different region of the model.
You can use incremental meshing to fine-tune the mesh in a selected region of your model without remeshing the entire model.

When you mesh a selected region, Abaqus/CAE tries to preserve the existing mesh in other regions of the model if possible. However, incremental meshing may force the nodes on the boundaries of the existing mesh to move and can reduce the mesh quality along the interfaces between the regions. If your part or part instance contains a seam crack, you must mesh the part or part instance in a single operation; incremental meshing is not supported.

In some cases Abaqus/CAE cannot proceed with an incremental meshing operation and must delete all the existing meshes before proceeding:

- Incremental meshing cannot proceed if the seeding between the existing mesh and the selected region cannot be honored. You must allow Abaqus/CAE to delete the existing mesh and remesh the original regions and the selected region.

  For example, consider the part instance in *Figure 1*.

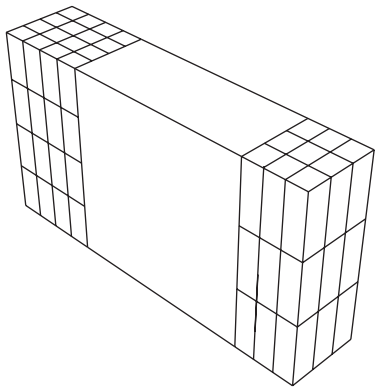


Figure 1: The central region cannot be meshed incrementally.

  The central region cannot be meshed incrementally because one end has a mesh with 4 × 4 mesh pattern and the opposite end has a mesh with a 3 × 3 mesh pattern. If you try to mesh only the central region, Abaqus/CAE will detect the problem and allow you to choose between the following:

  - Remesh the regions that are already meshed and the central region to generate a compatible mesh.
  - Cancel the operation to mesh the central region.

- Incremental meshing cannot proceed if the existing mesh needs to be derived from the mesh you are trying to create. For example, consider the part instance in *Figure 2*.

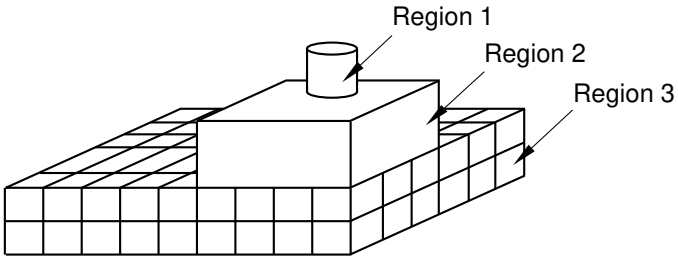


Figure 2: The regions must be meshed in the correct sequence.

To create a compatible mesh between region 1 and region 2, the mesh of region 2 is derived from the mesh of the cylinder in region 1. Similarly, the mesh of region 3 is derived from the mesh of region 2, which in turn was derived from the mesh of the cylinder in region 1. As a consequence, if you mesh region 3 first, Abaqus/CAE cannot incrementally mesh regions 1 and 2. You must allow Abaqus/CAE to mesh region 1 prior to remeshing regions that were already meshed.

If incremental meshing cannot proceed, Abaqus/CAE displays a warning message prior to deleting an existing mesh.

If you want to mesh the part or assembly incrementally, you can follow a strategy that will minimize the number of times Abaqus/CAE has to delete the existing mesh. The meshing strategy depends on the topology of the regions, the element shapes, the meshing technique, and the mesh seeding.

- Changes to the seeding always propagate out to the boundaries. As a result, you should start meshing from the interior of the part or part instance and continue out to the boundaries of the part or instance.
- However, if you can identify a set of adjacent three-dimensional regions that will be meshed using the swept method, you should start meshing on one side of the part or part instance and continue the mesh through the interior to the other side of the part or instance.
- Regions that are meshed by triangles or tetrahedral elements will never force the entire mesh to be deleted during incremental meshing. The same is also true for regions that are meshed by quadrilateral-dominated elements using the advancing front algorithm. Abaqus/CAE can always remesh these regions, and you can mesh them at any time.

Can I change the geometric order of the elements in a mesh?

If you have already meshed a part, a part instance, or a region, Abaqus/CAE allows you to change the element order without having to recreate the entire mesh. If you change between linear and quadratic elements, Abaqus/CAE simply adds or removes the midside nodes as required.

If the part or part instance contains orphan elements, you can change the order of all the elements or you can change the order of only selected elements. Orphan elements contain no underlying geometry information; therefore, you should take care when changing the order of the elements. If you change orphan elements from quadratic to linear, all information on the location of the midside nodes is lost. As a result, if you subsequently decide to change back from linear to quadratic elements, you will not be able to return to the original mesh. For more information, see *What kinds of files can be imported and exported from Abaqus/CAE?*.

Structured meshing and mapped meshing

This section describes the structured and mapped meshing techniques and the types of regions to which these techniques can be applied.

In this section:

- *What is structured meshing?*
- *What is mapped meshing?*
- *Two-dimensional structured meshing*
- *Three-dimensional structured meshing*
- *Using structured meshing near concave boundaries*
- *When can Abaqus/CAE apply mapped meshing?*

What is structured meshing?

The structured meshing technique generates structured meshes using simple predefined mesh topologies. Abaqus/CAE transforms the mesh of a regularly shaped region, such as a square or a cube, onto the geometry of the region you want to mesh. For example, *Figure 1* illustrates how simple mesh patterns for triangles, squares, and pentagons are applied to more complex shapes.

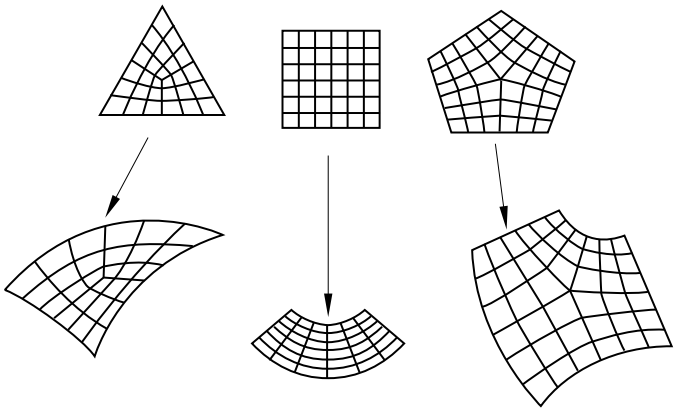


Figure 1: Two-dimensional structured mesh patterns.

You can apply the structured meshing technique to simple two-dimensional regions (planar or curved) or to simple three-dimensional regions that have been assigned the **Hex** or **Hex-dominated** element shape option. For more information about assigning element shapes to a region, see *Choosing an element shape*.

What is mapped meshing?

The terms structured meshing and mapped meshing are used interchangeably in the finite element analysis literature. However, Abaqus/CAE makes a subtle distinction between the two terms. Mapped meshing is a subset of structured meshing. Mapped meshing refers only to structured meshing of four-sided, two-dimensional regions—the square mesh pattern in *Figure 1*.

Some models that appear very complex actually contain faces with relatively simple geometry. When you mesh such a model with free or swept meshing, the resulting element quality can be poor on these faces. However, if you allow Abaqus/CAE to use the mapped meshing technique where the geometry is appropriate, it often generates elements of good quality, especially if the region is a long, thin rectangular face.

You cannot apply mapped meshing directly to a region. However, you can apply it indirectly by meshing a region and allowing Abaqus/CAE to apply mapped meshing where appropriate. For example, *Figure 1* shows the effect of free meshing a part and allowing Abaqus/CAE to use mapped meshing where appropriate.

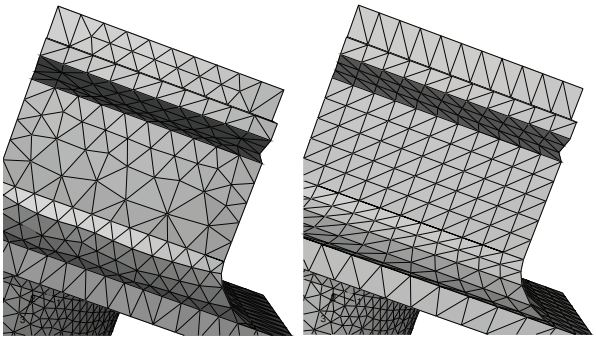


Figure 1: The effect of allowing mapped meshing.

By default, Abaqus/CAE uses mapped meshing where appropriate when you are doing the following:

- Using the advancing front algorithm to sweep mesh a solid region with hexahedral or hexahedral-dominated elements.
- Using the advancing front algorithm to free mesh a shell region with quadrilateral or quadrilateral-dominated elements.
- Free meshing a solid region with tetrahedral elements.
- Free meshing a shell region with triangular elements.

Two-dimensional structured meshing

A two-dimensional region can be meshed using the structured meshing technique if it has the following characteristics:

- The region has no holes, isolated edges, or isolated vertices. *Figure 1* shows regions that cannot be structured meshed.

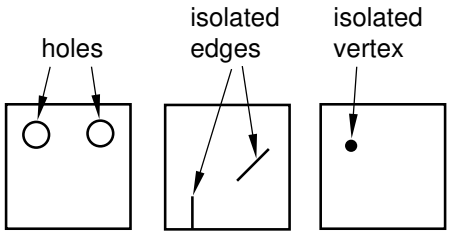


Figure 1: Regions that cannot be structured meshed.

- The region is bounded by three to five logical sides, where each side is a connected set of edges.

In general, structured meshing gives you the most control over the mesh that Abaqus/CAE generates. If you are meshing a four-sided region with all quadrilateral elements, the total number of element edges around the boundary must be even. For three- and five-sided regions, the constraint equations are more complex. Abaqus/CAE respects seed distribution wherever possible when generating a structured mesh. (Seed distribution describes the spacing of the seeds, not necessarily the number of seeds. For example, are the seeds evenly spaced along an edge or more concentrated at one end?) However, meshes must be compatible across regions, and Abaqus/CAE may adjust the nodes of a mesh region that is adjacent to a region that was meshed using the free meshing technique. As a result, the element nodes may not match the seeds exactly.

Figure 2 shows the seeds on the edges of a four-sided region and the effect of different mesh controls.

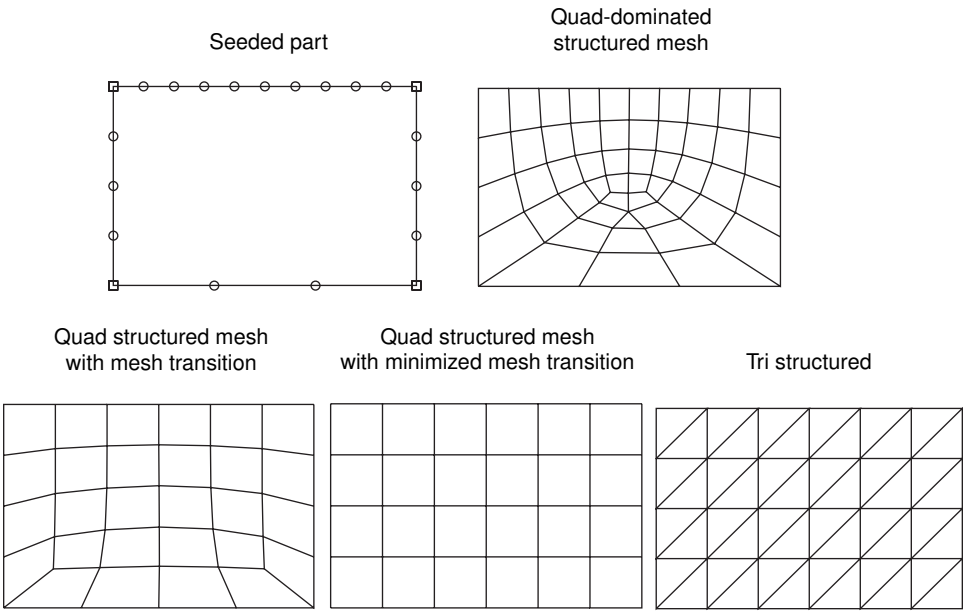


Figure 2: The effect of different mesh controls.

The mesh control effects are as follows:

- In four-sided regions the quadrilateral-dominated structured mesh matches the seeding exactly. There are an odd number of elements around the boundary, and Abaqus/CAE inserts a single triangle into the mesh. (In contrast, when you mesh a three- or a five-sided region with structured quadrilateral-dominated elements, Abaqus/CAE does not insert any triangles. The resulting mesh uses all quadrilateral elements; however, the resulting mesh may not match the mesh seeding exactly.)
- The two quadrilateral structured meshes do not match the seeding. This is even more apparent when you choose to minimize the mesh transition.
- The triangular structured mesh also does not match the seeding. Abaqus/CAE creates the triangular mesh by splitting the diagonals of the quadrilateral structured mesh with minimized mesh transition.

Abaqus/CAE combines edges into a logical side automatically if the edges subtend a shallow angle. For example, each region in *Figure 3* has five edges.

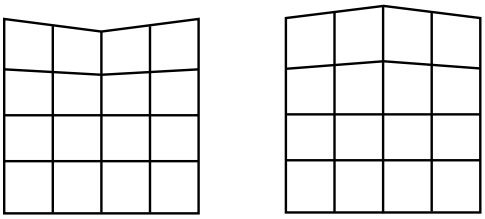


Figure 3: Edges subtending shallow angles.

However, since the top two edges in each region subtend a shallow angle, Abaqus/CAE considers these two edges to be one logical side. Therefore, the mesh pattern for four-sided regions is applied to these regions. If the region contains virtual topology, you can mesh the region using the structured meshing technique only if the region is bounded by four sides. You cannot use structured meshing to mesh three- or five-sided regions that contain virtual topology.

You can use the **Redefine Region Corners** button in the **Mesh Controls** dialog box to combine edges yourself, regardless of the angle they subtend. (To display the **Mesh Controls** dialog box, select **Mesh->Controls** from the main menu bar.) This technique allows you to control which structured mesh pattern is applied to the two-dimensional region. (This technique is not available for three-dimensional regions.) For more information, see *Redefining region corners*.

The region that you plan to mesh with structured quadrilateral elements must be well shaped; otherwise, Abaqus/CAE may create invalid elements as shown in *Figure 4*.

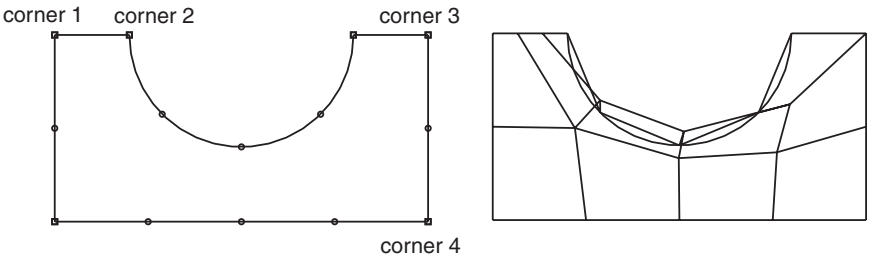


Figure 4: Regions must be well shaped.

If the mesh contains invalid elements, you can use several techniques to correct the mesh:

- Adjust the position of the mesh seeds.
- Redefine the region corners.
- Partition the face into smaller, better shaped regions.

The result of applying each technique is illustrated in *Figure 5*.

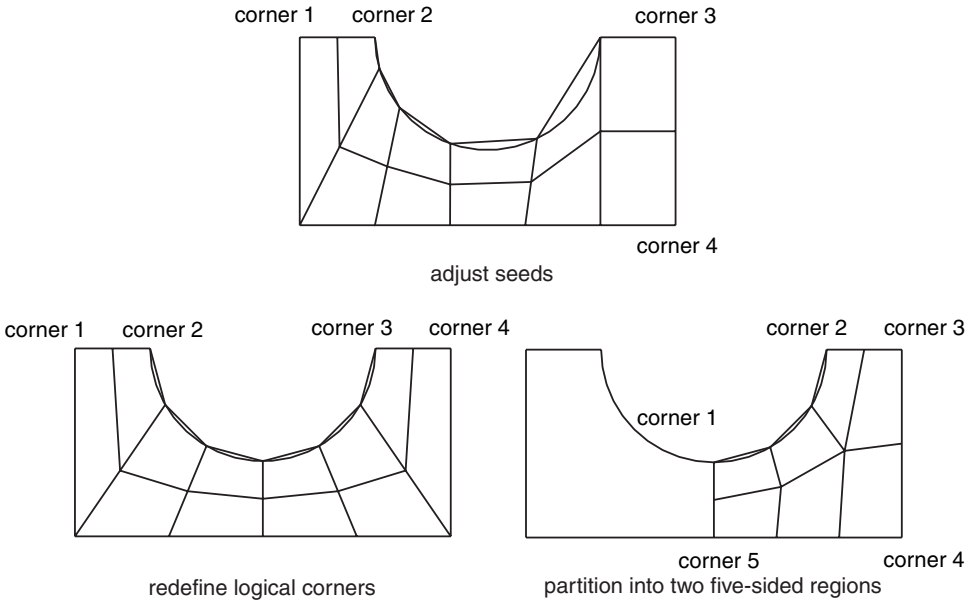


Figure 5: Correcting a mesh that contains invalid elements.

Three-dimensional structured meshing

Figure 1 illustrates examples of simple three-dimensional regions that can be meshed using the structured meshing technique.

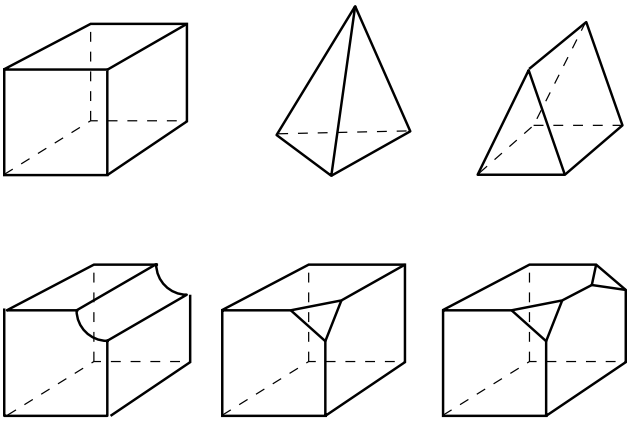


Figure 1: Regions that can be meshed using the structured meshing technique.

Meshing more complex regions with this technique may require manual partitioning. If you do not partition a complex region, your only meshing option may be the free meshing technique with tetrahedral elements. Meshes constructed using the structured meshing technique consist of hexahedral elements, which are preferred over tetrahedral elements.

The characteristics described below are required to mesh a three-dimensional region successfully using the structured meshing technique:

- The region cannot have any holes, isolated faces, isolated edges, or isolated vertices. For example, the regions shown in *Figure 2* cannot be meshed using the structured meshing technique.

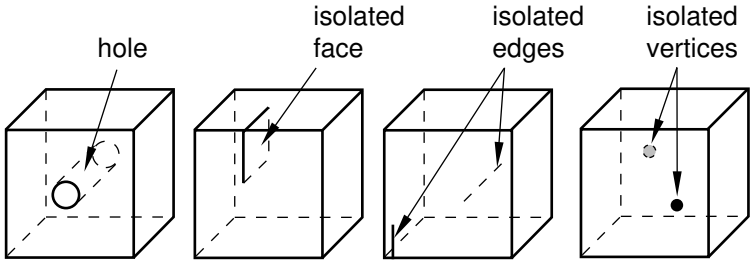


Figure 2: Regions that cannot be meshed using the structured meshing technique.

You can eliminate holes (whether they pass all the way through the part instance or just part way through) by partitioning their circumferences into halves, quarters, etc. For example, the four partitions in *Figure 3* convert the part instance from one region with a hole to four regions without holes.

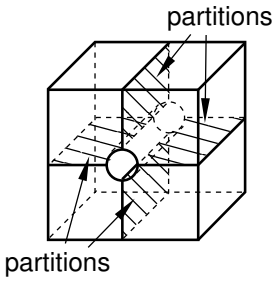


Figure 3: Partitions can make a part structured meshable.

- You should limit arcs to 90° or less to avoid concavities along sides and at edges. For example, the part instance in *Figure 4* has been partitioned so that the single region with 180° arcs becomes two regions with 90° arcs.

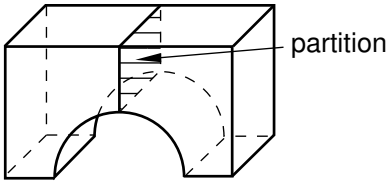


Figure 4: Limit arcs to 90° or less.

- All the faces of the region must have geometries that could be meshed using the two-dimensional structured meshing technique. For example, without partitioning, the semicircles at either end of the part in *Figure 5* have only two sides each. (A face must have at least three sides to be meshed using the structured meshing technique.) If you partition the part into two halves, each semicircle is divided into two faces with three sides each.

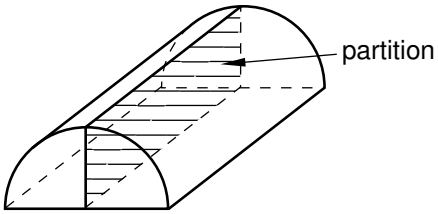


Figure 5: Partitioning creates two faces with three sides.

- Exactly three edges of the region must meet at each vertex. For example, the vertex at the top of an unpartitioned pyramid in *Figure 6* is connected to four edges. However, if you partition the pyramid into two tetrahedral regions, the vertex is connected to only three edges for each individual region.

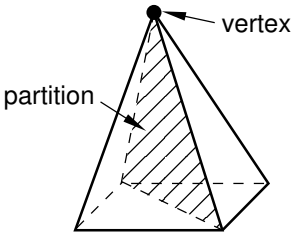


Figure 6: After partitioning the vertex is connected to only three edges for each individual region.

- The region must be bounded by at least four sides (a tetrahedral region). If a region is bounded by fewer than four sides, you can partition the region as necessary to create additional sides.
- If a region contains virtual topology, the region must be bounded by six sides.
- If a region cannot be meshed using the structured meshing technique, you can use virtual topology to combine faces until the region is bounded by six sides. *Figure 7* shows how you can use virtual topology to create a six-sided region that can be meshed using the structured meshing technique.

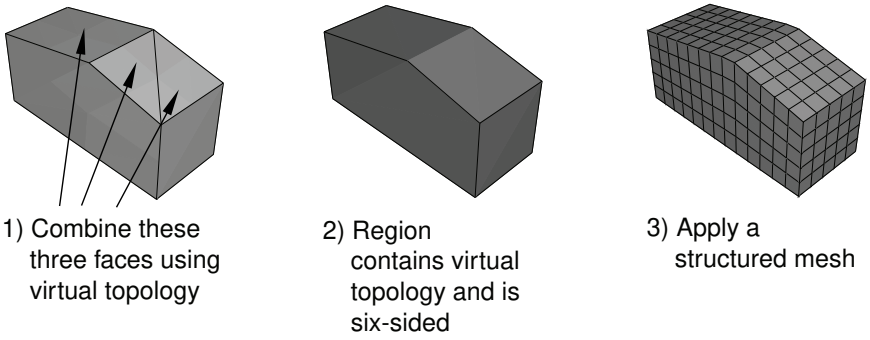


Figure 7: Virtual topology can make a part structured meshable.

- The angles between sides should be as close to 90° as possible; you should partition to eliminate angles greater than 150°.
- Each side of the region must match one of the following definitions:
  - If the region is not a cube, a side must correspond to a single face; that is, the side must not contain multiple faces.
  - If the region is a cube, a side can be a connected set of faces that are on the same geometric surface. However, each face must have four sides. In addition, the pattern of the faces must allow rows and columns of hexahedral elements to be created in a regular grid pattern along that entire side when the cube is meshed. For example, *Figure 8* shows two acceptable face patterns and the resulting regular grid pattern of elements created by meshing the cubes using the structured meshing technique.

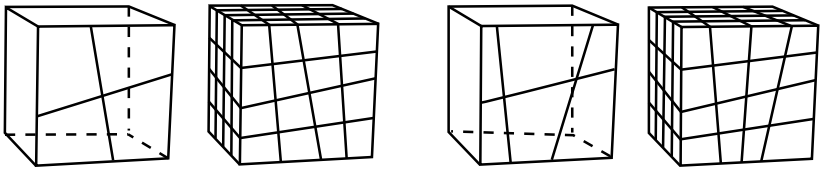


Figure 8: Acceptable face patterns and the resulting meshes.

The sides in *Figure 9* do not have acceptable face patterns:

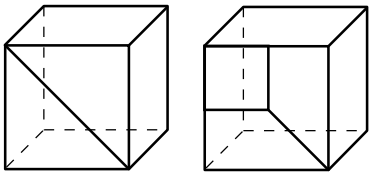


Figure 9: Unacceptable face patterns.

The face pattern shown on the left is unacceptable for structured meshing because each face has only three sides. Each face in the pattern shown on the right has four sides, but the pattern does not allow a regular grid of elements to be created on the partitioned side of the cube, as shown in *Figure 10*.

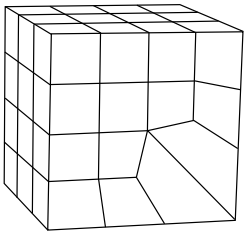


Figure 10: A regular grid of elements cannot be created.

Using structured meshing near concave boundaries

When you mesh a region using any meshing technique, the nodes on the boundary of the mesh are always located on the boundary of the geometric region. However, when Abaqus/CAE creates a mesh using the structured meshing technique, it is possible for nodes in the interior of the mesh to fall outside the region's geometry, which results in a distorted, invalid mesh. This problem typically occurs near concave boundaries.

For example, the region in *Figure 1* has five sides; therefore, when Abaqus/CAE meshes this region using the structured meshing technique, it applies the mesh pattern for a regular pentagon to the region.

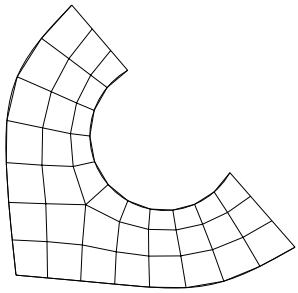


Figure 1: The mesh pattern for a regular pentagon is applied to the region.

However, if you seed the region so that the number of elements is reduced, as shown in *Figure 2*, a distorted mesh results due to the concavity at the highly curved edge. Nodes from the interior of the mesh pattern (indicated by closed circles in *Figure 3*) fall outside the region's geometry, while nodes on the boundary of the mesh (indicated by open circles in *Figure 3*) remain on the boundary of the region's geometry.

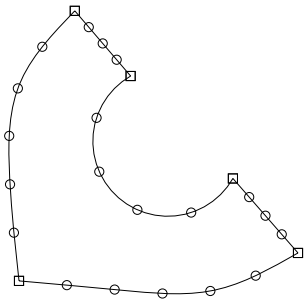


Figure 2: Seeds prescribing a coarser mesh.

When interior nodes fall outside the region's geometry, you can try the following techniques to improve the mesh:

- Change the mesh seeds and remesh. For example, the number of elements along the highly curved edge in *Figure 1* is greater than in *Figure 3*.

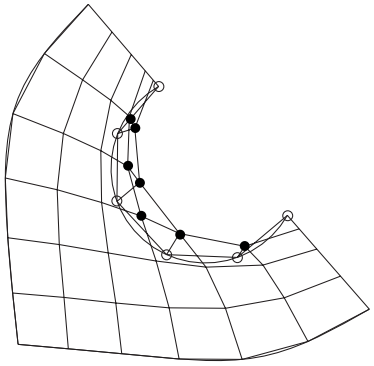


Figure 3: Nodes from the interior of the mesh fall outside the region's geometry.

- Partition the part instance into smaller, more regularly shaped regions. For example, the model was partitioned into three regions in *Figure 4*.

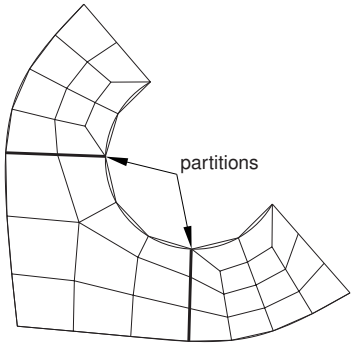


Figure 4: Partition the region.

- Select a different meshing technique. This option is most useful for two-dimensional regions, where you can switch from structured meshing to free meshing and still retain quadrilateral elements in the mesh. (Three-dimensional free meshing is limited to tetrahedral elements. For more information, see *Free meshing*.) *Figure 5* shows the region meshed using the free meshing technique.

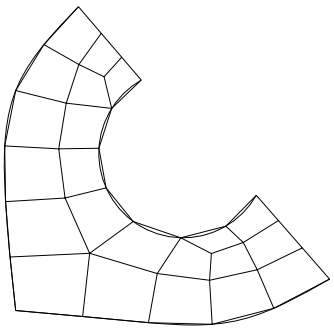


Figure 5: Mesh the region using the free meshing technique.

The mesh in *Figure 5* is not symmetric, which is typical of free meshes.

When can Abaqus/CAE apply mapped meshing?

Abaqus/CAE decides that mapped meshing is appropriate in a four-sided region when:

- it is likely that a regular mesh pattern with reasonable quality can be generated, and
- any minor adjustments to the mesh seeding will not violate the user's intent (as indicated by any existing seed constraints).

When Abaqus/CAE applies mapped meshing, it makes small adjustments to the mesh seeding to ensure that opposite sides of the rectangular region have the same number of seeds. If your model is large and includes many simple regions, it may take slightly longer for Abaqus/CAE to check for rectangular regions and adjust the seeds to produce a mapped mesh than to produce a mesh without mapped meshing. However, for most models the time difference to include mapped meshing is not significant compared to the improvement in mesh quality.

Figure 1 shows a shell part meshed with triangles using the following mesh techniques and algorithms available in Abaqus/CAE:

- Free meshing
- Free meshing using mapped meshing where appropriate
- Structured meshing

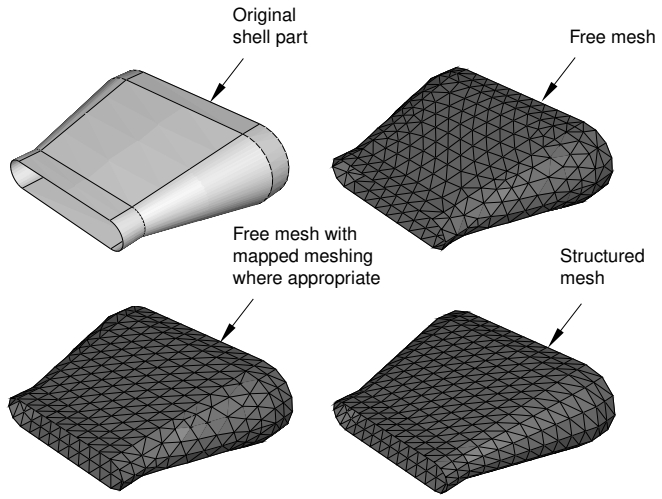


Figure 1: A shell part meshed with triangles in three different ways.

In many cases a free mesh of triangles with mapped meshing where appropriate will be the same as a structured mesh of triangles. The meshes are different in *Figure 1* because Abaqus/CAE honors the original seeding and determines that mapped meshing is not appropriate on the side of the part. In contrast, when Abaqus/CAE creates the structured mesh, it significantly adjusts the seeding to create the structured mesh that was requested by the user.

Swept meshing

This section explains the swept meshing technique and describes the types of regions to which this meshing technique can be applied.

In this section:

- *What is swept meshing?*
- *Swept meshing of surfaces*
- *Swept meshing of three-dimensional solids*
- *Swept meshing of cylindrical solids*
- *Characteristics of the geometry can prevent a part from being swept meshable*

What is swept meshing?

Abaqus/CAE uses swept meshing to mesh complex solid and surface regions. The swept meshing technique involves two phases:

- Abaqus/CAE creates a mesh on one side of the region, known as the source side.
- Abaqus/CAE copies the nodes of that mesh, one element layer at a time, until the final side, known as the target side, is reached. Abaqus/CAE copies the nodes along an edge, and this edge is called the sweep path. The sweep path can be any type of edge—a straight edge, a circular edge, or a spline. If the sweep path is a straight edge or a spline, the resulting mesh is called an extruded swept mesh. If the sweep path is a circular edge, the resulting mesh is called a revolved swept mesh.

For example, *Figure 1* shows an extruded swept mesh. To mesh this model, Abaqus/CAE first creates a two-dimensional mesh on the source side of the model. Next, each of the nodes in the two-dimensional mesh is copied along a straight edge to every layer until the target side is reached.

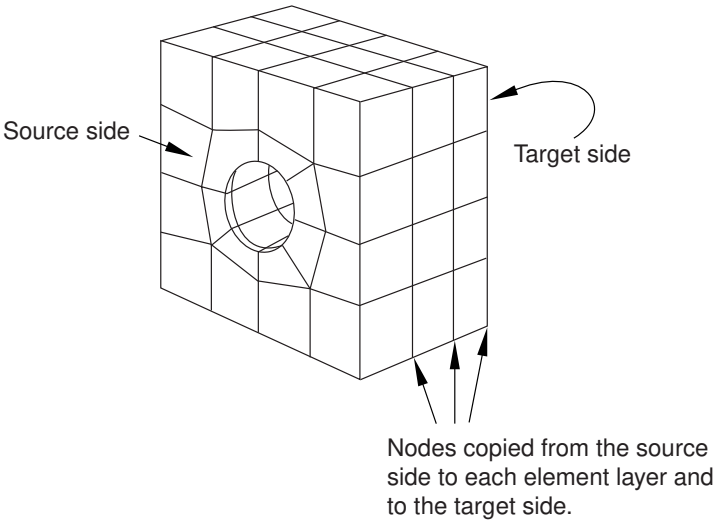


Figure 1: The swept meshing technique for an extruded solid.

To determine if a region is swept meshable, Abaqus/CAE tests if the region can be replicated by sweeping a source side along a sweep path to a target side. In general, Abaqus/CAE selects the most complex side (for example, the side that has an isolated edge or vertex) to be the source side. In some cases you can use the mesh controls to select the sweep path. If some regions of a model are too complex to be swept meshed, Abaqus/CAE asks if you want to remove these regions from your selection before it generates a swept mesh on the remaining regions. You can use the free meshing technique to mesh the complex regions, or you can partition the regions into simplified geometry that can be structured or swept meshed.

When you assign mesh controls to a region, Abaqus/CAE indicates the direction of the sweep path and allows you to control the direction. If the region can be swept in more than one direction, Abaqus/CAE may generate a very different two-dimensional mesh on the faces that it can select as the source side. As a result, the direction of the sweep path can influence the uniformity of the resulting three-dimensional swept mesh, as shown in *Figure 2*.

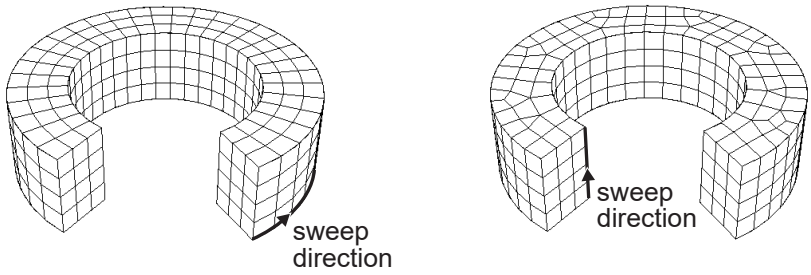


Figure 2: The sweep direction can influence the uniformity of the swept mesh.

In addition, the sweep path controls the default orientation of hexahedral and wedge elements that are used to model gaskets, continuum shells, cylindrical regions using cylindrical elements, and adhesive joints using cohesive elements. For more information, see *Assigning gasket elements to a region*, *Meshing parts with continuum shell elements*, *Swept meshing of cylindrical solids*, and *Creating a model with cohesive elements using geometry and mesh tools*.

Swept meshing of surfaces

Abaqus/CAE can apply the swept meshing technique only to surface regions that can be replicated by sweeping a source side along a sweep path to a target side. The sweep path is always an edge; however, for a surface region the source and target sides are also edges. The surface region can be extruded, revolved, swept, or planar. In addition, an extruded surface region can include twist, and a revolved surface region can include translation. You can apply the swept meshing technique to surface regions using either the **Quad** or **Quad-dominated** element shape options.

If you are creating a revolved swept mesh, Abaqus/CAE meshes the source side and revolves that mesh around the axis of revolution to the source side. The sweep path can be revolved a full 360°. You must use the **Quad-dominated** element shape option when the source side touches the axis of revolution at a point, because a layer of triangular elements is generated at that point. Abaqus/CAE cannot generate a revolved surface mesh when a single surface touches the axis of revolution at two points, unless the revolved surface is a sphere.

For example, the source side touches the axis of revolution at the top of the model shown in *Figure 1* and a layer of triangular elements is generated at that point.

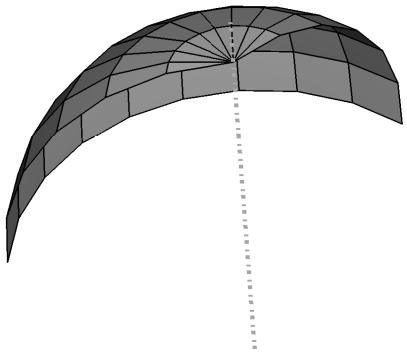


Figure 1: A layer of triangular elements.

(For more information on assigning element shapes to a region, see *Choosing an element shape*.)

Swept meshing of three-dimensional solids

Abaqus/CAE can apply the swept meshing technique to solid regions that can be replicated by sweeping a source side along an edge to the target side. For a three-dimensional solid the sweep path is an edge, but the source and target sides are faces. *Figure 1* illustrates an extruded swept mesh—Abaqus/CAE meshes the source side and extrudes that mesh along an edge to the target side. *Figure 2* illustrates a revolved swept mesh—Abaqus/CAE meshes the source side and revolves that mesh about the axis of the circular edge to the target side.

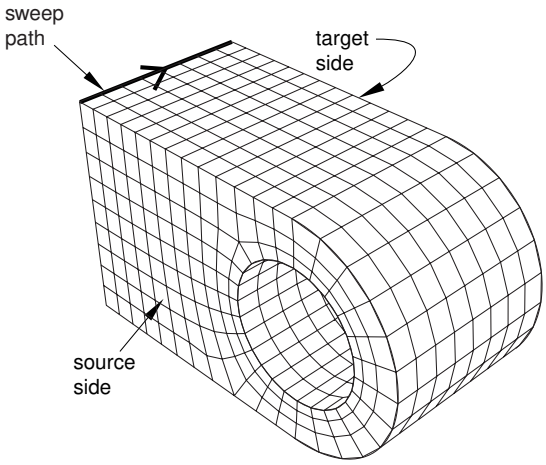


Figure 1: The extruded swept meshing technique sweeps the mesh on the source side along an edge.

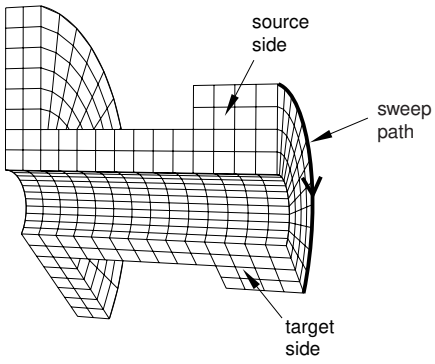


Figure 2: The revolved swept meshing technique sweeps the mesh on the source side along a circular edge.

If a region is swept meshable, Abaqus/CAE can generate the swept mesh on a region that has been assigned the **Hex**, **Hex-dominated**, or **Wedge** element shape option. To generate the preliminary two-dimensional mesh on the source side, Abaqus/CAE uses the free meshing technique with the **Quad**, **Quad-dominated**, or **Tri** element shape option, respectively.

You can choose between the medial axis and advancing front meshing algorithms when you mesh a solid region with hexahedral or hexahedral-dominated elements using the swept meshing technique. (Abaqus/CAE generates hexahedral and hexahedral-dominated meshes by sweeping the quadrilateral and quadrilateral-dominated elements generated by the two algorithms from the source side to the target side.) However, if the region to be meshed contains virtual topology, you can use only the advancing front algorithm to generate the swept mesh. For more information, see *What is the difference between the medial axis algorithm and the advancing front algorithm?*, and *Free meshing with quadrilateral and quadrilateral-dominated elements*.

If you select the advancing front algorithm, Abaqus/CAE will use mapped meshing, if it is appropriate, to improve the mesh for some regions. (Mapped meshing is the same as structured meshing but applies only to four-sided regions.) Abaqus/CAE determines whether it is appropriate to replace the advancing front algorithm with mapped meshing on any of the faces that belong to the source side. For more information, see *What is mapped meshing?*, and *When can Abaqus/CAE apply mapped meshing?*. Abaqus/CAE uses mapped meshing to create quadrilateral and quadrilateral-dominated elements on these appropriate boundary faces and then sweeps the elements from the source side to the target side to create the hexahedral and hexahedral-dominated elements.

A three-dimensional region can be meshed using the swept meshing technique if it has the following characteristics:

- Every side that connects the source side to the target side must have only a single face or be comprised of four-sided combined faces that form a regular grid pattern. *Figure 3* provides two examples of acceptable connecting face patterns.

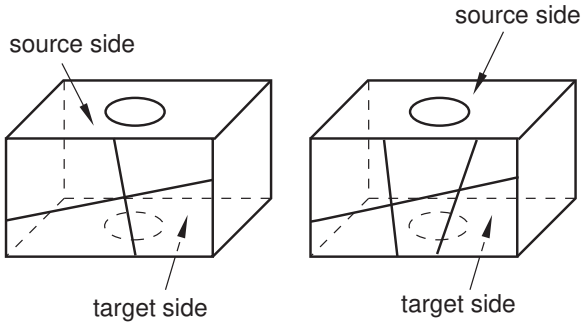


Figure 3: Acceptable connecting face patterns for swept meshing.

The partitioned connecting sides in *Figure 4* do not have acceptable face patterns for swept meshing; the model on the left side has a sweep face with two three-sided faces, while the model on the right side does not have a regular grid pattern.

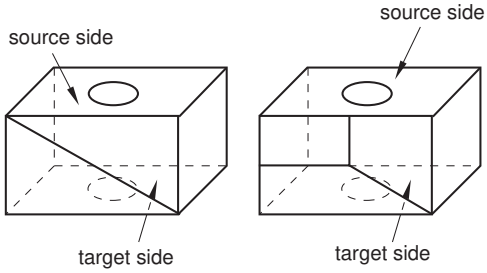


Figure 4: Unacceptable connecting face patterns for swept meshing.

- The target side must contain only a single face without isolated edges or isolated vertices. For example, the region on the left in *Figure 5* can be meshed using the swept meshing technique because all of the isolated edges are on the source side; the region on the right, however, cannot be meshed using this technique because the target side contains two faces.

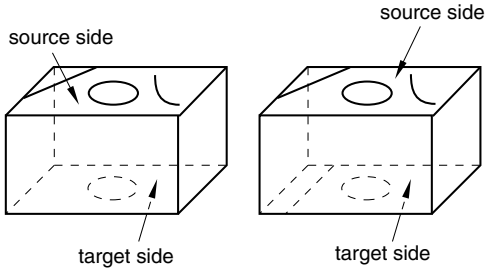


Figure 5: Only the region on the left can be meshed using the swept meshing technique.

Figure 6 illustrates a part that has been swept meshed along a varying cross-section. The part appears to be relatively complex; for example, the source side is nonplanar, and the cross-section of the part varies along the sweep path. However, the rules for generating a swept mesh still apply.

- Every side that connects the source side to the target side contains only a single face.
- Although the source side contains two faces, the target side contains only a single face.

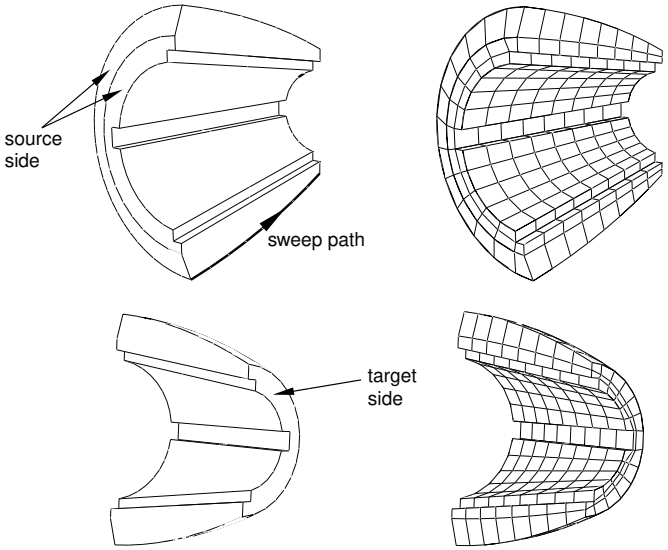


Figure 6: Sweeping the mesh along a varying cross-section.

You may be able to use virtual topology to combine faces on the target side to make a part swept meshable. *Figure 7* illustrates a part that was swept mesh after the five faces on the target side were combined into a single face using virtual topology. However, because the part now contains virtual topology, it can be swept meshed with only the advancing front algorithm.

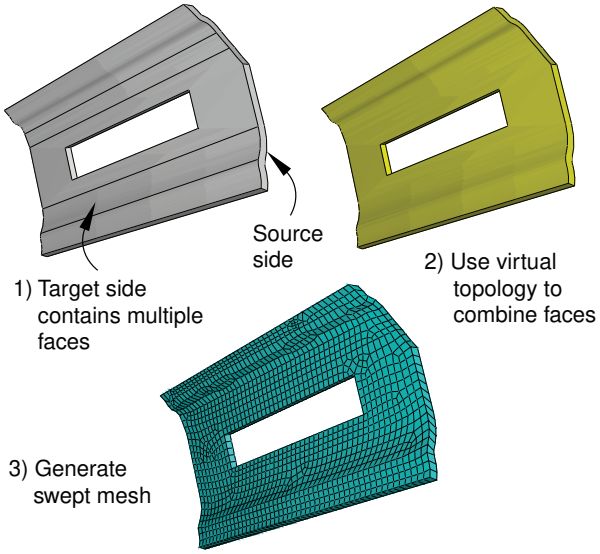


Figure 7: Combining faces makes a part swept meshable.

- For a revolved region, the profile that was revolved to create the region must not touch the axis of revolution at one or more isolated points, as shown in *Figure 8*.

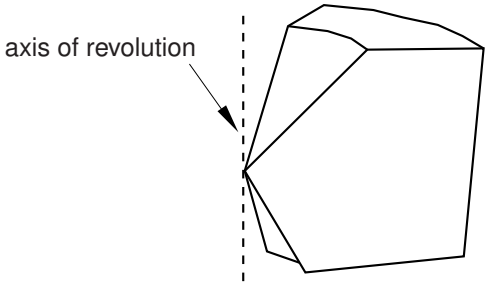


Figure 8: The swept meshing technique cannot mesh a part if an isolated point touches the axis of revolution.

- Similarly, Abaqus/CAE cannot mesh a region with hexahedral or wedge elements if one or more edges lie along the axis of revolution, as shown in *Figure 9*.



Figure 9: Abaqus/CAE cannot mesh a region with hexahedral elements if one or more edges lie along the axis of revolution.

However, Abaqus/CAE can mesh the region with hexahedral-dominated elements by generating layers of wedge elements along the axis, as shown in *Figure 10*.

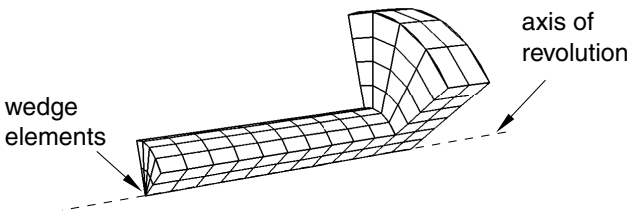


Figure 10: Abaqus/CAE can mesh the region with hexahedral-dominated elements.

As a result, you must select the **Hex-dominated** element shape option before you mesh the region. Alternatively, you can partition the region into simple structural mesh regions and select the **Hex** element shape option to create the mesh using all hexahedral elements. For more information, see *Sweep meshing a solid, revolved region whose profile touches the axis of revolution*.

- A fully revolved region that does not touch the axis of revolution is meshable only if all the edges that are associated with the profile being revolved exist. However, the edges that bound the profile must not create a face. *Figure 11* shows a meshable part instance where all of the edges of the revolved profile exist.

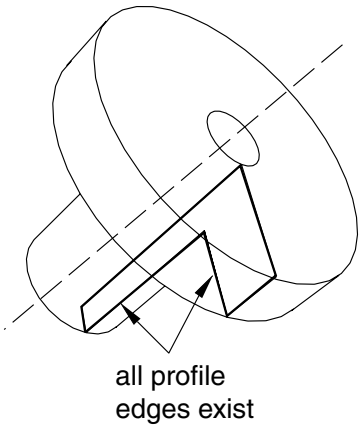


Figure 11: All of the edges of the revolved profile exist; hence, the part instance is meshable.

In this example the user sketched the profile, and Abaqus/CAE revolved the profile to create the part; however, the edges that bound the profile do not form a face. In contrast, *Figure 12* shows a part instance that is not meshable because some of the edges of the revolved profile are missing.

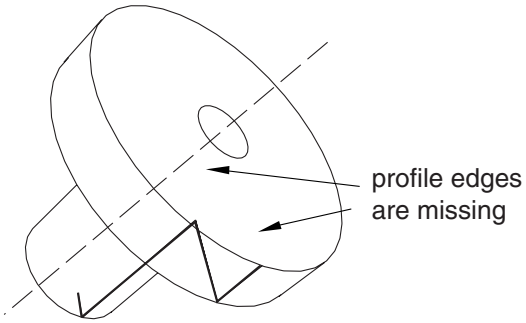


Figure 12: Some of the edges of the revolved profile are missing; hence, the part instance is not meshable.

- A fully revolved region that touches the axis of revolution is meshable only if all of the edges that are associated with the profile being revolved exist except the edges along the axis of revolution. *Figure 13* shows a part instance that is meshable because all of the edges of the revolved profile exist except for the edge along the axis of revolution. If the profile included the edge along the axis of revolution, the part instance would not be meshable.

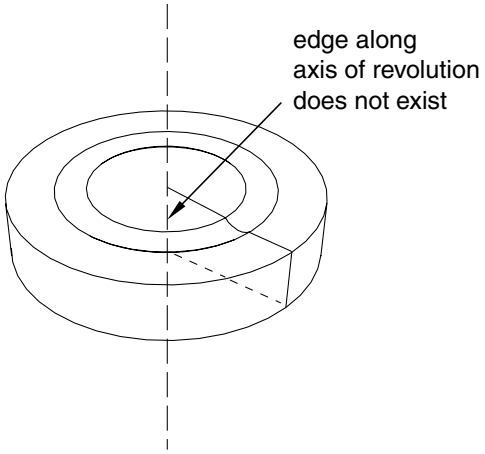


Figure 13: All of the edges of the revolved profile exist, except for the edge along the axis of revolution; hence, the part instance is meshable.

- If a revolved region was created by revolving a sketch that contains a spline, the region is meshable only if the vertices at each end of the spline are not on the axis of revolution.
- If a part was created by sweeping a cross-section along a sweep path that is composed of a closed spline, the resulting part is meshable only if it is split into two or more regions.

Swept meshing of cylindrical solids

You can use swept meshing to create a mesh on cylindrical geometry with many types of solid elements, including elements from the solid cylindrical element library. If you use solid cylindrical elements for a native mesh, Abaqus/CAE uses the sweep path of the selected sweep/revolve region as the circumferential direction for the generated cylindrical elements. You should confirm that the sweep path is set correctly before creating the mesh.

Abaqus considers the following requirements for cylindrical elements when you attempt to mesh solid geometry with cylindrical elements:

- The midside nodes on the circular edges of the cylindrical elements must be placed exactly halfway between the edge nodes.
- All nodes on a cross-section must lie on the same radial plane.
- All the circular edges within an element must span the same angle.
- The centers of the circular arcs within an element must lie on a common axis.
- An element cannot span more than 180°.

These requirements mean that you can create valid cylindrical elements only on revolved parts or parts that you can split into revolved domains. The S-shaped extrusion in *Figure 1* can be meshed with cylindrical elements after partitioning into revolved regions.

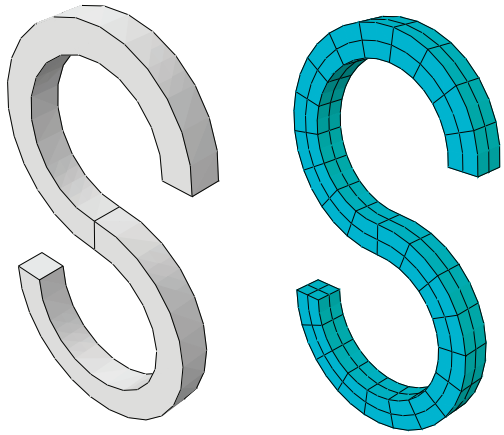


Figure 1: Acceptable geometry for solid cylindrical meshing.

Figure 2 shows three types of geometry—a helical spring, a revolved part with cuts, and a tapered part—that cannot be meshed with solid cylindrical elements because of these meshing requirements.

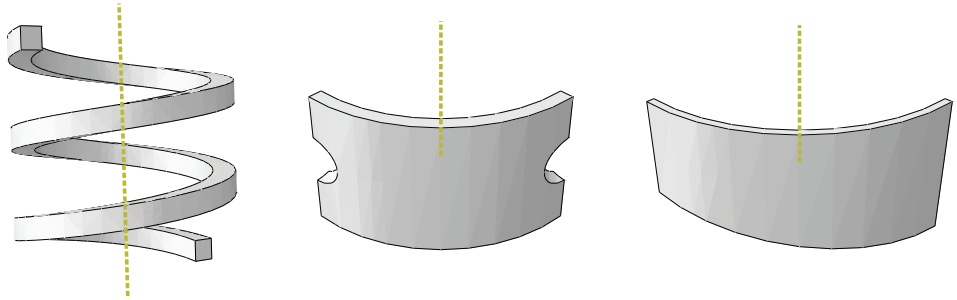


Figure 2: Forms of geometry that cannot be swept meshed with cylindrical elements.

If one of the connecting sides that connect the source face to the target face in a revolved region is separated into multiple faces due to the internal edge or edges that constrain the mesh, the generated elements are likely to be too distorted to be used as cylindrical elements. You must partition the solid region into simpler revolved regions, as shown in the example on the right side of *Figure 3*.

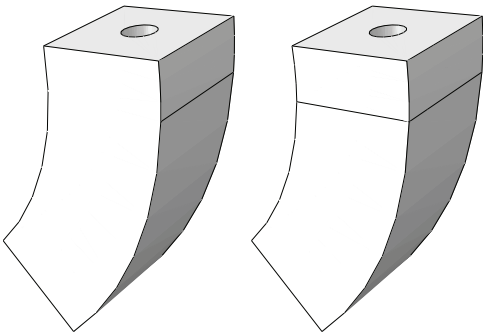


Figure 3: A solid region that is unsuitable for meshing with cylindrical elements (left) and the same region rendered suitable for meshing with cylindrical elements through the use of a partition (right).

When you edit a mesh part by assigning cylindrical elements to it, Abaqus/CAE considers Face 1 and Face 2 of each cylindrical element to be the faces along its radial planes. *Figure 4* indicates the location of these faces for several types of cylindrical elements. You should orient the stack direction for elements in your part before assigning a cylindrical element type to it; for more information, see the descriptions in *Orienting the stack direction*.

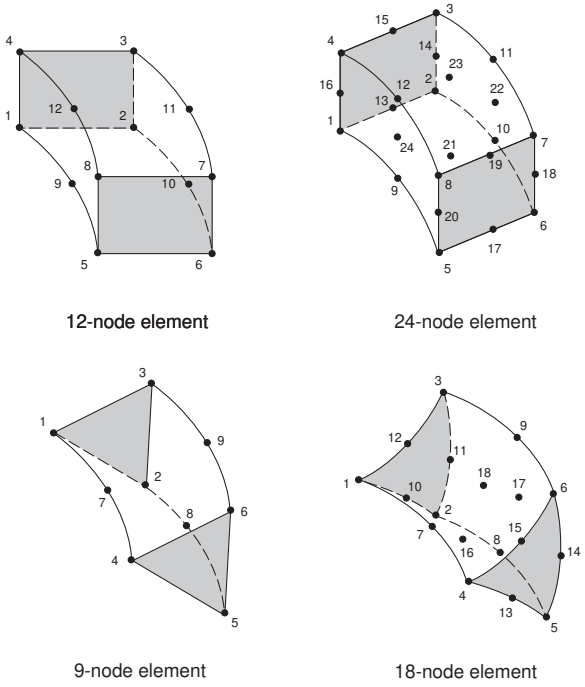


Figure 4: Illustration of node ordering and face numbering for cylindrical elements (with Faces 1 and 2 shaded).

Characteristics of the geometry can prevent a part from being swept meshable

The characteristics of a region described in *Swept meshing of three-dimensional solids* are referred to as "topological characteristics." In some cases, all of the topological characteristics will apply to a region; however, Abaqus/CAE will not allow the region to be swept meshed because some "geometrical characteristics" are not satisfied. The geometric characteristics that Abaqus/CAE looks for when determining if a region is swept meshable are hard to quantify. In general, a three-dimensional region can be meshed using the swept meshing technique if it satisfies the following geometrical characteristics:

- If the source side contains more than one face, the angle between the faces must be relatively flat (close to 180°).
- Each face that connects the source side to the target side (a connecting side) must have four corners. The angle at each of the corners must be close to 90°.
- The angles between the source side and each of the connecting sides should be close to 90°. Similarly, the angles between the target side and each of the connecting sides should be close to 90°.

For example, *Figure 1* shows a part with three faces on the source side. When the angle decreases between the faces that form the source side, the part no longer satisfies the geometric characteristics of a swept meshable region.

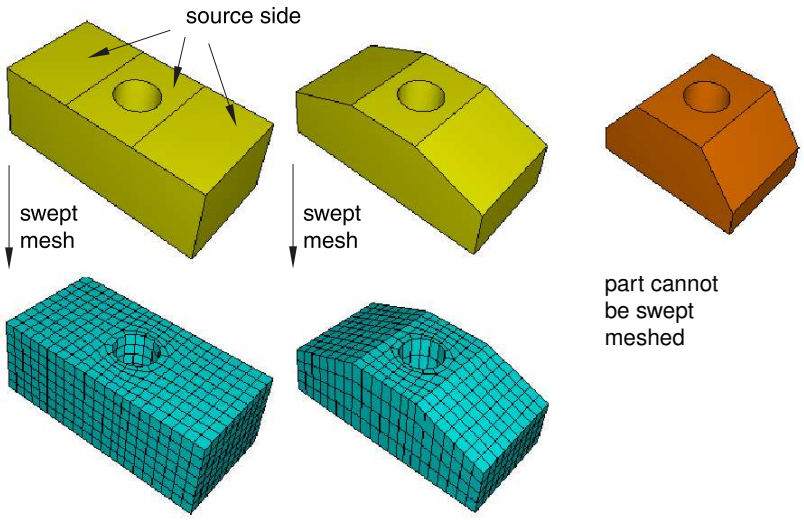


Figure 1: Swept meshing depends on geometrical characteristics.

You may be able to apply virtual topology to satisfy the geometrical characteristics and to make the part swept meshable. For example, *Figure 2* illustrates that the part becomes swept meshable when the three faces on the source side are combined using virtual topology. However, the resulting mesh is of poor quality.

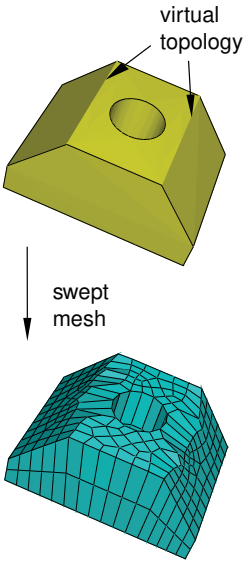


Figure 2: Applying virtual topology can result in poor mesh quality.

In some cases Abaqus/CAE will still allow you to create a swept mesh, even though the geometrical characteristics are not satisfied. The intention is to allow you to create a swept mesh wherever possible. However, the resulting mesh may be of poor quality, or it may be invalid. To ensure that your swept mesh is acceptable, you should use the mesh verify tool to verify its quality. For more information, see *Verifying element quality*.

Free meshing

This section explains how you can use the free meshing technique to mesh two- or three-dimensional models.

In this section:

- *What is free meshing?*
- *Free meshing with quadrilateral and quadrilateral-dominated elements*
- *Free meshing with triangular and tetrahedral elements*
- *What is a tetrahedral boundary mesh?*
- *What can I do with a boundary mesh?*

What is free meshing?

Unlike structured meshing, free meshing uses no preestablished mesh patterns. When you mesh a region using the structured meshing technique, you can predict the pattern of the mesh based on the region topology. In contrast, it is impossible to predict a free mesh pattern before creating the mesh.

Because it is unstructured, free meshing allows more flexibility than structured meshing. The topology of regions that you mesh with the free mesh technique can be very complex.

You can use this technique to mesh a region with the **Tri**, **Quad**, or **Quad-dominated** element shape options for two-dimensional regions or the **Tet** element shape option for three-dimensional regions. For more information on assigning element shapes to a region, see *Choosing an element shape*.

Free meshing with quadrilateral and quadrilateral-dominated elements

Free meshing with quadrilateral elements is the default meshing technique for two-dimensional regions. The free meshing technique with quadrilateral elements can be applied to any planar or curved surface. An example of a mesh generated with this technique is shown in *Figure 1*. Free meshes are usually not symmetric, even if the part or part instance itself is symmetric.

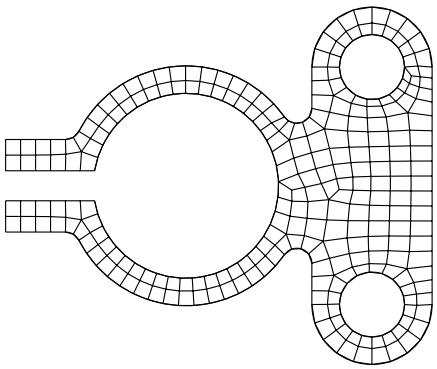


Figure 1: A free mesh generated with quadrilateral elements.

Abaqus/CAE allows you to choose between two meshing algorithms when you create a quadrilateral or quadrilateral-dominated mesh. For more information, see *What is the difference between the medial axis algorithm and the advancing front algorithm?*.

Medial axis

When you free mesh a complex region with quadrilateral elements using the medial axis algorithm, Abaqus/CAE creates internal partitions that divide the region into simple structured mesh regions and then seeds the smaller regions.

If you use the medial axis algorithm to mesh a region and then remesh the region (for example, after modifying the seeds), Abaqus/CAE stores the internal partitions, and the new mesh is generated more quickly. In addition, the internal partitions allow Abaqus/CAE to generate a fine mesh in a similar time to that required to generate a coarse mesh. You cannot use the medial axis algorithm to mesh regions that contain virtual topology, and it does not work well with imprecise parts.

In general, a medial axis-based free mesh with quadrilateral elements does not match the mesh seeds exactly for the following reasons:

- Abaqus/CAE tries to balance the seeds between adjacent regions and the smaller regions created by the internal partitioning.
- Abaqus/CAE tries to minimize element distortions.

However, when you apply fixed seed constraints, Abaqus/CAE matches the number of seeds exactly and attempts to match the seed positions exactly. *Figure 2* illustrates a two-dimensional plate with fixed seeds and movable seeds and the resulting all quadrilateral mesh.

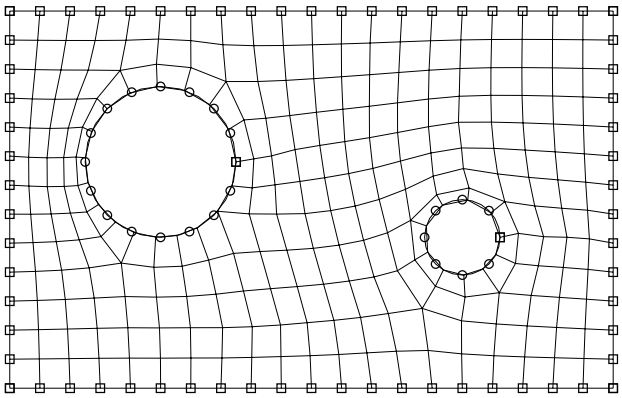


Figure 2: Fixed and movable seeds and the medial axis meshing algorithm.

You should specify fixed seeds on only a few edges, or Abaqus/CAE may not be able to generate a mesh. For example, if you specify fixed seeds around one of the holes in the plate shown in *Figure 2*, the global seeding becomes overconstrained, and Abaqus/CAE cannot generate a mesh.

Using the medial axis algorithm, a free mesh generated with quadrilateral-dominated elements is similar to a free mesh generated with all quadrilateral elements and without transition minimization; however, Abaqus/CAE inserts a few isolated triangles in an effort to match the mesh seeds more closely. Abaqus/CAE can generate a mesh with quadrilateral-dominated elements faster than it can generate a mesh with all quadrilateral elements.

Advancing front

When you free mesh a complex region with quadrilateral elements using the advancing front algorithm, Abaqus/CAE generates quadrilateral elements at the boundary of the region and continues to generate quadrilateral elements as it moves systematically to the interior of the region.

When you choose the advancing front algorithm, Abaqus/CAE matches the seeds exactly (except when you are creating a three-dimensional revolved mesh, and the profile being revolved touches the axis of revolution). A quadrilateral mesh that matches the seeds exactly is shown in *Figure 3*.

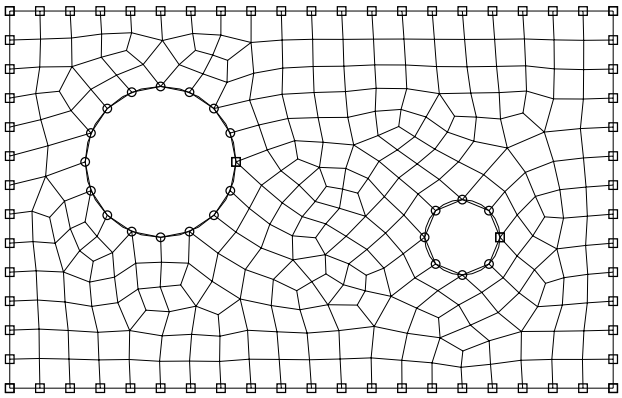


Figure 3: Fixed and movable seeds and the advancing front meshing algorithm.

In general, the mesh transitions generated with the advancing front algorithm are more acceptable than the transitions generated by the medial axis algorithm; however, matching the seeds exactly in narrow regions may compromise the mesh quality. In contrast with the medial axis algorithm, you can use the advancing front algorithm in conjunction with imprecise parts and on regions that contain virtual topology.

If you select the advancing front algorithm, Abaqus/CAE will also use mapped meshing where appropriate. (Mapped meshing is the same as structured meshing but applies only to four-sided regions.) For more information, see *What is mapped meshing?*, and *When can Abaqus/CAE apply mapped meshing?*. When mapped meshing is used, Abaqus/CAE makes minor adjustments to the mesh seeding. If you do not want the seeding to change, you can use mesh controls to prevent the use of mapped meshing. For more information, see *Assigning mesh controls*.

By default, Abaqus/CAE minimizes the mesh transition when it generates a free quadrilateral mesh using the medial axis algorithm. Minimizing the mesh transition results in a better mesh that is generated more quickly; however, the generated nodes deviate further from the mesh seeds. *Figure 4* illustrates the same planar part instance meshed using the medial axis algorithm with and without minimizing the mesh transition and meshed using the advancing front algorithm.

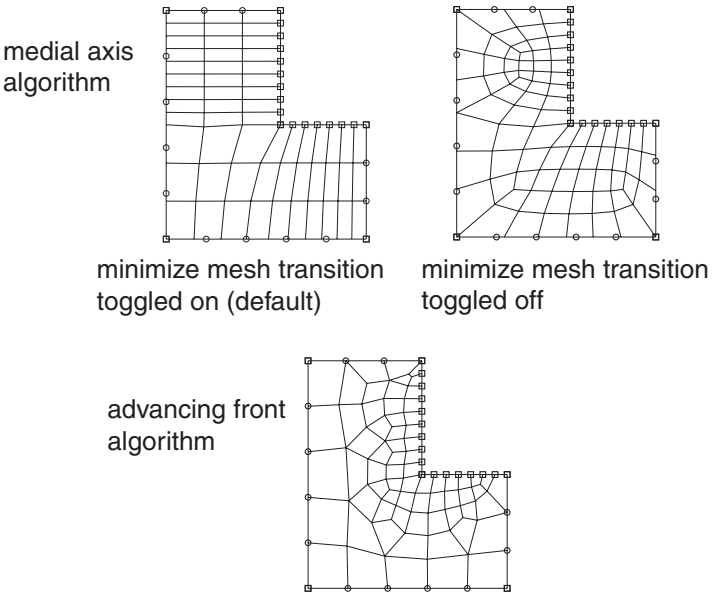


Figure 4: The effect of mesh transition and the meshing algorithm.

Free meshing with triangular and tetrahedral elements

Free meshing with triangular elements can be applied to any planar or curved surface, and the part can be precise or imprecise. For more information, see *What is a valid and precise part?*. A free mesh of triangular elements matches the mesh seeds exactly. This meshing technique can handle large variations in element size, which is useful when you want to refine only part of a mesh. The time taken for Abaqus/CAE to compute a free triangular mesh grows approximately linearly with the number of elements and nodes. *Figure 1* shows an example of a mesh generated using this technique.

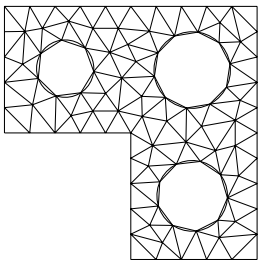


Figure 1: A free mesh generated with triangular elements.

Free meshing with tetrahedral elements can be applied to almost any three-dimensional region; in fact, very complex models can be meshed using this technique without the help of partitioning. Free meshing with tetrahedral elements is similar to free meshing with triangular elements in that the part can be precise or imprecise. In general, Abaqus/CAE matches the mesh seeds exactly although the mesh may be finer around small holes if the mesh seeds are not fully constrained. *Figure 2* shows an example of a free tetrahedral mesh.

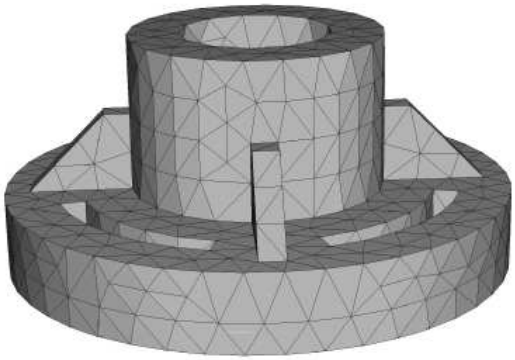


Figure 2: A free mesh generated with tetrahedral elements.

Free meshing of three-dimensional solids using hexahedral elements is not supported.

You should use the Query toolset in the Part module or the Mesh module to check the geometry of the parts or of the assembly before you try to generate a free mesh with tetrahedral elements. You should check the following:

- There are no free edges in the solid.
- There are no short edges, small faces, or small face corner angles.

For more information, see *Using the geometry diagnostic tools*.

When you create a free mesh with tetrahedral elements, you can use the default mesh generation algorithm, or toggle off the default algorithm to use the algorithm that was included with Abaqus/CAE 6.4 and earlier. The default algorithm

is significantly more robust, particularly when meshing complex shapes and thin solids. In addition, the default algorithm allows you to increase the size of the interior elements. If the mesh density is adequate for the model being analyzed and the areas of interest are on the mesh boundary, increasing the size of the interior elements will increase the computational efficiency.

When you are using free meshing to create triangular elements on a surface, Abaqus/CAE will use mapped meshing for some regions if it is appropriate. (Mapped meshing is the same as structured meshing but applies only to four-sided regions.) For more information, see *What is mapped meshing?*, and *When can Abaqus/CAE apply mapped meshing?*. Abaqus/CAE replaces free triangular meshing with mapped triangular meshing only when it is likely that the resulting mesh quality will be improved. If desired, you can use mesh controls to prevent Abaqus/CAE from allowing mapped meshing. For more information, see *Assigning mesh controls*.

Similarly, when you are using free meshing to create tetrahedral elements, you can allow Abaqus/CAE to decide whether mapped meshing is appropriate. After Abaqus/CAE determines that the resulting mesh quality will be improved, it replaces free meshing with mapped meshing on simple boundary faces and converts the resulting triangles into tetrahedral elements.

Figure 3 shows the effect of allowing mapped meshing where appropriate in a free mesh of a solid part using tetrahedral elements. The mesh quality is improved in four-sided regions that can be mapped meshed.

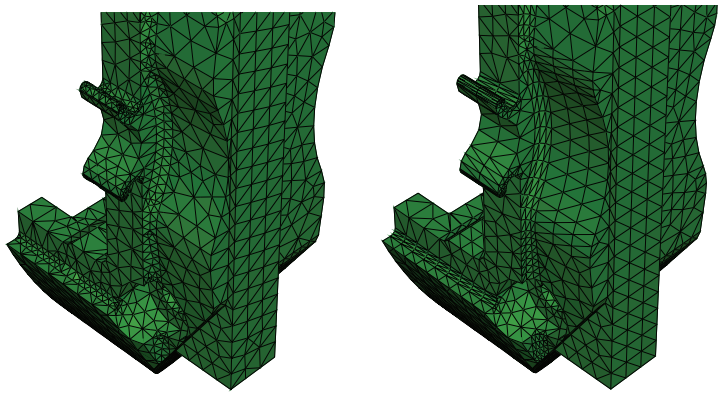


Figure 3: The effect of allowing mapped meshing.

To view the internal tetrahedral elements generated by Abaqus/CAE, you can create a new mesh part from the current mesh and use display groups to remove selected elements. You can also view the internal elements after the analysis is complete by using view cuts in the Visualization module. For more information, see *Using display groups to display subsets of your model* and *Cutting through a model*.

What is a tetrahedral boundary mesh?

When Abaqus/CAE generates a free mesh on a solid with tetrahedral elements, the meshing process consists of two phases:

1. A boundary mesh of triangles is generated on the faces of the solid regions.
2. Abaqus/CAE generates a tetrahedral mesh using the triangles as faces of the tetrahedral elements.

The same process is used to create a quadrilateral boundary mesh and a hexahedral solid mesh for geometric faces of bottom-up mesh regions (for more information, see *Creating the boundary mesh for a bottom-up region*).

If your part is complex, generating a free tetrahedral mesh can be time consuming. Previewing the mesh on the boundary faces can help you identify problems before generating a mesh for the entire part. You can preview the triangles on the boundary faces after the first phase of the meshing process by toggling on **Preview boundary mesh** in the prompt area before you generate the mesh. If the mesh is acceptable, you can continue meshing the interior of the region. If the mesh is not acceptable or if some regions failed to mesh, Abaqus/CAE provides a variety of tools for correcting the problems. For more information, see *What can I do with a boundary mesh?*.

If you decide to preview the triangular boundary mesh for a region, Abaqus/CAE allows you to select faces to mesh even though your final intent is to mesh a solid. When you toggle on **Preview boundary mesh**, Abaqus/CAE automatically changes the selection filter in the prompt area to select **Faces**. For more information, see *Filtering your selection based on the type of object*.

When the **Mesh defaults** color mapping is selected, Abaqus/CAE displays a boundary mesh using white to represent the boundary elements, which is in contrast to the cyan color that Abaqus/CAE uses to represent the final mesh. When you query the tetrahedral boundary mesh, Abaqus/CAE refers to the elements as **Tri boundary elements**. In contrast, when you query the final mesh, Abaqus/CAE refers to the elements as **Linear tetrahedral elements** or **Quadratic tetrahedral elements**. The triangles on the boundary faces have no concept of geometric order.

What can I do with a boundary mesh?

When Abaqus/CAE generates a free mesh on a solid with tetrahedral elements, it first generates a boundary mesh of triangles on the exterior faces of the solid regions, as described in *What is a tetrahedral boundary mesh?*.

When you are creating a free tetrahedral mesh, Abaqus/CAE can create either a free mesh or a mapped mesh on the boundary faces; two mechanisms control the mesh pattern for this triangular surface mesh:

Global mesh control associated with the three-dimensional region

Global controls always exist. They are set when you assign the free tetrahedral meshing technique to the region. You can use them to create a mapped mesh in areas where Abaqus/CAE determines that mapped meshing is appropriate or to create a free mesh on all bounding faces.

Local mesh controls specified on selected faces of the region

To assign local controls, you select one or more faces of the region and specify whether Abaqus/CAE should create a free mesh or a mapped or structured mesh on the selection. If you define local controls, they override the global controls.

If you assign the structured technique to faces of solid regions that will be tetrahedral meshed, these faces are colored green to show the technique assignment for the surface mesh. (For detailed instructions on assigning mesh controls, see *Assigning mesh controls*.)



Note:

The green coloring is applied only while you are assigning mesh controls to the faces. When the procedure ends, the global mesh control colors are displayed.

In addition, Abaqus/CAE highlights any faces on the boundary that failed to mesh. These failures are usually due to mesh seeding that is too coarse or to tiny edges or faces. You can save the highlighted faces in a set, and you can apply finer seeds to only the faces in the set. For more information about seeding faces in a set, see *Can I seed a face or a cell?*. You can use display groups to display only the faces in the set. For more information, see *Plotting display groups*.

You can query a boundary mesh using the Query toolset. In addition, you can check the quality of a boundary mesh using the mesh verify tool. The mesh verify tool allows you to check the quality of all boundary triangles, or you can use the selection filters to check the quality of the boundary triangles of only selected faces.

If tiny edges or faces prevent Abaqus/CAE from generating an acceptable tetrahedral mesh, you can try the following:

- Use the geometry diagnostics tool to find small entities such as short edges, small faces, and faces with small face corner angles that can affect the mesh quality. You can create a set containing these small entities. For more information, see *Using the geometry diagnostic tools*.
- Use the Geometry Edit toolset to remove redundant edges and vertices. You can also remove a face and stitch over the resulting gap. For more information, see *Editing techniques*.
- Use the Virtual Topology toolset to ignore tiny edges or faces. For more information, see *What can I do with the Virtual Topology toolset?*.
- Add partitions to reduce the aspect ratio of long, narrow faces or cells. For more information, see *The Partition toolset*.
- Use the Edit Mesh toolset to modify the preview mesh. You can do the following in the Mesh module:
  - Edit nodes
  - Collapse element edges
  - Swap the diagonal of a pair of adjacent triangular elements

- Split element edges

For more information, see *What can I do with the Edit Mesh toolset?*.

In some cases you will not be able to mesh an imported solid part with tetrahedral elements because of very thin triangular elements in the surface mesh or because some sliver faces cannot be meshed with triangles. *Using a combination of tools to mesh an imported solid part with tetrahedral elements*, describes how you can use the Edit Mesh toolset and other tools in the Mesh module to mesh the part successfully.

Bottom-up meshing

This section explains the bottom-up meshing technique and describes the types of regions to which this meshing technique can be applied. The related topic of mesh associativity is also explained.

In this section:

- *What is bottom-up meshing?*
- *The bottom-up meshing domain*
- *Bottom-up meshing methods*
- *Selecting parameters for a bottom-up mesh*
- *Creating the boundary mesh for a bottom-up region*
- *Improving the quality of boundary meshes for a bottom-up region*
- *Defining connecting sides for a bottom-up swept mesh*
- *Creating a bottom-up mesh*
- *An example including bottom-up meshing techniques*

What is bottom-up meshing?

Bottom-up meshing is a manual, incremental meshing process that allows you to build a hexahedral mesh in any solid region. Structured, swept, and free meshing techniques all work in a top-down manner—they are tied directly to the geometry such that the resulting mesh fills the geometry. Bottom-up meshing relaxes the constraint that ties the mesh to the geometry so that you can build a mesh that ignores some geometric features. You can also use bottom-up meshing techniques to modify a mesh part, in which case there are no geometric features to consider. If you work with geometry, the elements that you create in a bottom-up mesh are always associated with the solid region that you meshed, but the mesh boundaries may not be associated with the geometric boundaries of the region. This allows you to create a mesh using only hexahedral elements where top-down meshing techniques might require extensive partitioning or the use of tetrahedral elements to complete the mesh. However, the relaxed geometry constraints also mean that you must carefully choose the parameters used to create the mesh, since it can vary significantly from the geometry. Once Abaqus has generated a bottom-up mesh, you need to evaluate whether it is suitable for the analysis and, if geometry is present, verify that the mesh is correctly associated with it.

You can apply bottom-up meshing to any solid region, including regions that can be meshed using the top-down techniques, and to meshes that do not have any associated geometry. Since bottom-up meshing is a manual process that can be very time consuming, it is recommended that you use this method only when the top-down methods fail to produce a satisfactory mesh.

Generating the desired mesh may require multiple applications of the bottom-up meshing technique. If multiple applications are required, each bottom-up mesh becomes an incremental building block for the next mesh until you complete the mesh for the region. Each application of the bottom-up meshing technique involves four phases:

1. You select the domain within which Abaqus/CAE will create the mesh. You can choose either a three-dimensional geometry region or orphan elements.
2. You select the method that Abaqus/CAE will use to create the mesh. You can choose from the following methods:
   - Sweep
   - Extrude
   - Revolve
   - Offset (available only for orphan element selections)
3. You select the side, called the source side, that Abaqus/CAE will use to create a two-dimensional mesh to be swept, extruded, or revolved to fill a three-dimensional region.
4. You select the remaining parameters to complete the definition of the bottom-up mesh. For example, if you chose the sweep method, you can choose connecting sides and a target side.

In addition, you may need to edit bottom-up meshes or associate them with geometry before you can use them to continue generating the region mesh. For more information, see *The Edit Mesh toolset*.

Figure 1 shows a part that cannot be meshed with hexahedral elements using top-down techniques.

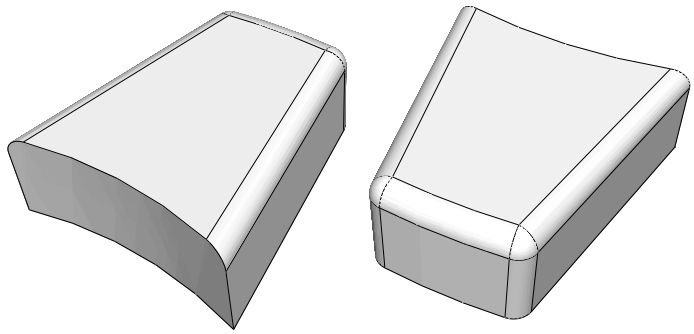


Figure 1: Complex geometry such as fillets can prevent top-down meshing.

The bottom-up sweep method was used to mesh the part in *Figure 2*. The user selected six faces for the source side (magenta), selected the hidden rear face as the target side (white), and used the six remaining sides as connecting sides (yellow), as indicated in the figure. In this example the selected sides are all geometric faces; Abaqus/CAE also allows you to select element faces and two-dimensional elements to define the source side and the connecting sides of a bottom-up mesh.

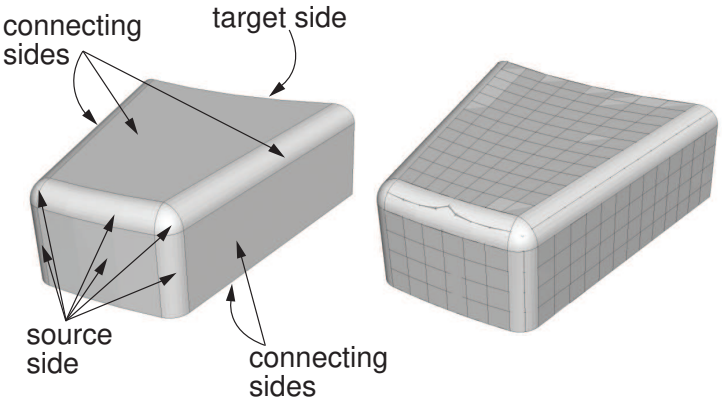


Figure 2: A bottom-up mesh using the sweep method.

You can choose from the sweep, extrude, or revolve methods to mesh a selected geometric region, and you can also choose the offset method if you are working with orphan elements. Depending on the shape of the region and the analysis intent, more than one method may produce acceptable results. The method that you select and the corresponding parameters will determine how well the mesh conforms to the geometry, if applicable. For example, *Figure 3* shows the same part as *Figure 2*; this time the part is meshed with the extrude method. The parameters for the extrude method are a source side, an extrusion vector, and a number of layers. As shown in *Figure 3*, the extruded mesh does not capture the tapered sides and the rounded corners at the back of the model.

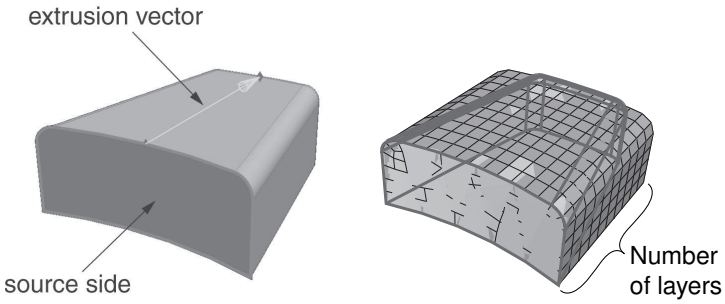


Figure 3: A bottom-up mesh using the extrude method.

If these details are not important, this simplified mesh may be acceptable. Alternatively, you could edit the resulting mesh by modifying or removing the nodes and elements that extend beyond the original geometry. For more information on editing a mesh, see *The Edit Mesh toolset*.

If a resulting bottom-up mesh is not acceptable, you can delete it and try a different bottom-up method for the same region or partition the region and mesh the resulting simpler regions. You can also use the mesh **Undo** and **Redo** functions to undo steps in the bottom-up meshing procedure rather than deleting the entire bottom-up mesh for the region and starting over.

Note:

Undo and redo are not available after some operations, such as deleting a region mesh.

The undo and redo functions are the same as those used for the Edit Mesh toolset (for more information, see *Undoing or redoing a change in the Edit Mesh toolset*).

The bottom-up meshing domain

When you begin creating a bottom-up mesh, the first requirement is to define the domain in which the mesh will be created. The domain determines whether the bottom-up mesh is created as a native mesh or as a collection of orphan nodes and elements. To create a native mesh, you choose a three-dimensional geometric model region that has been assigned the bottom-up meshing technique in the **Mesh Controls** dialog box. (For more information, see *Assigning mesh controls*.) If there are no geometric regions with the bottom-up technique assigned, you must create one or use the orphan element domain.

Selecting a three-dimensional region as the domain also indicates that you intend to fully associate the bottom-up mesh with geometry. The association between mesh and geometry is necessary to transfer loads, boundary conditions, and other information from the geometry to the mesh. Mesh-geometry association is discussed in *Mesh-geometry association*.

Selecting orphan elements as the domain indicates that the bottom-up mesh will not be fully associated with geometry. The created elements may either have no association with geometry, or you may associate the created orphan mesh entities with adjacent geometric faces. Associating the orphan mesh faces with adjacent geometric faces allows Abaqus/CAE to create a compatible mesh when you mesh the geometry.

Bottom-up meshing methods

The control parameters for each bottom-up method are similar to the parameters used internally by Abaqus/CAE to create a comparable top-down swept mesh for a three-dimensional solid. The bottom-up methods and associated parameters are defined as follows:

Sweep

The sweep method creates a three-dimensional mesh by moving a two-dimensional mesh along a sweep path. The sweep method is illustrated in *Figure 2*. You should use the bottom-up sweep meshing method when the region cross-section changes between the starting and ending sides. To use the sweep method, you must first choose a **Source side** that defines the face or faces on which Abaqus/CAE will create a two-dimensional mesh. The source side can be any combination of geometric faces, element faces, and two-dimensional elements. You can define the sweep path by selecting **Connecting sides** that define the sides of the desired sweep region. If you define connecting sides, the mesh conforms closely to the geometry or mesh along the selected sides. Alternatively, for geometry you can select a **Target side** and specify a **Number of layers** and allow Abaqus/CAE to create the sweep path by interpolating between the source and target sides. The **Target side** is a single face that Abaqus/CAE uses to end the mesh. The number of layers refers to the number of element layers that will be placed between the source and the target sides—if you use connecting sides, the two-dimensional meshes of the connecting sides define the number of element layers. Abaqus/CAE sweeps the two-dimensional mesh from the source side into the volume of the solid region to create the mesh.

Extrude

The extrude method is a special case of the sweep method with a linear path defined by a direction and a distance. The extrude method is illustrated in *Figure 3*. You should use the extrude method for regions with a constant cross-section and a linear sweep path. There are three required parameters for a bottom-up extruded mesh. As with the sweep method, you choose the **Source** side that defines the area on which Abaqus/CAE will create a two-dimensional mesh. You then select the starting and ending point of a **Vector** that defines the extrusion direction and can also be used to define the extrusion distance. Finally, you indicate the **Number of layers** to define the number of elements between the source side and the end of the extruded mesh. If you use the vector to define the extrusion distance, the definition is complete. However, you can **Specify** a distance or use **Project to target** and select a target side to define the extrusion distance. The target side can be selected from any geometry, mesh, or datum plane in the viewport; it need not be part of the same part instance as the source. Abaqus/CAE extrudes the two-dimensional mesh from the source side in the direction of the extrusion vector. If you select a target side to define the extrusion distance, Abaqus/CAE ends the extruded mesh at the target side. *Figure 1* shows the source side and target side on the left; the extrusion vector (not shown) extends from the center of the rectangular source side to the center of the cylinder. The resulting extruded mesh is an extension of the source side mesh. It closely matches the target side shape, but no attempt is made to match the node positions of the mesh on the target side.

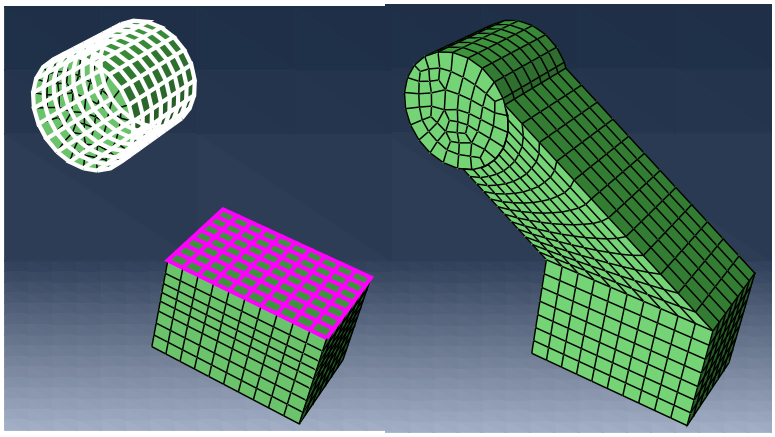


Figure 1: The optional target side (colored white) is used to define the extrusion distance.

The **Bias ratio** parameter defines a change in the element thickness between the source side and the end of an extruded bottom-up mesh in which more than one layer is created. The bias ratio is the ratio of the thickness of the first layer of elements in the extruded mesh to the thickness of the last layer of elements. The default bias ratio of `1.0` has equal thickness elements throughout the extrusion distance.

Revolve

The revolve method is another special case of the sweep method. In this case the sweep path is a circular path defined by an axis and an angle of revolution. The revolve method is illustrated in *Figure 2*. You should use this method for regions with a constant cross-section and a circular sweep path. As with the sweep and extrude methods, you choose the **Source side** that defines the area on which Abaqus/CAE will create a two-dimensional mesh. The source side cannot intersect the axis of revolution, but it may contain edges that coincide with the axis. If so, the elements contacting the axis create a layer of wedge elements in the revolved mesh. The source side should not include any triangular elements along the axis of revolution. You then select the starting and ending point of an **Axis** that defines the axis of revolution. Finally, you indicate the **Angle** and the **Number of layers** to define the number of elements between the source side and the end of the revolved mesh. Abaqus/CAE revolves the two-dimensional mesh from the source side by the specified angle and evenly divides the resulting region into the desired number of element layers. The direction of revolution is clockwise if you look along the axis of revolution from the starting point to the ending point.

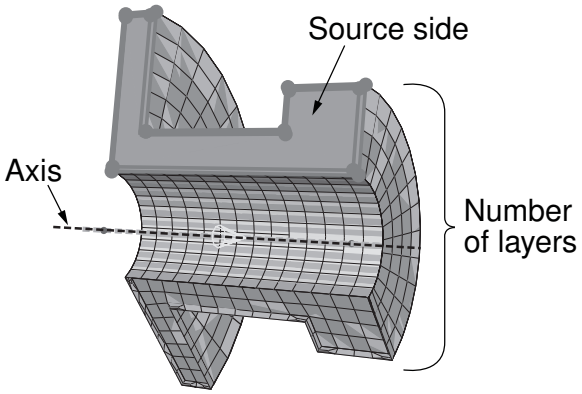


Figure 2: The bottom-up revolve method sweeps the source side mesh by the specified angle about the axis.

Offset

The offset method works the same as **Offset (create solid layers)** in the Edit Mesh toolset; it creates one or more layers of solid elements by offsetting the selected elements. Offset is available only when you are working with orphan elements. To create an offset bottom-up mesh, you enter a total thickness for the offset elements and the desired number of element layers. You can create an element set or extend an existing set, but you cannot create top and bottom surfaces as you can with the Edit Mesh toolset. If you select shell elements as the source, you must indicate in the prompt area the desired offset direction; you can offset shell elements in both directions. If your shell element selection contains sharp corners, toggle on **Constant thickness around corners** to maintain the same total thickness where the elements meet in the corner as in the rest of the selection. Using this option can reduce element distortion and prevent collapsed elements, especially if elements are offset to the inner side of a corner (for more information, see *Reducing element distortion and collapse during mesh offsetting*).

Selecting parameters for a bottom-up mesh

The parameters that you use to create a bottom-up mesh are similar to those that Abaqus/CAE uses to create a comparable top-down swept mesh. In the top-down meshing methods, Abaqus/CAE automatically selects the required geometric sides for the region. Top-down swept meshes—including extruded and revolved meshes—are discussed in *Swept meshing of three-dimensional solids*. The parameters used in the bottom-up mesh methods are described below.

Source side

The **Source side** is a required parameter that defines the face or faces on which Abaqus/CAE will create a two-dimensional mesh that will be swept, extruded, or revolved to create a three-dimensional mesh. The source side may be a combination of geometric faces, element faces, and two-dimensional elements; you can also select saved surfaces instead of selecting from the viewport. For a top-down mesh Abaqus/CAE limits the angularity between multiple faces that it selects as the source side (for more information, see *Characteristics of the geometry can prevent a part from being swept meshable*). There is no limit on the angle between the faces that you can select as the source side of a bottom-up mesh. However, increased angles between the source side faces or between the sourcewill result in decreased quality in the three-dimensional elements.

Connecting sides

Connecting sides define the direction of a swept region. Connecting sides may be a combination of geometric faces, element faces, and two-dimensional elements. Every connecting side must have only a single four-sided geometric face or be comprised of four-sided combined geometric faces, element faces, and two-dimensional elements that form a regular grid pattern.

In a top-down swept mesh Abaqus/CAE creates connecting sides along all edges of the source side. The connecting sides extend completely from the source side to the target side. In a bottom-up swept mesh connecting sides are optional for geometry, but it is recommended that you include as many connecting sides as possible. Connecting sides help to control the mesh as it is swept through the three-dimensional region and enforce *associativity* in the resulting mesh. Including connecting sides reduces the amount of work needed to clean up and associate the bottom-up swept mesh with the geometry. Connecting sides for a bottom-up mesh may extend completely from the source side to the target side or they may cover only part of the distance. They may also be used without a target side, in which case the end of the connecting sides defines the end of the bottom-up swept mesh. For more information, see *Defining connecting sides for a bottom-up swept mesh*.

Target side

The **Target side** defines the end of a swept mesh or an extruded mesh. You cannot select a target side for revolved or offset bottom-up meshes. The target side of a swept mesh must be a single geometric face; the target side for an extruded mesh can be one or more geometric faces, a group of element faces, or a datum plane. The target side is not required for a swept mesh unless you have not included any connecting sides. However, including a target side helps you control the mesh and, if applicable, enforce *associativity* of the mesh to geometry. Abaqus/CAE creates a mesh for the target side by projecting the nodes from the source side if connecting sides are not used or from the last layer of nodes if connecting sides are used. The projected nodes may be positioned outside of the selected geometric target side. The target side is optional for an extruded mesh and is used only to establish the extrusion distance; selection of mesh faces for a target side will not change the extruded mesh to conform to the existing nodes and element faces.

Number of layers

The **Number of layers** parameter defines the number of layers of elements that Abaqus/CAE generates along the sweep, extrude, or revolve direction. If you select the bottom-up sweep method and select connecting sides, the number of element layers is driven by the mesh or seeding of the connecting sides, similar to the top-down

meshing techniques. If you do not select connecting sides for the sweep method or if you choose the extrude or revolve methods, you must specify how many element layers Abaqus/CAE should generate.

Vector

The **Vector** parameter defines the extrusion direction and, optionally, the distance for an extruded mesh. You select an extrusion vector by picking a start point and an end point from the nodes, vertices, datum points, and interesting points in the viewport. Abaqus/CAE extrudes the mesh from the source side by the direction and, if applicable, distance of the vector regardless of where in the viewport you define the vector.

Axis

The **Axis** parameter defines the axis of revolution for a revolved mesh. You can define the axis of revolution by selecting two vertices, nodes, datum points, or interesting points from the viewport.

Angle

The **Angle** parameter defines the angle of revolution for a revolved mesh. You must specify the angle of revolution for the bottom-up revolve technique to indicate where Abaqus/CAE should end the revolved mesh.

Bias ratio

The **Bias ratio** parameter defines a change in the element thickness between the source side and the end of an extruded bottom-up mesh in which more than one layer is created. The bias ratio is the ratio of the thickness of the first layer of elements in the extruded mesh to the thickness of the last layer of elements. A bias ratio less than one creates thinner layers near the source side, a ratio of exactly one has equal thickness layers throughout, and a ratio greater than one creates thicker layers near the source side. Abaqus/CAE evenly distributes the bias through the number of layers in the extrusion.

As you select the parameters to define a bottom-up mesh, consider not only the shape of the current bottom-up mesh but also the desired final mesh shape. Many complex parts will require several bottom-up meshing iterations to generate a complete mesh. For example, you may use connecting sides or element faces of a bottom-up swept mesh as the source side of a bottom-up extruded mesh. If you cannot completely mesh the selected region in the current iteration, consider how you can add another bottom-up mesh or use the Edit Mesh toolset to complete the mesh. For a detailed, step-by-step example of combining top-down and bottom-up techniques to mesh a part, see *An example including bottom-up meshing techniques*.

Creating the boundary mesh for a bottom-up region

When you create a bottom-up mesh for a geometric region, the meshing process consists of two phases:

1. A boundary mesh of quadrilateral elements is generated on the source side—and on the connecting sides of a swept mesh if they are included.
2. Abaqus/CAE generates a hexahedral mesh using the quadrilateral elements as faces of the hexahedral elements.

Creating a boundary mesh allows you to preview the first phase of producing the bottom-up mesh. Viewing the boundary elements can help you identify problems that may prevent generating a bottom-up mesh. The boundary mesh functions for geometric faces of bottom-up mesh regions are identical to the tetrahedral boundary mesh capabilities in Abaqus/CAE except that a bottom-up boundary mesh is composed of quadrilateral elements instead of tetrahedral elements (for more information on tetrahedral boundary meshes, see *What is a tetrahedral boundary mesh?*, and *What can I do with a boundary mesh?*).

To create the boundary mesh for faces of a bottom-up region you must use the top-down region meshing process.

1. From the main menu bar, select **Mesh->Region**.
2. Toggle on **Preview boundary mesh** in the prompt area.

   Abaqus/CAE automatically switches the default selection option from **Regions** to **Faces**.
3. Select the geometric faces of a bottom-up region for which you want to create the boundary mesh.
4. Click **Done** in the prompt area.

   Abaqus/CAE creates a two-dimensional mesh on the selected faces.

When the **Mesh defaults** color mapping is selected, Abaqus/CAE displays a boundary mesh using white to represent the quadrilateral elements, which is in contrast to the cyan color that Abaqus/CAE uses to represent the final solid mesh. When you query the bottom-up boundary mesh, Abaqus/CAE refers to the two-dimensional boundary elements as **Quad boundary elements**. In contrast, when you query the final mesh, Abaqus/CAE refers to the three-dimensional solid elements as **Linear hexahedral elements**.

Improving the quality of boundary meshes for a bottom-up region

The quality of the boundary mesh on the source and connecting sides is a major factor affecting the quality of a swept mesh. Creating the boundary meshes allows you to improve their quality before you use them to create a bottom-up mesh. For example, to use a combination of unassociated geometric and unassociated element faces to create an acceptable source side, you may need to create the boundary mesh and merge the nodes of the two-dimensional mesh. Likewise, you cannot use a combination of geometric and element faces to create a connecting side, but you can create a single boundary mesh and use a combination of its two-dimensional elements and the element faces of an adjacent three-dimensional mesh to create a connecting side.

You can query a boundary mesh using the Query toolset. In addition, you can check the quality of a boundary mesh using the mesh verify tool. The mesh verify tool allows you to check the quality of all the boundary elements, or you can use the selection filters to check the quality of the boundary elements of only selected faces.

To improve the boundary meshes for a bottom-up region, you can try the following:

- Modify the mesh controls for the faces of a bottom-up region. For more information, see *Assigning mesh controls*.
- Use the geometry diagnostics tool to find small entities such as short edges, small faces, and faces with small face corner angles that can affect the mesh quality. You can create a set containing these small entities. For more information, see *Using the geometry diagnostic tools*.
- Use the Geometry Edit toolset to remove redundant edges and vertices. You can also remove a face and stitch over the resulting gap. For more information, see *Editing techniques*.
- Use the Virtual Topology toolset to ignore tiny edges or faces. For more information, see *What can I do with the Virtual Topology toolset?*.
- Add partitions to reduce the aspect ratio of long, narrow faces or cells. For more information, see *The Partition toolset*.
- Use the Edit Mesh toolset to modify the boundary mesh. You can do the following in the Mesh module:
  - Edit nodes
  - Collapse element edges
  - Swap the diagonal of a pair of adjacent triangular elements
  - Split element edges

  For more information, see *What can I do with the Edit Mesh toolset?*.

Defining connecting sides for a bottom-up swept mesh

Defining the connecting sides for a bottom-up swept mesh is a complex task. Connecting sides are required only when creating a swept mesh that uses orphan element faces as the source side. However, even when the connecting sides are not required, using them can save you a significant amount of time after you create the mesh. Connecting sides help control the shape of the swept mesh, and they enforce associativity between the mesh and geometry or force the mesh to match the existing mesh along the connecting sides, depending on whether you are working with geometry or a mesh. In other words, connecting sides force a bottom-up swept mesh to be more similar to a top-down swept mesh. Without connecting sides, you may need to edit the mesh shape, merge nodes, or otherwise modify the mesh to comply with adjacent geometry or meshes, and you will need to manually associate the mesh with the surrounding sides if there are any loads or boundary conditions applied to those sides.

Connecting sides for a bottom-up mesh may be a combination of geometric faces, element faces, and two-dimensional elements. Elements or element faces must be quadrilaterals, and a single connecting side cannot contain both mesh entities and geometric faces. The criteria below and the examples that follow should help you select valid connecting sides or create valid connecting sides from sides that are initially invalid:

- Each geometric face must have four logical sides.
- The corners of each geometric face must be consistent with the mesh sweeping operation.
- Connecting sides must share a common edge with the source side.
- Geometric faces or mesh entities must form a regular grid pattern.

Each geometric face must have four logical sides

In *Figure 1* the top face contains a hole and the front face includes a triangular face. Neither face can be used as a connecting side because of these additional features.

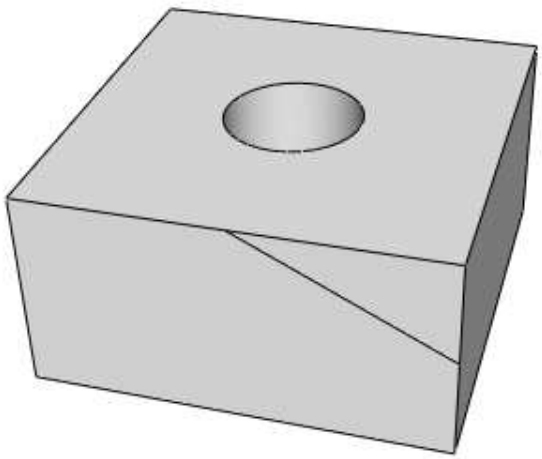


Figure 1: Connecting sides must have four logical sides.

If you select sides such as these, Abaqus/CAE will display an error message indicating that the faces are topologically not four-sided.

In addition to topological tests, Abaqus applies geometric tests to determine whether faces that have four or more edges should be considered four-sided. The geometric tests evaluate the angles at the vertices of the face and determine whether a good quality structured mesh can be generated. You can override the geometric tests by using the mesh controls to assign the structured meshing technique to these faces.

For example, in *Figure 2* the front face is topologically four-sided; however, because the sides meet the top edge along a curve, Abaqus/CAE does not recognize all four of the corners.



Figure 2: Faces with corner radii may require editing for use as connecting sides.

When you finish selecting connecting sides, Abaqus/CAE highlights faces that are not geometrically four-sided and displays an error message. You can use the mesh controls to redefine the region's corners as highlighted in the figure so that the face can be used as a connecting side. For more information on using mesh controls on the faces of bottom-up mesh regions, see *Improving the quality of boundary meshes for a bottom-up region*.

The corners of each geometric face must be consistent with the mesh sweeping operation

The four corners shown on the selected face in *Figure 3* do not match the corners of the adjacent source side (the bottom side of the part). In this case you can finish selecting the connecting sides; but when you attempt to apply the bottom-up swept mesh, Abaqus/CAE indicates that it cannot map the mesh on the connecting sides into a regular grid. If you use mesh controls to remove the corner at the semicircular cut and add a corner at the bottom left corner of the side, you can use the face as a connecting side.

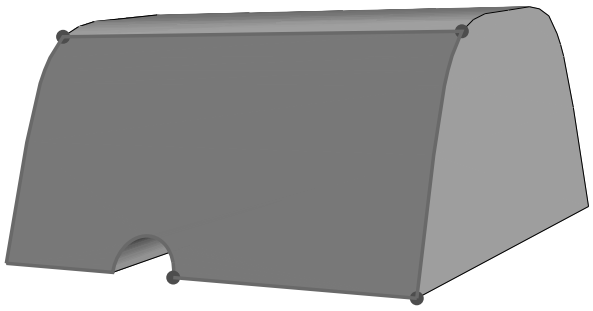


Figure 3: Four-sided faces whose corners do not match the corners of the source side.

Connecting sides must share a common edge with the source side

The common edge shared between the source side and connecting sides cannot extend beyond the edges of the source side. In addition, if you use two-dimensional elements or element faces as the source side and select geometric faces as a connecting side, or vice versa, the element edges must be associated with the geometric edge or Abaqus will not recognize them as sharing a common edge. As an alternative to associating the elements with the geometric edge, you can create a bottom-up boundary mesh on the geometric face and manually merge the nodes of the source side and connecting side meshes (for more information, see *Creating the boundary mesh for a bottom-up region*). If a connecting side does not share a common edge with the source side, Abaqus/CAE indicates that it cannot map the mesh on the connecting sides into a regular grid.

Geometric faces or mesh entities must form a regular grid pattern

The geometric faces or element faces and two-dimensional elements that you select must form a regular grid pattern. *Figure 4* provides two examples of acceptable connecting face patterns.

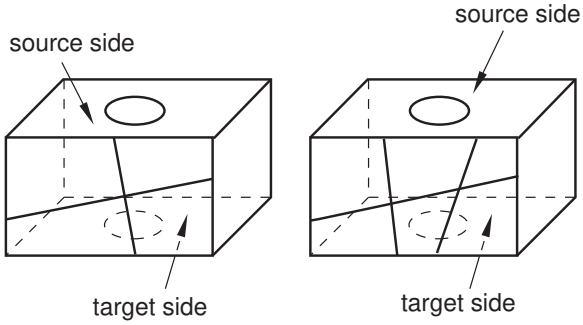


Figure 4: Acceptable connecting face patterns for swept meshing.

The combination of the three four-sided faces in *Figure 5* makes it unacceptable as a connecting side. In this case when you indicate that you are done selecting connecting sides, Abaqus/CAE displays an error message stating that the selected geometric faces do not form a grid pattern.

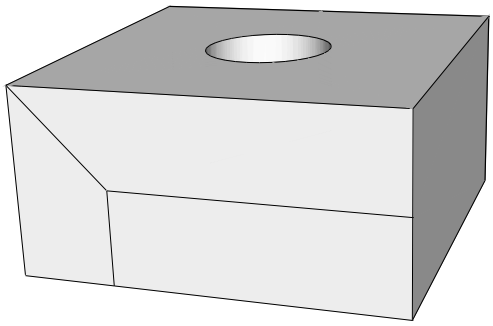


Figure 5: An unacceptable face pattern.

If you are unable to create acceptable connecting sides, you can omit the connecting sides. The resulting mesh may still be acceptable for analysis. However, you should remember to associate the elements along the sides of the region with the geometry, especially if there are loads or other analysis attributes applied to the sides of the mesh. For more information, see *Mesh-geometry association*.

Creating a bottom-up mesh

Bottom-up meshing is a manual process that may be time consuming and require some trial and error to produce an acceptable mesh. Bottom-up meshing is intended for use primarily when you need hexahedral elements for an analysis and you cannot generate them using top-down meshing techniques. You can also use bottom-up meshing in cases where extensive partitioning would be required to create a high-quality top-down mesh on geometry or where you want to extend a mesh part. The following procedure includes the basic steps for creating a bottom-up mesh. To create a bottom-up mesh, you must first have either a mesh region or a solid region with the bottom-up meshing technique assigned (for more information, see *Assigning mesh controls*). You can also create a bottom-up mesh on an assembly containing independent instances of native parts or dependent instances of parts containing any combination of orphan mesh and geometry.

1. From the main menu bar, select **Mesh->Create Bottom-Up Mesh**.

   Tip: You can also click the  tool in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

   Abaqus/CAE opens the **Create Bottom-Up Mesh** dialog box and displays prompts in the prompt area to guide you through the procedure.

2. Select the **Domain** for which you want to create the bottom-up mesh.

   If there is only one appropriate domain for the current part or assembly, Abaqus/CAE automatically makes the selection. If there are multiple geometry regions, or cells, with the bottom-up technique assigned, click **Select** to select the cell that will be associated with the bottom-up mesh that you are creating. Regions that you can select are colored light tan.

   If you work with orphan elements, there is no association between the elements and any geometric region in the part.

   Note:

   After selecting a geometric region for bottom-up meshing, you can use the **Edit** button to change your selection.

3. Select one of the following meshing methods:

   - **Sweep**
   - **Extrude**
   - **Revolve**
   - **Offset** (available only for element face selections)

   For a description of each method, see *Bottom-up meshing methods*.

4. If you previously selected bottom-up meshing parameters for the current region, you can click **Fetch Last Selections** to reuse your previous selections.

   Abaqus/CAE saves your selections for the domain, source side, connecting sides, target side, extrusion vector, and axis of revolution in the session until they are overwritten by new selections.

   Note:

   If you add a partition to the model, you cannot reuse previously selected geometry; however, as long as you have not modified the mesh, you can reuse mesh selections such as element faces and two-dimensional elements.

5. Click **Select** next to **Source side**; and select the region faces, element faces, or two-dimensional elements where you want the mesh to begin.

   Tip: Use **[Shift] + Click** to select more than one face.

   If you are creating an offset mesh and you selected two-dimensional elements, Abaqus/CAE prompts you to indicate the side of your selection that will be offset. You can choose the **Brown** side, the **Purple** side, or **Both sides**.

   Abaqus/CAE colors your selection magenta.

6. When you have finished selecting the source side, click **Done** in the prompt area.
7. If you chose the **Sweep** method, you must select **Connecting sides** or a **Target side** or both to complete the mesh definition. Complete the following steps to make your selections:
   a. Toggle on **Connecting sides**, and click **Select** to choose one or more sides that connect the source side to the end of the mesh. Connecting geometric sides must be four-sided. You can also select four-sided element faces or two-dimensional elements that form a regular grid pattern. For more information, see *Defining connecting sides for a bottom-up swept mesh*.
   b. If you are working with geometry, you can toggle on **Target side** and click **Select** to choose a single face that Abaqus/CAE uses to end the swept mesh.
8. If you chose the **Extrude** method, you must specify an extrusion vector and choose the extrusion depth.
   a. Click **Select** next to **Vector** to select two points from the viewport or enter coordinates in the prompt area to define the start and end points of the extrusion vector.

      Abaqus/CAE displays a yellow arrow in the viewport to indicate the extrusion vector.
   b. Select a method to define the extrusion depth:

      - Toggle on **Use vector length** to extrude by the length of the vector created in the previous step.
      - Toggle on **Specify** and enter an extrusion depth.
      - Toggle on **Project to target** and click **Select** to specify a target side. The target side must lie along the direction prescribed by the extrusion vector.

        Abaqus/CAE uses the target side to set the extrusion distance. The extruded mesh will closely match the shape of the target side, but it will not conform to a mesh on the target side. (For more information, see *Bottom-up meshing methods*.)

9. If you chose the **Revolve** method, you must specify an axis and an angle of revolution.
   a. Click **Select** next to **Axis** to select two points from the viewport or enter coordinates in the prompt area to define the start and end points of the axis of revolution.

      Abaqus/CAE displays a yellow arrow in the viewport to indicate the axis.
   b. In the **Angle** field, enter the desired angle of revolution, in degrees, for the revolved mesh.
10. If you chose the **Offset** method, you must specify the total thickness, and you can change the way two-dimensional mesh corners are offset.
    a. In the **Total thickness** field, enter the thickness for the offset mesh.
    b. For source sides containing shell elements that form sharp corners, toggle on **Constant thickness around corners** to maintain the same total thickness where the elements meet in the corner as in the rest of the selection. Using the option can reduce element distortion and prevent collapsed elements, especially if elements are offset to the inner side of a corner (for more information, see *Reducing element distortion and collapse during mesh offsetting*).
11. Where applicable, enter the **Number of layers** to define the number of element layers between the source side and the end of the bottom-up mesh.



Note:

The **Number of layers** parameter is required for the **Sweep** method only if you specify a target side with no connecting sides. It is always required for the **Extrude**, **Revolve**, and **Offset** methods.

12. If desired, enter a **Bias ratio** for an extruded bottom-up mesh to skew the thickness of element layers toward or away from the source side. **Bias ratio** is used in conjunction with **Number of layers**.

    The bias ratio is the ratio of the thickness of the element layer at the source side to the thickness of the element layer furthest from the source side.

13. If desired, use the **Options** area to add the new bottom-up elements to element sets:

Extend existing sets

Each new bottom-up element will be added to any mesh sets containing its parent element from the source side.

Create a set for new elements

Create a new element set for the bottom-up elements. You can use the default set name or create a name for the set. If you are working with orphan elements, you can also create a separate set for each layer of elements. If you choose this option, Abaqus/CAE increments the base set name with `-Layer-n`, where *n* is the layer number.

14. The **Undo** field near the bottom of the **Create Bottom-Up Mesh** dialog box allows you to undo and redo bottom-up mesh changes, mesh-geometry associativity changes, and mesh edits. You can access the same functions in the Edit Mesh toolset. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.
15. Click **Mesh** in the **Create Bottom-Up Mesh** dialog box to generate the bottom-up mesh.
16. Repeat Steps 3 through 15 to create another bottom-up mesh in the same region, or click **Cancel** to end the procedure.

Additional information

- *Selecting objects within the viewport*
- *Bottom-up meshing*
- *Selecting parameters for a bottom-up mesh*
- *Using the Mesh module toolbox*

An example including bottom-up meshing techniques

The following is a step-by-step example of meshing a part using a combination of top-down and bottom-up methods.
The procedures and images describe one way that you can create a hexahedral mesh for this part. Before you begin, you should be familiar with top-down meshing, creating partitions, and using the Edit Mesh toolset. You should also have read the sections discussing bottom-up meshing, mesh-geometry associativity, and related techniques:

- *What is bottom-up meshing?*
- *Bottom-up meshing methods*
- *Selecting parameters for a bottom-up mesh*
- *Creating the boundary mesh for a bottom-up region*
- *Improving the quality of boundary meshes for a bottom-up region*
- *Defining connecting sides for a bottom-up swept mesh*

In this section:

- *Partitioning the unmeshable region and meshing the top-down regions*
- *Beginning the bottom-up mesh*
- *Continuing the bottom-up mesh*
- *Completing the mesh for the bottom-up region*
- *Checking the association of the bottom-up mesh with the geometry*
- *Associating the remainder of the bottom-up mesh*

Partitioning the unmeshable region and meshing the top-down regions

The example part (`bottomup_mesh_example_part.sat`) is imported from an ACIS file.

The file is included with the Abaqus installation, and you can use the following utility to obtain a copy:

```
abaqus fetch job=bottomup_mesh_example_part.sat
```

For more information about ACIS files, see *Importing parts from an ACIS-format file*.

Figure 1 shows the original part. There are three solid regions; the regions colored green and yellow can be meshed using the top-down structured and swept meshing techniques, respectively, and the orange region is unmeshable with the automated top-down techniques and hexahedral elements.

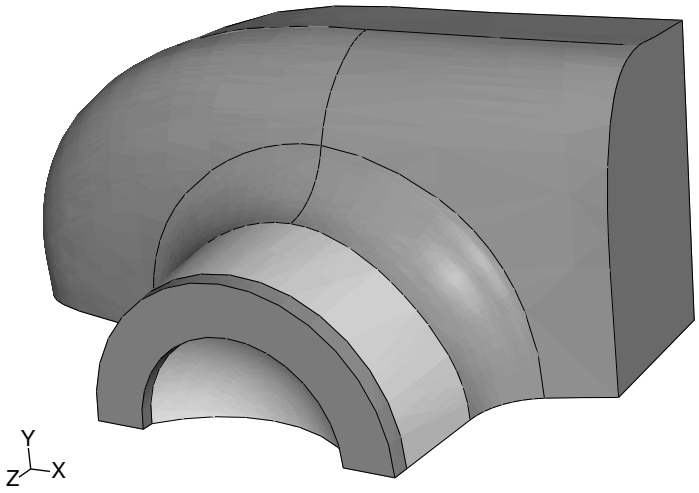


Figure 1: Default mesh color coding on the imported part.

To mesh the part, we will first create three partitions. The first two partitions create another top-down swept meshable region near the outer edge of the unmeshable region. The third partition is a face partition that you will use later as a vector for the bottom-up extrude method. You could apply bottom-up meshing techniques to the entire unmeshable region without partitioning. However, the resulting mesh would contain some poorly formed elements near the outer edge. In practice, you may create several bottom-up meshes before you make a satisfactory mesh for the entire region. Especially for more complex parts, you may want to save copies of a part with different meshing approaches until you decide which approach yields the best mesh.

1. Rotate the part so that you are viewing the bottom of the part, as shown in *Figure 2*.

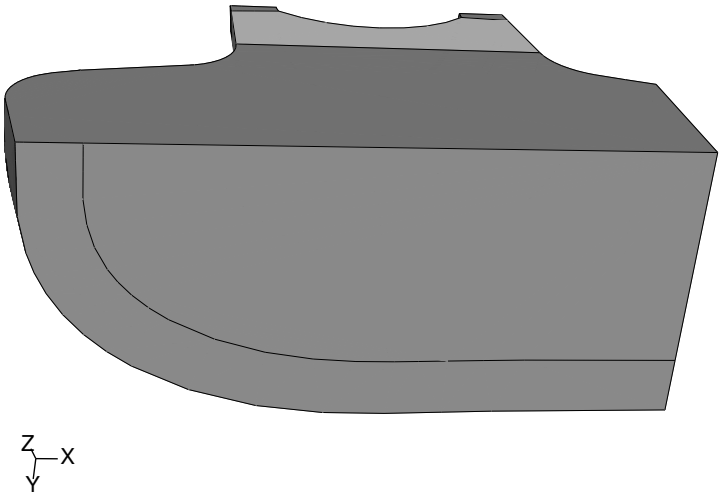


Figure 2: Creating an offset partition on the bottom face.

2. Complete the following steps to partition the face:

   a. From the partition face tools in the module toolbox, select the sketch method tool .
   b. Select the bottom face of the part, and select one of the straight edges to be vertical and on the right.

      Abaqus/CAE opens the Sketcher.

   c. From the Sketcher toolbox, select the offset tool .
   d. Select the curved edge, and click mouse button 2 to accept the selected edge.
   e. Enter `2.5` in the prompt area as the offset distance.

      Abaqus/CAE displays a preview of the offset partition.

   f. Click **OK** if the offset is shown correctly (toward the interior of the part), or click **Flip** if the offset is shown outside the part.
   g. Click mouse button 2 or click **Done** in the prompt area to partition the face.

   Abaqus/CAE returns to the Mesh module and displays the partitioned face as shown in *Figure 2*.

3. Extrude the face partition through the region.

   a. From the partition cell tools in the module toolbox, select the extrude/sweep method tool .
   b. Select the face partition created in Step 1 as the edge to extrude.
   c. Click **Extrude Along Direction** in the prompt area, and pick the edge shown in *Figure 3*.

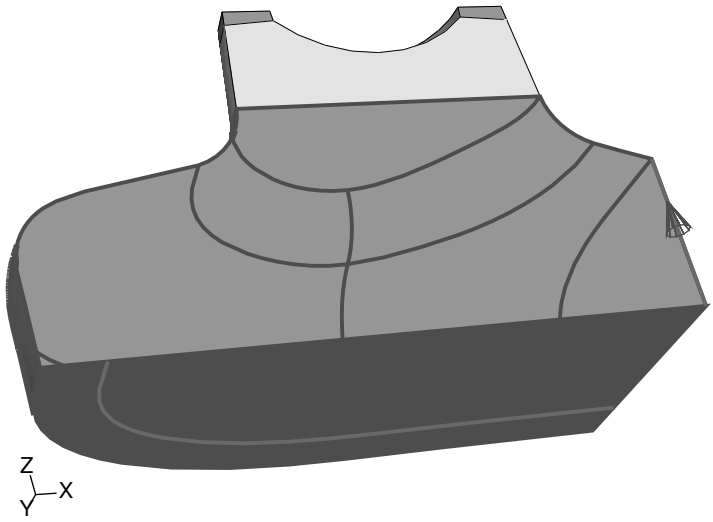


Figure 3: Extruding the face partition through the cell.

d. Click **OK** if the extrusion direction is shown correctly (through the region), or click **Flip** to change the direction.

e. Click **Create Partition** to partition the cell.

Abaqus/CAE displays the outer region of the part in yellow, indicating that it can now be meshed using top-down swept meshing.

4. Partition the front face of the unmeshable region as shown in *Figure 4*.

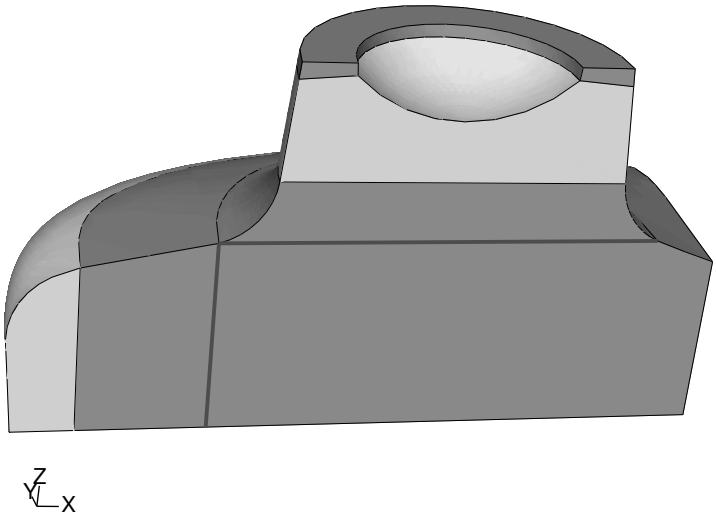


Figure 4: Creating face partitions.

a. From the partition face tools in the module toolbox, select the sketch method tool .

b. Select the front face of the unmeshable region, and select an edge to be vertical and on the right. Abaqus/CAE opens the Sketcher.

c. Use the vertical construction line tool  to create a construction line as shown in *Figure 5*.

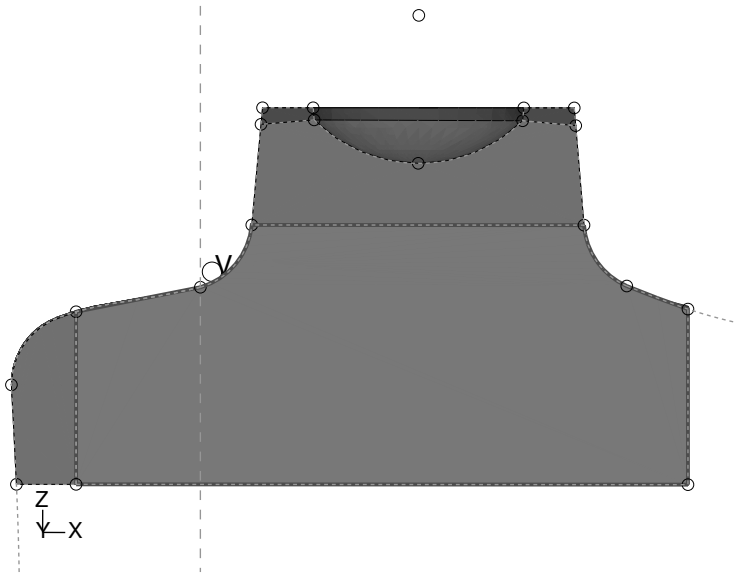


Figure 5: Using a vertical construction line to partition the face.

d. Use the connected lines tool  to create a line connecting the two points where the vertical construction line intersects the face of the part.

e. Create a second line connecting the upper point of the vertical line to the point where the fillet ends on the front face—the line should be nearly horizontal.

f. Click mouse button 2 or click **Done** in the prompt area to partition the face.

There are now three regions in the part that you can mesh using the top-down meshing techniques.

5. Assign global edge seeds to the entire part using an approximate size of `0.9` and the default settings for curvature control.

6. Select **Mesh->Part** from the main menu bar to mesh the top-down regions.

Abaqus/CAE highlights the unmeshable region and displays a warning that you cannot mesh it automatically.

7. Click **OK** to mesh the three top-down regions. *Figure 6* shows the resulting partial mesh.

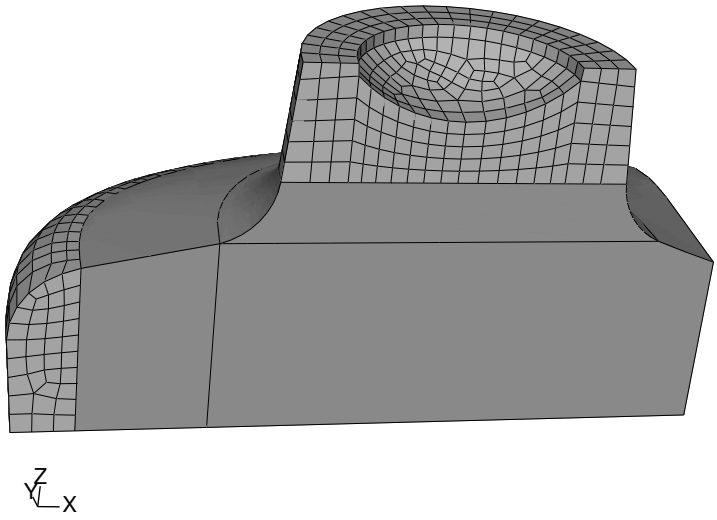


Figure 6: Automatic meshing of the top-down regions.

Beginning the bottom-up mesh

Now that the top-down mesh is complete, you can assign the bottom-up meshing technique and begin creating a hexahedral mesh for the rest of the part.

1. Use the mesh controls tool  to assign the bottom-up meshing technique to the unmeshable region. Abaqus/CAE colors the region light tan.
2. Select **Mesh->Create Bottom-Up Mesh** from the main menu bar.

   Tip: You can also click the  tool in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

   Abaqus/CAE displays the **Create Bottom-Up Mesh** dialog box and makes the unmeshed regions translucent. The degree of translucency is determined by the translucency slider, located in the **Color Code** toolbar (for more information, see *Changing the translucency*). In a more complex part, translucency allows you to select internal regions more easily.
3. Select the **Sweep** method, and click **Select** to the right of **Source side**.
4. Select the semi-circular face at the bottom of the protrusion, colored magenta in *Figure 1*, as the source side of the first bottom-up mesh; and click **Done** in the prompt area.

   You must change the default selection options to select the interior face (for more information, see *Using the selection options*).

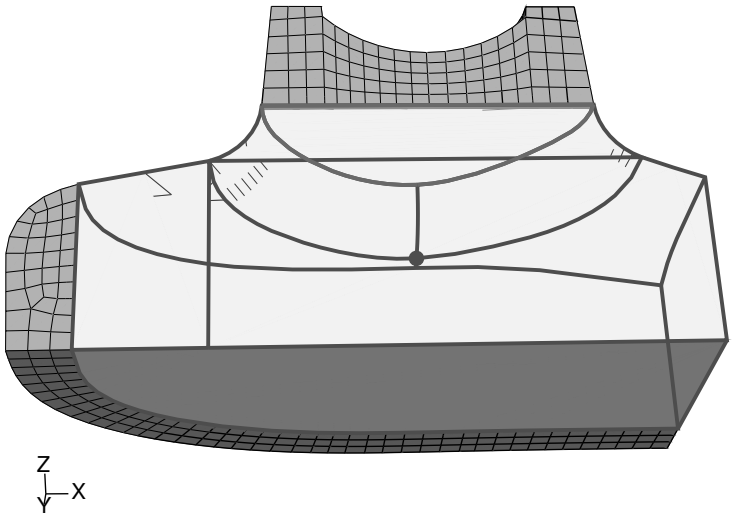


Figure 1: Selecting a geometric face as the source side.



Tip: You can also choose two-dimensional elements or element faces as sides of a bottom-up mesh. In this case an alternative source side selection would be all the element faces on the bottom of the swept mesh.

5. Toggle on **Connecting sides**, and click **Select**. Rotate the part as needed to select the three faces that represent the fillet between the top protrusion and the body of the part and the corresponding flat face on the front of the part, colored yellow in *Figure 2*.

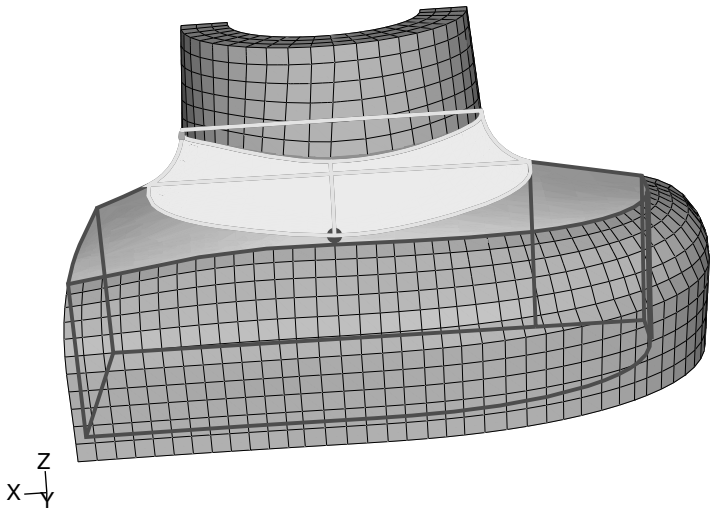


Figure 2: Selecting the connecting sides.

Click **Done** in the prompt area to end selection.

6. Click **Mesh** in the **Create Bottom-Up Mesh** dialog box to create the mesh.

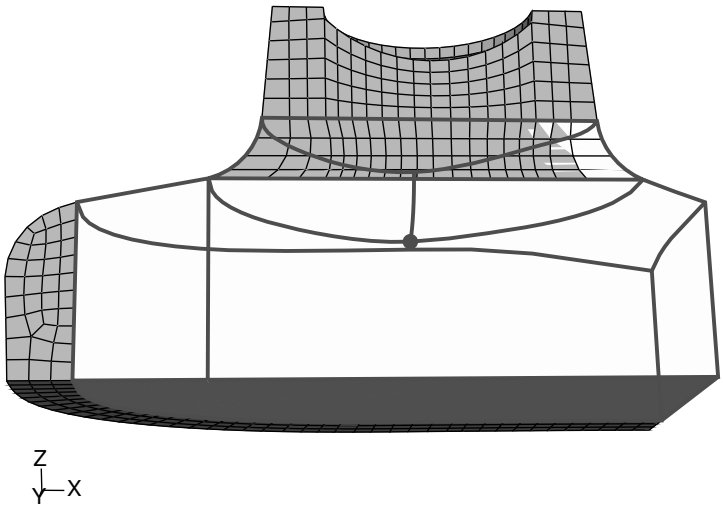


Figure 3: Bottom-up swept mesh of the filleted area.

The mesh extends approximately as far into the region as the selected connecting sides, the bottom-up region remains selected, and the **Create Bottom-Up Mesh** dialog box remains open for the next step.

Continuing the bottom-up mesh

There are at least three methods that you can use to complete the mesh for the part. The simplest method would be to create another swept mesh, using the bottom of the mesh that you just completed and the two faces extending out from the fillet on the top of the remaining unmeshed area as the source side, the three geometric vertical faces and the set of vertical element faces as the connecting sides, and the unmeshed portion of the bottom face as the target side. This method would complete the part mesh in a single bottom-up meshing step, and the elements would be fully associated with the selected faces. You could also use the bottom-up extrude method with the bottom face of the part as the extrude distance. However, for demonstration purposes we will use a longer process that combines use of the bottom-up extrude method, the associativity tool, and the Edit Mesh toolset.

Create a bottom-up extruded mesh

1. Select the **Extrude** method, and click **Select** to the right of **Source side**.
2. Select the element faces at the bottom of the previous bottom-up mesh, as shown in *Figure 1*, as the source side for the second bottom-up mesh.

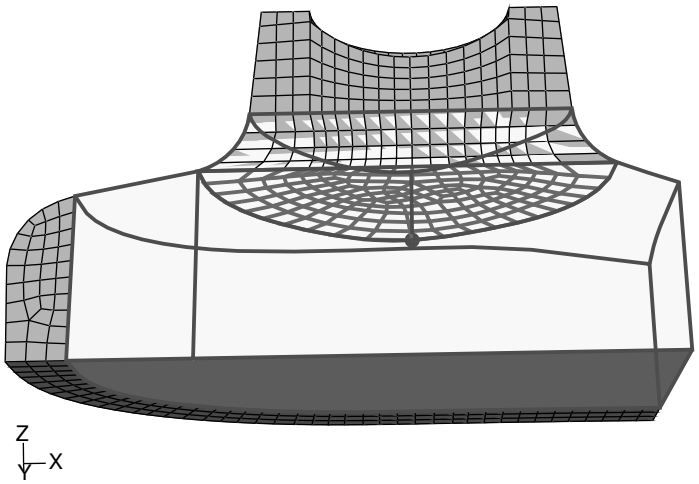


Figure 1: Selecting element faces as the source side.

3. Click **Done** in the prompt area to accept the selected faces.

   The **Create Bottom-Up Mesh** dialog box reappears.
4. Click the **Select** button to choose a vector for the extruded mesh.
5. Select the upper endpoint of the partition as the starting point of the vector, and select the lower endpoint as the end of the vector. *Figure 2* shows the extrusion vector.

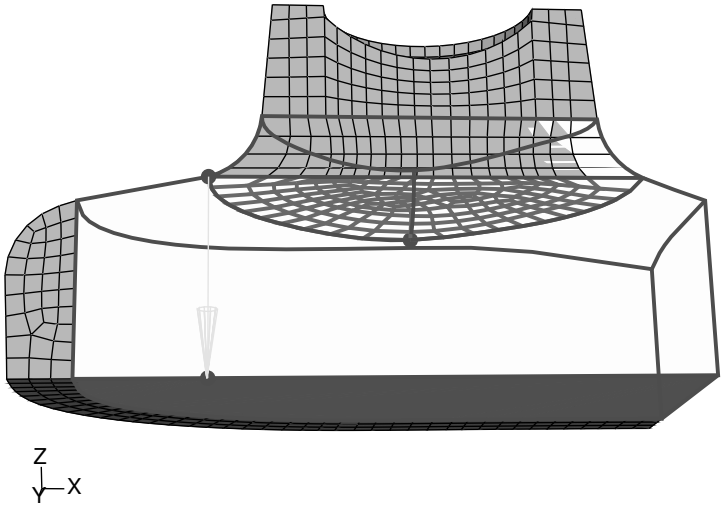


Figure 2: Selecting the vector for the bottom-up extruded mesh.

6. Enter `10` for the **Number of layers**.

   There are 10 elements on the inner face of the top-down swept mesh. Using the same number of elements for the extruded mesh will provide a better match when you create the third, and final, bottom-up mesh.

7. Verify that the extrude depth is set using the default **Use vector length** method, and click **Mesh** in the **Create Bottom-Up Mesh** dialog box to create the mesh.

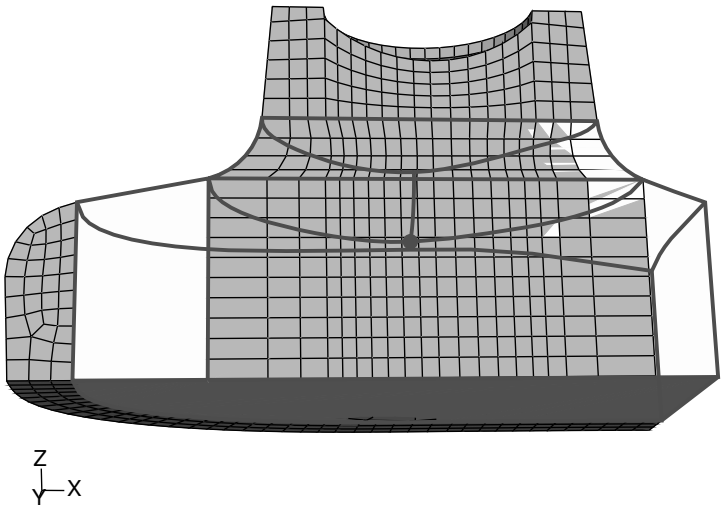


Figure 3: The bottom-up extruded mesh.

Extend the extruded mesh

The extruded bottom-up mesh ends near the bottom face of the region, as dictated by the nonplanar source side and the length of the extrusion vector. You can edit the nodes in the last extruded element layer so that they end exactly on the bottom face of the region.

1. Select the Edit Mesh toolset , located at the bottom of the Mesh module toolbox.
2. Select the **Node** category, and click on **Project** in the **Edit Mesh** dialog box.
3. Use the *angle method* to select all the nodes on the bottom of the bottom-up extruded mesh, as shown in *Figure 4*, and then click **Done** in the prompt area.

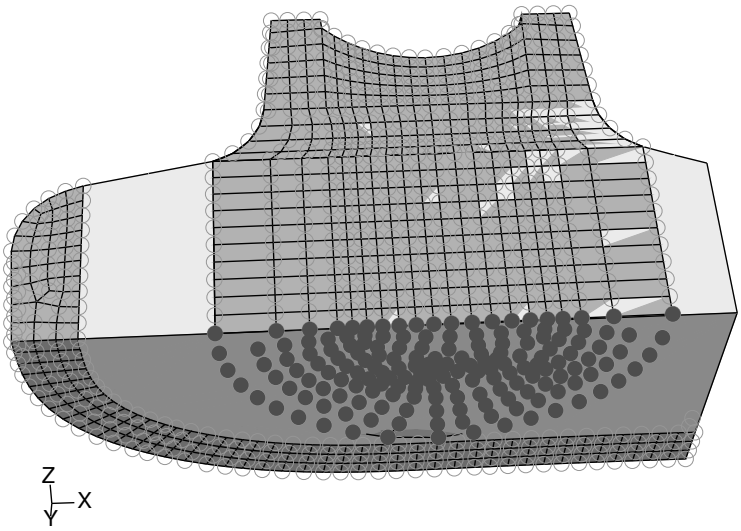


Figure 4: Selecting the nodes to project.

4. Select the bottom face of the bottom-up region, and click **Yes** in the prompt area to project the nodes onto the face.

Completing the mesh for the bottom-up region

We will now create a final swept mesh to complete the mesh for the part.

1. From the main menu bar, select **Mesh->Create Bottom-Up Mesh**.
2. Select the bottom-up region.

   Abaqus/CAE displays the **Create Bottom-Up Mesh** dialog box.
3. Select the **Sweep** method, and click **Select** to the right of **Source side**.
4. Select the two remaining unmeshed faces on the top of the part, as shown in *Figure 1*, and click **Done** in the prompt area.

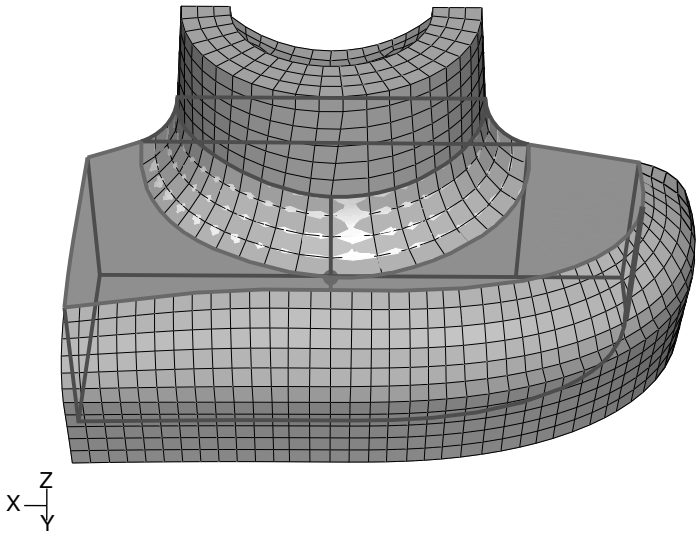


Figure 1: Selecting the source side for the final bottom-up mesh.

5. Toggle on **Connecting sides**, and click **Select**. Rotate the part, and use the angle method to select the interior element faces from the outer swept mesh and the exterior element faces of the extruded mesh as shown in *Figure 2*.

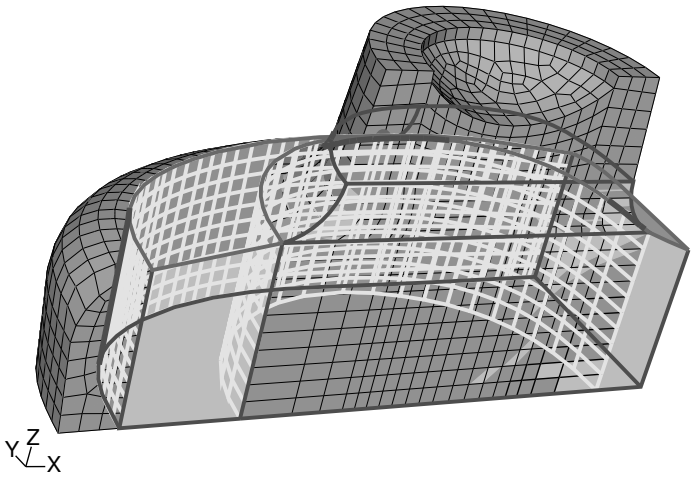


Figure 2: Selecting the element faces as connecting sides.

Click **Done** in the prompt area to end selection.

6. Toggle on **Target side**, and click **Select**.
   Abaqus/CAE prompts you to select a target side for the swept mesh.
7. Select the bottom face of the region.
8. Click **Mesh** in the **Create Bottom-Up Mesh** dialog box to create the mesh.

The part is now completely meshed with hexahedral elements, as shown in *Figure 3*.

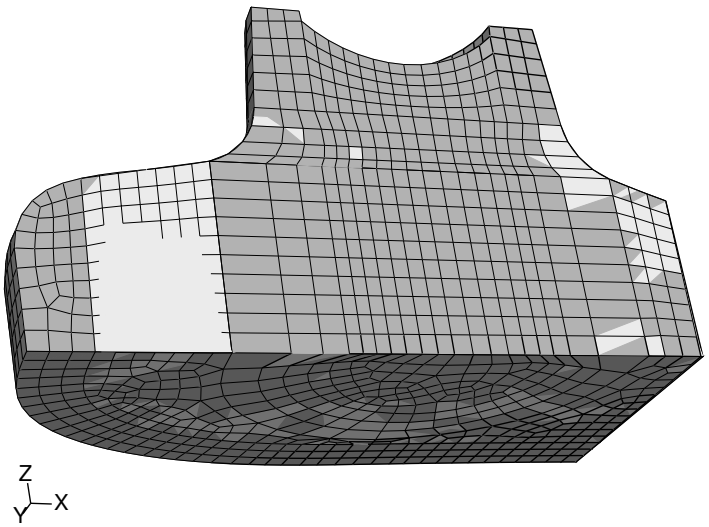


Figure 3: The final meshed part.

Checking the association of the bottom-up mesh with the geometry

Before using the completed bottom-up mesh for an analysis, you should check the association between the region geometry and the bottom-up mesh elements. Loads, boundary conditions, and other attributes in Abaqus are applied to geometry, and they will not transfer correctly to the elements of a bottom-up mesh unless the mesh is correctly associated with the geometry. At minimum, you should check the association for areas of a bottom-up mesh where loads and boundary conditions are applied.

In most cases if you select a geometric feature, such as a face, to define the bottom-up mesh, Abaqus/CAE automatically associates the appropriate elements with that face. However, in cases where the geometry is not used, such as the extruded bottom-up mesh in the center of the example part, the elements are associated only with the region, not the nearby face where a load might be applied. (For more information, see *Mesh-geometry association*.) The following procedure associates the elements on the bottom and front faces of the part with the geometric faces:

1. From the main menu bar, select **Mesh->Associate Mesh with Geometry**.
2. Select the bottom face of the bottom-up region, as shown in *Figure 1*.

   The elements created in the final bottom-up meshing step are colored yellow because they were already associated when the face was used as the target side of the final bottom-up swept mesh. However, the elements in the semi-circular extruded mesh are not associated with the face.

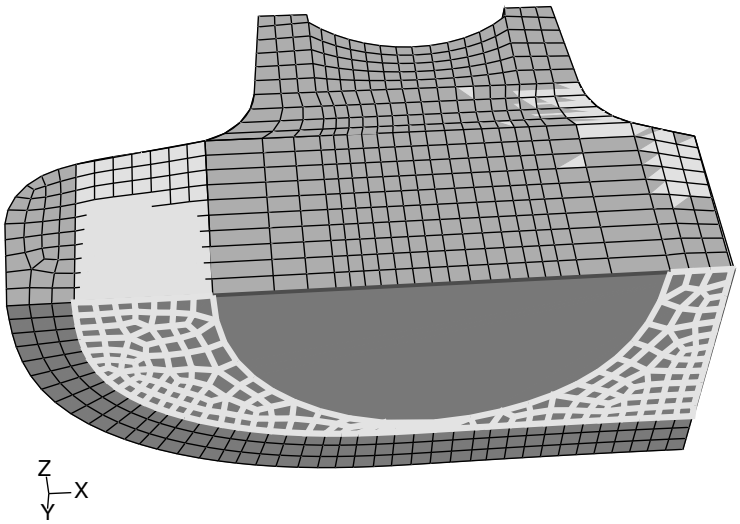


Figure 1: Existing mesh associations on the bottom face.

3. Remove the top-down meshed cell that extends from the bottom-up region to the outer curved face of the part.
   a. Select **Tools->Display Groups->Create**.

      Abaqus/CAE displays the **Create Display Group** dialog box.
   b. Select **Cells** from the item list, and click **Edit Selection**.
   c. From the viewport, pick the top-down swept region that extends along the curved outer edge of the part and click **Done** in the prompt area.
   d. In the **Create Display Group** dialog box, click the **Remove** button.

      Abaqus/CAE removes the selected cell from the viewport.

   For more information, see *Using display groups to display subsets of your model*."

4. Use the angle method to select all the faces on the bottom of the bottom-up region. When you are finished, all of the element faces on the bottom of the bottom-up region should be colored yellow, as shown in *Figure 2*.

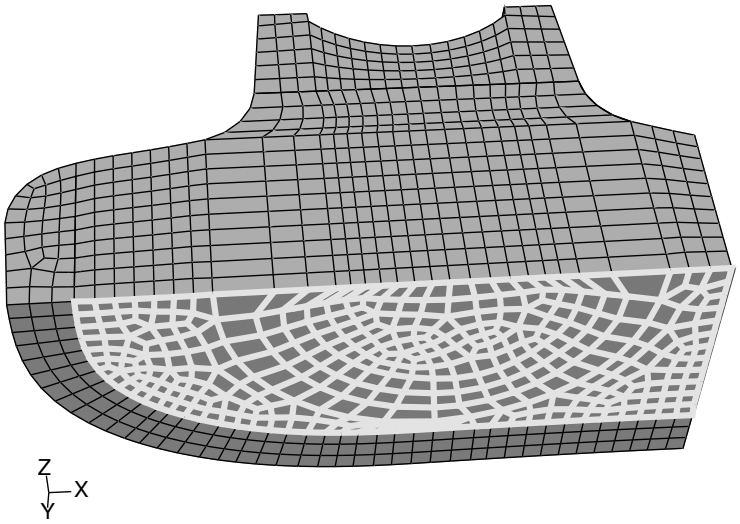


Figure 2: Final mesh associations on the bottom face.

Note:

If you did not remove the outer top-down meshed cell, use of the angle method would have selected the top-down element faces for association as well as the bottom-up faces, leading to an error message when you attempted to associate the faces.

5. Click **Done** in the prompt area to associate the selected elements' faces with the region face.

Associating the remainder of the bottom-up mesh

To fully associate the bottom-up mesh, you should verify and edit the association for the faces on the front side, the right side, and on each of the bounding edges and vertices in the bottom-up region. You associate edges and vertices the same way as you associated the element faces in the previous section, except that you select edges and element edges or vertices and nodes, respectively. Abaqus/CAE attempts to associate nodes with vertices and element edges with geometric edges based on proximity. If all the elements along an edge are associated with the faces bounded by that edge, Abaqus/CAE automatically associates the element edges with the geometric edge.

Once you have a part completely meshed and you have checked the association of the bottom-up mesh, you should save the model. Prior to running an analysis, you should also verify the quality of the mesh. Mesh verification helps ensure that there are no hidden problems, regardless of whether you created the mesh automatically (top-down) or used bottom-up techniques and mesh editing to construct the mesh. For more information on mesh verification, see *Verifying element quality*.

Mesh-geometry association

You can create an association between orphan or bottom-up mesh entities and adjacent geometry. Abaqus/CAE automatically associates top-down meshes with the underlying geometry. Abaqus/CAE can make this association since the mesh conforms exactly to the geometry. In contrast, a bottom-up mesh does not have to conform to geometry and, therefore, Abaqus/CAE may not associate parts of the mesh with the geometry. Similarly, orphan mesh entities are not associated with geometry because they were either created independent of geometry or the geometry from which they were created is not available in the current model.

Creating an association between orphan or bottom-up mesh entities (elements, element faces, element edges, and nodes) and adjacent geometry allows the transfer of loads, interactions, and boundary conditions from the geometry to the mesh. If you fully associate orphan or bottom-up mesh entities with an adjacent geometric face, you can use that face to create a native mesh for the geometry region that is compatible with the orphan or bottom-up mesh with which you started. Full association means that:

- The selected geometric face is associated with element faces that cover the entire face.
- All edges of the geometric face are associated with element edges that span the entire edge.
- All vertices of the face are associated with nodes.

Figure 1 shows an example of the use of mesh-geometry association between a two-dimensional orphan element region and an adjoining geometric region. Associating the geometric edge—the yellow line in the left image—between the two regions with the orphan element edges and nodes results in a compatible hybrid mesh.

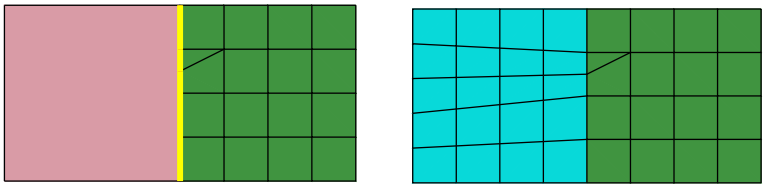


Figure 1: Associating an edge enforces creation of a compatible mesh.

Conversely, if a model has a mesh that you need to preserve—because it includes extensive edits or is ideally suited to an analysis—and you do not want to make an orphan mesh for the entire part, you can delete the association between the mesh and selected entities to create orphan nodes and elements in selected regions. Deleting mesh associativity prevents Abaqus/CAE from deleting the mesh of a region when you edit the geometry. After making your edits, you can reestablish associativity between the geometric faces and the surface entities of the mesh. Nodes along edges are merged with any nodes that currently exist on the geometry.

Note:

If you delete the mesh-geometry associativity for a solid region, the only ways to reestablish a native mesh are to use the **Undo** function in the **Edit Mesh** dialog box or to assign the bottom-up mesh technique to the region and recreate the associativity.

The following rules apply to the association of bottom-up elements when you work with a region of solid geometry:

- Abaqus/CAE always associates bottom-up elements with the selected region.
- When the underlying geometry is used to define the shape of the mesh, Abaqus/CAE associates the mesh with that geometry.
- When portions of the geometry are not used to define the bottom-up mesh, Abaqus/CAE does not associate them, even if the mesh and geometry are in the same location.
- You can edit the mesh-geometry association of any geometry-based bottom-up meshed region.

- Even if you edit a generated mesh so that it matches the geometry, you must still manually associate the mesh with the geometry.

There are three considerations that make mesh-geometry association critical to creating a good analysis model:

1. When you work with geometry, attributes such as loads and boundary conditions are applied to the geometry. Proper mesh-geometry association ensures that these attributes are transferred correctly to the mesh during the analysis.
2. If you select a geometric face as a source or connecting side of a bottom-up mesh, Abaqus reuses the existing mesh entities on that face to create a compatible mesh only if the mesh entities are fully associated with the selected face.
3. Abaqus tries to merge meshes that are associated with the same geometric entities. Unassociated meshes may require you to merge nodes along mesh boundaries using the Edit Mesh toolset.

You should always check that the mesh-geometry associativity is correct when working with bottom-up meshes or editing the associativity of any mesh. Abaqus/CAE will issue an error in the Job module if you submit a job and attributes are applied to geometry with no associated mesh. However, Abaqus/CAE cannot determine whether the association is correct. For example, if a load is applied to a geometric face that should have several hundred elements but only one element is associated with that face, Abaqus/CAE will attempt to analyze the model with the entire load applied at the single associated element. Incorrect association produces incorrect analysis results.

You can use the mesh-geometry association  and delete mesh associativity  tools in the Mesh module toolbox to view and edit or to delete mesh-geometry associations. For detailed instructions, see *Viewing and editing mesh-geometry associativity* and *Deleting mesh-geometry associativity*.

Understanding adaptive remeshing

This section describes how you perform adaptive remeshing in Abaqus/CAE. Adaptive remeshing is described in detail in *Adaptive Remeshing*.

In this section:

- *What are remeshing rules?*
- *Which mesh controls can I use with adaptive remeshing?*
- *Which procedures can I use with adaptive remeshing?*
- *What is the difference between automatic adaptive remeshing and manual adaptive remeshing?*
- *When do I need to use manual adaptive remeshing?*

What are remeshing rules?

Remeshing rules enable Abaqus/CAE to adapt your mesh iteratively to meet error indicator goals that you have specified. You can allow Abaqus/CAE to perform the iterative remeshing and analysis operations, or you can remesh manually and study the effect of your remeshing rule on the mesh and the resulting analysis. Abaqus/CAE remeshes the faces and cells to which you assigned an adaptivity rule and any adjacent faces or cells; the mesh on other regions does not change. For more information, see *About Adaptive Remeshing*.

A remeshing rule describes all aspects of your adaptive meshing specification:

- The region to which the remeshing rule is applied. You can apply a remeshing rule to the entire model or to selected regions.
- A specific step during which Abaqus/CAE will apply the rule. The remeshing rule will be applied only during this step; however, you can apply a different remeshing rule with the same settings to another step in your model.
- The error indicator output variables—the output variables that will be used to calculate the error estimate. For more information, see *Selection of Error Indicators Influencing Adaptive Remeshing*.
- The sizing method—the method that Abaqus/CAE will use to calculate the size of the elements in the mesh. For more information, see *Solution-Based Mesh Sizing*.
- Any constraints on the remeshing calculations.

A remeshing rule works in combination with the edge seeding, element type, and meshing method to determine the mesh at a particular adaptivity iteration. Remeshing rules are stored in the model database and are maintained between sessions. To create a remeshing rule, select **Adaptivity**->**Remeshing Rule**->**Create** from the main menu bar. For more information, see *Creating a remeshing rule*.

You can define multiple remeshing rules over multiple regions of your model. If you apply multiple remeshing rules to the same region of a model, Abaqus/CAE applies a conservative element size specification, and the rule that defines a finer mesh at a particular point takes precedence. If you assign a remeshing rule to a dependent instance, Abaqus/CAE remeshes the original part and each dependent instance of the part inherits the same mesh.

Abaqus/CAE requests error indicator output variables in every job that you create while a remeshing rule is active. The remeshing rule has no effect on the mesh during the first job. However, during the first job Abaqus uses the remeshing rule to calculate the error indicator output variables. In subsequent adaptive remesh iterations the remeshing rule augments your mesh size specification to produce a mesh that attempts to optimize element size and placement to achieve the error indicator goals described in the rule.

Which mesh controls can I use with adaptive remeshing?

Table 1 shows the mesh controls that must be assigned to a region that will use adaptive remeshing.

Table 1: Mesh controls and adaptive remeshing.

Dimensionality	Element Shape	Algorithm	Technique
Two-dimensional	Triangular	n/a	Free
Two-dimensional	Quad-dominated	Advancing front	Free
Three-dimensional	Tetrahedral	n/a	Free

In addition, you must assign the above supported mesh controls to any face or cell that is adjacent to a region that is included in a remeshing rule.

When Abaqus/CAE generates a tetrahedral mesh, it first creates a boundary mesh of triangles on the exterior faces of the solid region and then generates the tetrahedral mesh using the triangles as faces of the exterior tetrahedral elements. When you assign adaptive remeshing to a solid region, Abaqus/CAE applies adaptive remeshing to the boundary mesh of triangles on the exterior faces. For more information, see *Free meshing with triangular and tetrahedral elements*.

If Abaqus/CAE is using the mapped meshing technique in a region you have selected for adaptive remeshing, the mesh resizing algorithm may be slow to converge and require more remeshing iterations to achieve a given target error. To prevent Abaqus/CAE from using mapped meshing, you can toggle off **Use mapped meshing where appropriate** in the **Mesh Controls** dialog box. For more information, see *Setting the mesh algorithm*.

Which procedures can I use with adaptive remeshing?

Adaptive remeshing is available only for Abaqus/Standard. In addition, adaptive remeshing is available only for the following Abaqus/Standard procedures:

- Static (general and linear perturbation)
- Quasi-static
- Heat transfer
- Fully coupled thermal-stress
- Coupled pore fluid diffusion-stress
- Coupled thermal-electrical

What is the difference between automatic adaptive remeshing and manual adaptive remeshing?

If you choose to allow Abaqus/CAE to remesh your model iteratively, the adaptivity process in the Job module controls the adaptive remeshing for you. You need only define the remeshing rule in the Mesh module and apply the rule to the regions in your model that you want to be remeshed.

Conversely, you can manually apply modified remeshing rules and view the impact of your modifications on the generated mesh. When you are satisfied that your remeshing rule is producing the desired mesh, you can use the rule to drive a sequence of iterative remeshing and analysis operations that is controlled by Abaqus/CAE. Select **Adaptivity->Manual Adaptive Remesh** from the main menu bar to apply the adaptive remesh rules manually. For more information, see *Manually resizing and remeshing*.

When you apply manual adaptive remeshing, you must enter the name of the output database file that was generated by a previous analysis of your model. This output database contains the error indicator output variables that Abaqus/CAE uses to calculate the mesh sizing functions. The error indicator output variables stored in the output database limit the changes that you can make to a remeshing rule. For example, if your original rule specified energy density in a certain region, you will not be able to switch the rule to use error in equivalent plastic strain without first rerunning the analysis. You can modify the sizing method and the element size constraints in the remeshing rule and still use the output database from a previous analysis. However, the output database cannot be used if you modify the step, the region, or the error indicator output variables.

When do I need to use manual adaptive remeshing?

You can use manual adaptive remeshing to do the following:

- To learn about the impact of various size function and error indicator output variables on the mesh generated by Abaqus/CAE.
- If your analysis is expected to take a long time, you can close the Abaqus/CAE session and make the license tokens available for another user or for a new session. However, to continue the adaptive process, you must read the output database generated by the analysis and manually remesh the model.
- If the analysis ended prematurely (for example, because of insufficient memory), you can use manual remeshing to continue the adaptivity process.

Advanced meshing techniques

This section contains information about how to accomplish advanced meshing tasks that are not straightforward.

In this section:

- *Meshing multiple three-dimensional solid regions*
- *Meshing multiple two- and three-dimensional shell regions*
- *Compatible meshes between part instances*
- *Parametric modeling*
- *Meshing complex solids with hexahedral elements*

Meshing multiple three-dimensional solid regions

Abaqus/CAE assigns a default meshing technique to each region depending on its geometry and topology. However, sometimes the default meshing techniques applied to adjacent regions of a three-dimensional part or part instance are not compatible, and Abaqus/CAE cannot generate a compatible mesh.

For example, Abaqus/CAE cannot generate a compatible mesh over the entire part instance in *Figure 1* using the default meshing techniques because the nodes from the structured mesh on the left cannot be merged with the nodes of the swept mesh on the right. (The cube on the right side of the part instance is a swept region because it is joined to the cylinder, which is also a swept region.)

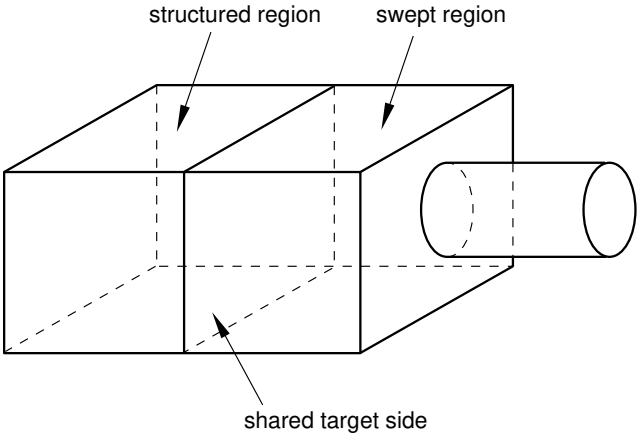


Figure 1: A compatible mesh is impossible using these default meshing techniques.

The mismatch that would occur between the nodes of the structured region and the nodes of the swept region is obvious if you mesh the two regions separately, as shown in *Figure 2*.

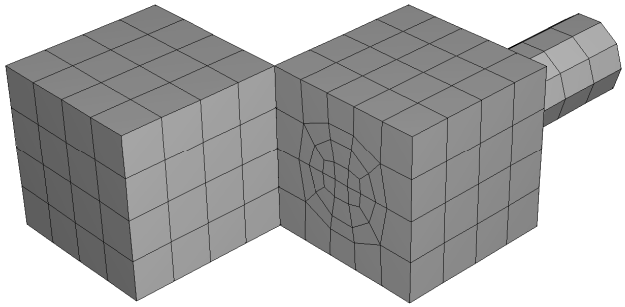


Figure 2: The structured region and the swept region meshed separately.

If you initiate the meshing procedure and Abaqus/CAE cannot generate a compatible mesh using the default meshing techniques, Abaqus/CAE attempts to replace the default meshing techniques with new meshing techniques. These new techniques are determined not only by the region's geometry and topology but also by the characteristics of neighboring regions in the part or part instance. Abaqus/CAE evaluates the interfaces between regions and tries to minimize the number of incompatible interfaces.

For example, the default meshing technique for the cube on the left side of the part instance in *Figure 1* is structured, and the resulting incompatible mesh is shown in *Figure 2*. However, this cube can also be meshed using the swept meshing technique. Therefore, Abaqus/CAE changes the meshing technique assigned to this region from structured

to swept, and a compatible mesh is generated over the whole part instance. (The element shapes assigned to a region remain unchanged when Abaqus/CAE changes the meshing technique assigned to the region.)

When you initiate the meshing procedure for a three-dimensional part or part instance, Abaqus/CAE determines if a compatible mesh can be generated using the default techniques assigned to each region. If a compatible mesh is possible, meshing proceeds. If a compatible mesh cannot be generated using the default techniques, Abaqus/CAE checks to see if it can replace the default meshing techniques with different techniques that will allow a compatible mesh to be generated.

- If different techniques will allow a compatible mesh, Abaqus/CAE highlights the incompatible interfaces and prompts you to select one of the following options:
  - Cancel the meshing procedure.
  - Allow Abaqus/CAE to replace the default techniques as necessary and generate a compatible mesh.
  - Allow Abaqus/CAE to use the default meshing techniques and automatically generate tie constraints across the incompatible interfaces. Abaqus/CAE automatically chooses one side of the interface as the secondary surface and the other as the main surface for the automatically generated tie constraint but creates common (merged) nodes on the perimeter of the incompatible interface. The nodes on the secondary surface are constrained to have the same value of displacement, temperature, pore pressure, or electrical potential as the point on the main surface to which they are tied. Abaqus/CAE generally selects the surface with the finer mesh to be the secondary surface. The computation for the depth of the secondary node adjustment zone for the tie constraint is based on the bounding dimensions of the interfacing regions. (For more information on tie constraints, see *Mesh Tie Constraints*.)
- If different techniques will still not allow a compatible mesh, Abaqus/CAE highlights the incompatible interfaces and prompts you to select one of the following options:
  - Cancel the meshing procedure.
  - Automatically generate tied surface interactions across the incompatible interfaces, as described above.

If a compatible mesh cannot be generated, you can try one of the following approaches:

- Partition as necessary to generate a compatible mesh.
- Use the free meshing technique to mesh the entire part or part instance.

In general, the following restrictions apply to generating a compatible mesh on a three-dimensional solid part or part instance:

- A swept region cannot share its target side with a structured region. However, it can share a source side or a connecting side with a structured region, as shown in *Figure 3*.

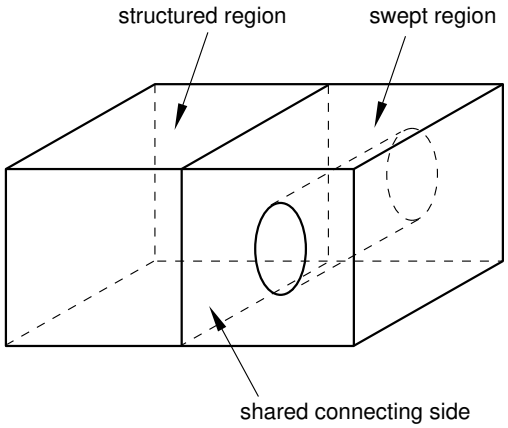


Figure 3: The connecting side of the swept region is shared with the structured region.

- In some situations Abaqus/CAE cannot mesh a part or part instance that contains multiple regions that have all been assigned the swept meshing technique. For example, Abaqus/CAE cannot sweep a mesh along the part instance shown in *Figure 4* because a compatible mesh cannot be generated on the shared target face.

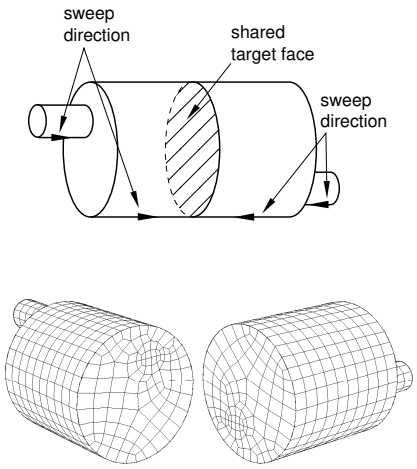


Figure 4: A compatible swept mesh cannot be generated along this part instance.

However, *Figure 5* shows how you can use partitions to produce a mesh that incorporates four swept regions.

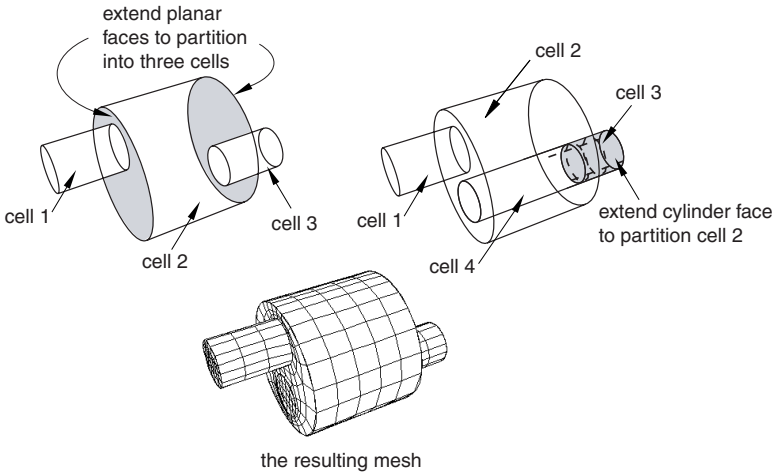


Figure 5: Using partitions to generate a compatible swept mesh.

Different regions of the same part or part instance can be meshed with hexahedral and tetrahedral elements as shown in *Figure 6*.

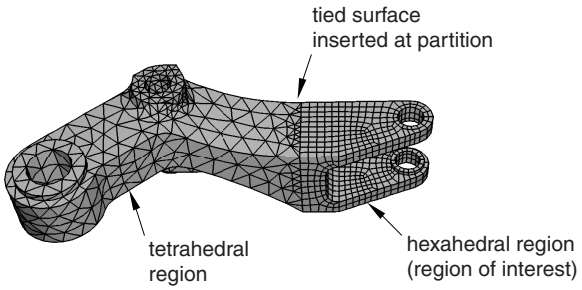


Figure 6: Different regions of the same part or part instance can be meshed with hexahedral and tetrahedral elements.

You can use the hexahedral elements where accuracy is important, such as adjacent to contact surfaces or in areas of special interest that require a fine mesh. You can use tetrahedral elements in other regions, and Abaqus/CAE creates tied surfaces where the regions connect. When you mesh one region, Abaqus/CAE does not adjust an existing mesh on adjacent regions.

Meshing multiple two- and three-dimensional shell regions

Meshing multiple three-dimensional solid regions, describes how the default meshing techniques applied to adjacent regions of a three-dimensional solid part or part instance may not allow you to generate a mesh that is compatible across the regions. In contrast, adjacent regions of a two- or three-dimensional shell part or part instance are always compatible.

Figure 1 illustrates a three-dimensional shell part instance with adjacent regions that Abaqus/CAE can mesh using the free, swept, and structured meshing techniques.

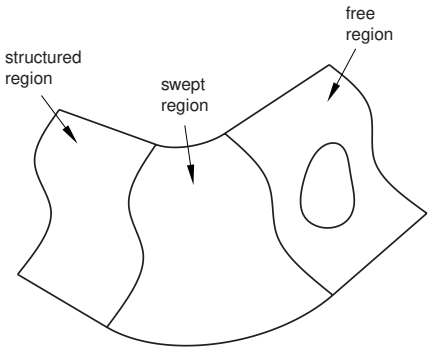


Figure 1: Adjacent regions of a three-dimensional shell part instance.

Figure 2 illustrates the resulting mesh.

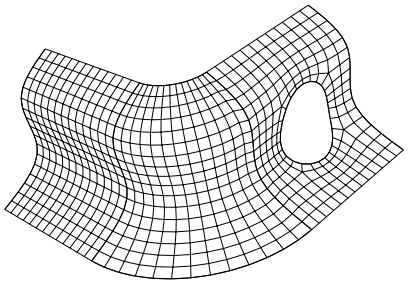


Figure 2: The resulting mesh.

Compatible meshes between part instances

You cannot prescribe meshes that are compatible between part instances. If you require mesh compatibility between two or more instances, you can do one of the following:

- Create a single part that contains all the bodies so that multiple instances are not necessary.
- Assemble instances of the parts in the Assembly module and use the **Merge/Cut** tool to merge the instances into a single instance. If you need to maintain the concept of separate part instances, you can create a partition at the common interface of the merged instances.

  Similarly, you can create a single part instance from multiple instances containing orphan elements and nodes, using the **Merge/Cut** tool to merge duplicate nodes. For more information, see *Performing Boolean operations on part instances*.
- If you must use separate parts, you can use tied contact to avoid the issue of mesh compatibility. Keep in mind that this is not true compatibility, and the accuracy of the solution may suffer. For more information on tied contact, see *Understanding interactions*.

Parametric modeling

A useful feature of the Mesh module is the ability to regenerate partitions and mesh attributes—such as element type assignments, seeds, and mesh controls—after a part has been modified. (You must always recreate the mesh itself after modifying a model.)

For example, the model shown in *Figure 1* has been partitioned into four regions and then seeded to specify an approximate element size of 3.

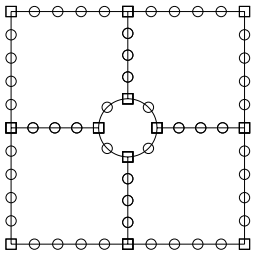


Figure 1: Seeded model with small hole.

You can return to the Part module and modify the hole at the center of the model so that it is slightly larger. When you return to the Mesh module, the partitions and the seeds are regenerated, as shown in *Figure 2*.

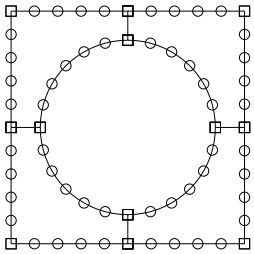


Figure 2: Seeds are regenerated after the part is modified.

In addition, settings in the **Mesh Controls** and **Element Type** dialog boxes (such as element shape, element type, and meshing technique) are also regenerated. (You can display these two dialog boxes by selecting **Mesh->Controls** and **Mesh->Element Type** from the main menu bar.)

Note:

If you drastically modify the part, the seeds and partitions may fail to regenerate. In these cases you must create new seeds and partitions after reentering the Mesh module.

Meshing complex solids with hexahedral elements

You can use the Part module to create complex solid revolved parts that include a translation along the axis of revolution, and you can also create solid extruded parts that include a twist about a selected center point. Free meshing allows you to mesh these parts with tetrahedral elements, as described in *Free meshing with triangular and tetrahedral elements*. In addition, you can use hexahedral elements to mesh an extruded part that is twisted, as shown in *Figure 1*.

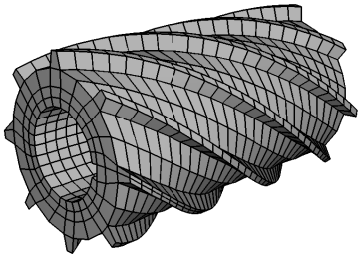


Figure 1: A hexahedral swept mesh on an extruded part with twist.

However, if you want to use hexahedral elements to mesh a revolved part that is translated, you will probably have to introduce partitions to make the part swept meshable, as described in the following paragraphs.

For the Mesh module to create a three-dimensional swept mesh of hexahedral elements, every side that connects the source side to the target side must contain only a single face (see *Figure 4* in *Swept meshing of three-dimensional solids*). However, when you create a part that is revolved more than 180° and then translated along the axis of revolution, Abaqus/CAE inserts rings along the length of the solid. These rings exist only on the surface of the part, and they do not create faces that cut through the part. As a result, the side that connects the source side and the target side now contains more than one face, and the part is not swept meshable with hexahedral elements unless you introduce partitions.

For example, *Figure 2* shows a part that represents a coil spring. The figure also shows the rings that Abaqus/CAE inserted when the part was created. These rings divide the connecting face between the source side and the target side; as a result, you cannot mesh an instance of the coil spring with a swept mesh of hexahedral elements until you partition the part.

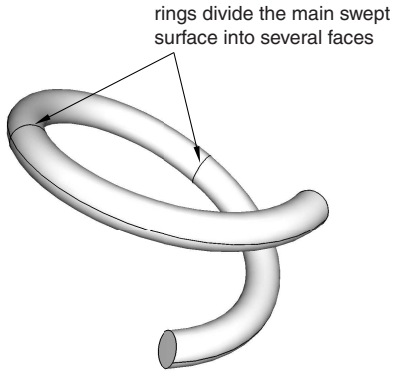


Figure 2: Rings divide the coil into several segments.

The partitioning operation introduces partitions through the solid coil using N-sided patches to define the new faces. You select the rings to define the N-sided patches, as shown in *Figure 3*.

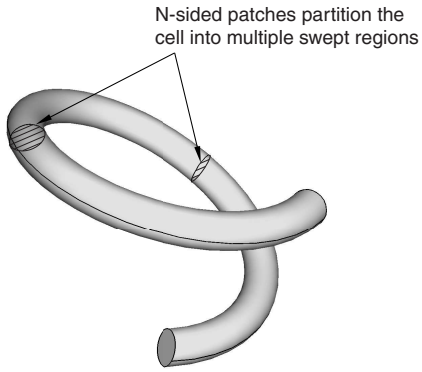


Figure 3: N-sided patches partition the cell.

You can now seed the part instance and generate a swept mesh using hexahedral elements, as shown in *Figure 4*.

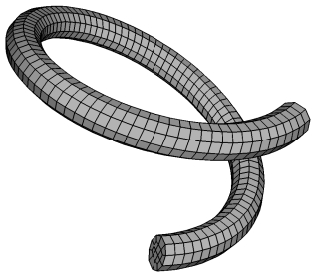


Figure 4: The model is swept meshed with hexahedral elements.

Using the Mesh module toolbox

You can access all the Mesh module tools through either the main menu bar or the toolbox. *Figure 1* shows the hidden icons for all the tools in the Mesh module toolbox.

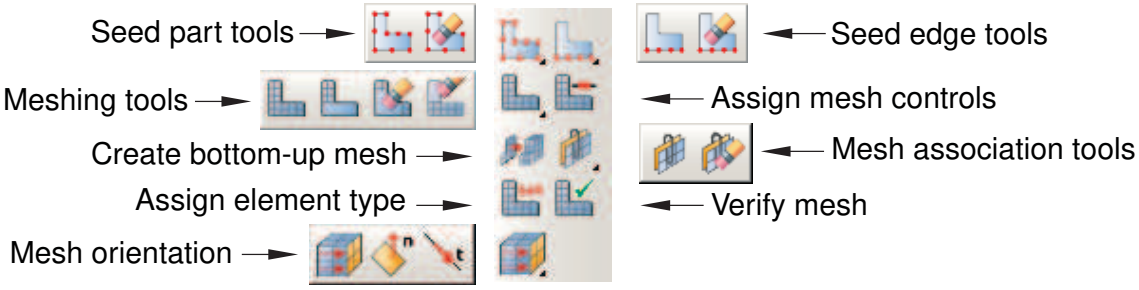


Figure 1: The Mesh module toolbox.

For information on using each of the Mesh module tools, refer to the following sections:

- *Bottom-up meshing*
- *Mesh-geometry association*
- *Seeding a model*
- *Creating and deleting meshes*
- *Controlling mesh characteristics*
- *Obtaining mesh information and statistics*

Seeding a model

This section explains how to use the seeding tools to apply seeds throughout a part instance.

In this section:

- *Defining seed density for an entire part or part instance*
- *Seeding an edge by prescribing the number of elements*
- *Seeding an edge by prescribing element size*
- *Prescribing biased seeding along an edge*
- *Applying constraints to edge seeds*
- *Seeding previously meshed parts, part instances, or regions*
- *Deleting part or instance seeds*
- *Deleting edge seeds*
- *Relaxing constraints using the error dialog box*

Defining seed density for an entire part or part instance

You can select **Seed->Part** or **Seed->Instance** from the main menu to define the approximate element size for all edges of a part or part instance that do not already have magenta-colored edge seeds. Seeds defined in this way are called part seeds or instance seeds and are colored white. (For more information, see *Controlling the seed density*.) You should apply seeds to all edges. If a uniform seed distribution is sufficient, the recommended approach is to seed the entire part or part instance.

1. From the main menu bar, select **Seed->Part** or **Seed->Instance**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also seed a part or part instance using the  tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. If you are seeding a part instance and your assembly contains more than one part instance, select the part instance to seed and click mouse button 2.

   Note:

   Edge seeds always override part and instance seeds; therefore, if you have already individually seeded all the edges of the part or part instance, the instance seeds are unused and do not appear. If necessary, use the seed deletion tool, described in *Deleting edge seeds*, to remove any unwanted edge seeds; if you have assigned part or instance seeds, these seeds automatically appear on the edges where you delete edge seeds.

3. In the **Global Seeds** dialog box that appears, enter an approximate element size.
4. By default, Abaqus/CAE applies curvature control to the seeding of your part or part instance such that small holes or regions of high curvature are properly approximated in the mesh. Curvature control allows Abaqus/CAE to calculate the seed distribution based on the curvature of the edge along with the target element size. To control the effect of curvature on seeding, enter a value for the **Maximum deviation factor**. The deviation factor is a measure of how much the element edges deviate from the original geometry. To help you visualize the influence of the deviation factor, Abaqus/CAE displays the number of elements it would create around a circle corresponding to the setting that you enter.
5. If desired, change the **Minimum size control**. Specifying a minimum size control helps prevent Abaqus/CAE from creating unnecessarily fine meshes in areas of high curvature if these areas are not important to the analysis intent. Choose one of the following options:

   - Specify the minimum as a fraction of the global element size. Abaqus/CAE uses this method with a value of `0.1` (10%) as the default minimum size.
   - Enter an absolute minimum element size. The size must be greater than `0.0` and less than the approximate global element size.

6. Click **Apply** to view the seeding that Abaqus/CAE will use, and adjust the values that you entered in the **Global Seeds** dialog box if necessary.
7. Click **OK** to commit the element size and to close the dialog box.

   White seeds appear on all edges of the part or part instance except those already assigned magenta edge seeds.

8. To exit the part or instance seeding procedure, press **[Enter]** or click mouse button 2.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Seeding an edge by prescribing the number of elements

You can define the element size along selected edges by entering the number of elements to create. Abaqus/CAE allows you to select edges, faces, or cells to seed. However, seeds are positioned only along edges—the edges that you select or the edges of faces and cells that you select.

All the edge seeding tools generate edge seeds, which are displayed in magenta. Edge seeds override any part or instance seeds you have specified. You should apply seeds to all edges.

1. From the main menu bar, select **Seed->Edges**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also seed an edge by prescribing the number of elements using the tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Select the edges, faces, or cells you want to seed.

   By default, Abaqus/CAE allows you to select only edges to seed. To select faces or cells to seed, use the **Selection** toolbar to change the type of object that you can select to **Face**, **Cells**, or **All**. For more information, see *Filtering your selection based on the type of object*.

3. When you have finished selecting edges, faces, or cells, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

4. From the **Local Seeds** dialog box that appears, choose **By number**.

   Note:

   If you selected edges that you previously seeded using a combination of seeding methods, Abaqus/CAE provides an **As Is** option that allows you to retain the seeding method on the selected edges.

5. Enter the number of elements that Abaqus should generate along each edge.
6. Choose a bias control of **None**.
7. If desired, change the default seed constraints on the **Constraints** tabbed page. See *Applying constraints to edge seeds*, for further information on setting seed constraints.
8. If desired, toggle on **Create set with name** to create a set containing the edges, faces, and cells you selected and enter the name of the set. If you subsequently want to change the seeding, you can select the set in Step 2 without having to reselect the edges, faces, and cells.
9. Click **Apply** to view the seeding that Abaqus/CAE will use.

   Magenta seeds appear on the selected edges.

10. If necessary, adjust the values that you entered in the **Local Seeds** dialog box.
11. Click **OK** to commit the element seeding and to close the dialog box.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Seeding an edge by prescribing element size

You can define the element size along selected edges by entering the approximate element size. You can also define curvature control parameters that Abaqus/CAE uses to adjust the element size in regions of high curvature. Abaqus/CAE allows you to select edges, faces, or cells to seed. However, seeds are positioned only along edges—the edges that you select or the edges of faces and cells that you select.

All the edge seeding tools generate edge seeds, which are displayed in magenta. Edge seeds override any *part or instance seeds* you have specified. You should apply seeds to all edges.

1. From the main menu bar, select **Seed->Edges**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also seed an edge by element size using the tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Select the edges, faces, or cells you want to seed.

   By default, Abaqus/CAE allows you to select only edges to seed. To select faces or cells to seed, use the **Selection** toolbar to change the type of object that you can select to **Face**, **Cells**, or **All**. For more information, see *Filtering your selection based on the type of object*.

3. When you have finished selecting edges, faces, or cells, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

4. From the **Local Seeds** dialog box that appears, choose **By size**.

   Note:

   If you selected edges that you previously seeded using a combination of seeding methods, Abaqus/CAE provides an **As Is** option that allows you to retain the seeding method on the selected edges.

5. Enter the approximate element size to be used along the selected edges.

6. Choose a bias control of **None**.

7. By default, Abaqus/CAE applies curvature control to the seeding of your part or part instance to allow for small holes or regions of high curvature that you wish to model. Curvature control allows Abaqus/CAE to calculate the seed distribution based on the curvature of the edge along with the target element size. To control the effect of curvature on seeding, do the following:

   a. Enter a value for the deviation factor. The deviation factor is a measure of how much the element edges deviate from the original geometry. To help you visualize the influence of the deviation factor, Abaqus/CAE displays the number of elements it would create around a circle corresponding to the setting that you enter.

   b. If desired, specify a minimum size factor as a fraction of the global element size. Specifying a minimum size factor prevents Abaqus/CAE from creating very fine meshes in areas of high curvature that you have no interest in modeling.

8. If desired, change the default seed constraints by clicking the **Constraints** button in the prompt area and responding to the dialog box that appears. See *Applying constraints to edge seeds*, for further information on setting seed constraints.

9. If desired, toggle on **Create set with name** to create a set containing the edges, faces, and cells you selected and enter the name of the set. If you subsequently want to change the seeding, you can select the set in Step 2 without having to reselect the edges, faces, and cells.

10. Click **Apply** to view the seeding that Abaqus/CAE will use.
    Magenta seeds appear on the selected edges.
11. If necessary, adjust the values that you entered in the **Local Seeds** dialog box.
12. Click **OK** to commit the element seeding and to close the dialog box.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Prescribing biased seeding along an edge

You can define a nonuniform distribution of elements along selected edges by defining the size of the coarsest and finest element along the edges or the number of elements and the ratio of the two sizes.
You can define a single bias that changes the mesh density from one end of the edge to the other. Alternatively, you can define a double bias that changes the mesh density from the center of the edge to each end of the edge.

For example, *Figure 1* shows a combination of edges with single- and double-bias seeding.

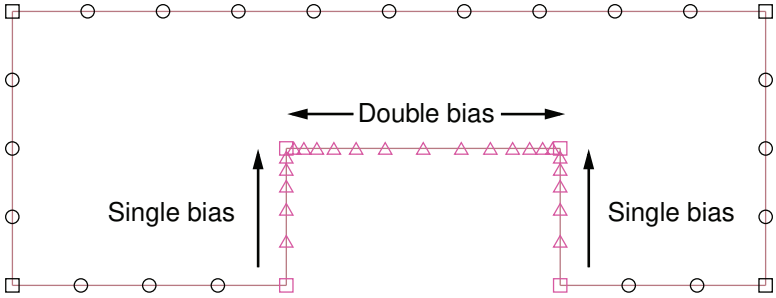


Figure 1: Single- and double-bias seeding.

All the edge seeding tools generate edge seeds, which are displayed in magenta. Edge seeds override any part or instance seeds you have specified. You should apply seeds to all edges.

1. From the main menu bar, select **Seed->Edges**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Choose the approach for picking from the viewport:

   - For single-bias seeding, toggle on **Use single-bias picking** and select the edges you want to seed. You can select only edges for single-bias seeding, and you must select each edge near the end where you expect the mesh to be denser.
   - For double-bias seeding, toggle off **Use single-bias picking** and select the edges, faces, or cells you want to seed. The location of your selection does not influence the seeding.

     By default, Abaqus/CAE allows you to select only edges to seed. To select faces or cells to seed, use the **Selection** toolbar to change the type of object that you can select to **Face**, **Cells**, or **All**. For more information, see *Filtering your selection based on the type of object*.

3. When you have finished selecting edges, faces, or cells, click **Done** in the prompt area.
4. From the **Local Seeds** dialog box that appears, choose the bias control (**Single** or **Double**).

   Note:

   If you selected edges that you previously seeded using a combination of bias seeding methods, Abaqus/CAE provides an **As Is** option that allows you to retain the bias seeding method on the selected edges.

   If you selected single-bias seeding, Abaqus/CAE displays arrows on each selected edge indicating the direction in which the element density will increase. If you selected double-bias seeding, Abaqus/CAE

displays arrows at the center of each edge indicating the directions in which the element density will increase.

5. Choose the sizing method (**By size** or **By number**).
   a. If you selected **By size**, enter the minimum and maximum size of the elements—the approximate size of the elements at each end of the biased seeding. You cannot apply curvature control to biased seeding.
   b. If you selected **By number**, enter the number of elements and the bias ratio. The bias ratio is the approximate ratio of the size of the largest and smallest element on the edge and must be greater than one.
   c. If desired, click **Select** to reverse the direction of the seeding bias. (When you select an edge to which biased seeding will be applied, the element density increases toward the end of the edge that is closest to the location of your pick.)

      Abaqus/CAE reverses the direction of the arrow on each selected edge.



Note:

If you selected edges that you previously seeded using a combination of element size or number parameters, Abaqus/CAE provides an **As Is** option that allows you to retain the size parameters on the selected edges.

6. If desired, change the default seed constraints by clicking the **Constraints** button in the prompt area and responding to the dialog box that appears. See *Applying constraints to edge seeds*, for further information on setting seed constraints.
7. If desired, toggle on **Create set with name** to create a set containing the edges, faces, and cells you selected and enter the name of the set. If you subsequently want to change the seeding, you can select the set in Step 2 without having to reselect the edges, faces, and cells.
8. Click **Apply** to view the seeding that Abaqus/CAE will use.

   Magenta seeds appear on the selected edges.
9. If necessary, adjust the values that you entered in the **Local Seeds** dialog box.
10. Click **OK** to commit the element seeding and to close the dialog box.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Applying constraints to edge seeds

You can apply constraints to seeds at the same time you define them by using the **Constraints** tabbed page of the **Local Seeds** dialog box that appears during the edge seeding procedures.



Warning: Use care when constraining mesh seeds; overconstraint can make automatic mesh generation impossible if you are trying to generate a mesh that contains quadrilateral- or hexahedral-shaped elements.

1. From the main menu bar, select **Seed->Edges**. Then select the edges that you want to seed, and click mouse button 2. For more information on performing these tasks, see the following sections:
   - *Seeding an edge by prescribing the number of elements*
   - *Seeding an edge by prescribing element size*
   - *Prescribing biased seeding along an edge*

   Note:

   If you selected edges that you previously seeded using a combination of constraints, Abaqus/CAE provides an **As Is** option that allows you to retain the constraints on the selected edges.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click the **Constraints** tabbed page of the **Local Seeds** dialog box.
3. Choose the desired constraint. The choices are:

Allow the number of elements to increase or decrease

This option leaves the seeds totally unconstrained. Therefore, the number of elements that Abaqus/CAE creates along an edge can be greater or smaller than the number of elements dictated by the seeds; however, the mesh generator tries to adhere to the requested seed pattern as much as possible. This option provides the greatest flexibility to the automatic mesh generator and, therefore, gives the greatest chance of success when generating the mesh.

Allow the number of elements to increase only

This option (default) partially constrains the seeds so that the number of elements along an edge can only be greater than or remain the same as that dictated by the seeds. Unconstrained seeds that are located near partially constrained seeds tend to behave as though they are partially constrained.

Do not allow the number of elements to change

This option fully constrains the seeds so that the exact number of elements that you have specified with your seeds is maintained.

Use this option with care, since adjustment of the number of elements along the boundary of a region is usually required to generate a mesh; preventing such adjustments can make mesh generation impossible.



Note:

The constraint options described above allow you to control only the number of elements along an edge, not the positions of the nodes along the edge. Abaqus/CAE creates meshes that adhere to the seeds as closely as possible.

4. Click **OK**.

   The mesh seeds change shape to indicate the constraint you have chosen:

   - Circle: unconstrained seed.
   - Triangle: partially constrained seed. The triangle points upward, indicating that the number of elements along the edge can only increase.
   - Square: fully constrained seed.

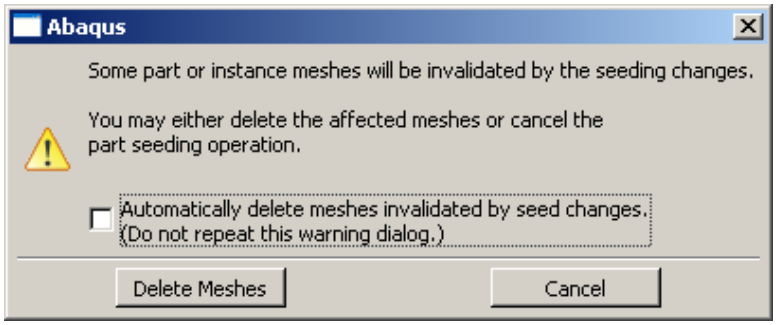
Additional information

- *Seeding a model*
- *Understanding seeding*
- *Selecting objects within the viewport*

Seeding previously meshed parts, part instances, or regions

To reseed a part, a part instance, or a region that has already been meshed, you must first delete the mesh in one of the following ways:

- Before seeding, use the mesh deletion tools, described in *Deleting a mesh*, to delete the mesh from the part, part instance, or region of interest.
- Do not delete the mesh yourself; instead, attempt to seed the part, part instance, or region of interest. As soon as you select the parts or edges to seed, Abaqus/CAE displays a window similar to the following:



You can delete the mesh by clicking **Delete Meshes**, or you can keep your mesh and exit the seeding procedure by pressing **Cancel**.

You can avoid this warning message for the remainder of the current session by toggling on **Automatically delete meshes invalidated by seed changes**. The next time you attempt to seed a part, part instance, or region that already contains a mesh, the mesh will be deleted immediately without any warning being displayed.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Selecting objects within the viewport*

Deleting part or instance seeds

Delete part or instance seeds by selecting **Seed->Delete Part Seeds** or **Seed->Delete Instance Seeds** from the main menu bar.

1. From the main menu bar, select **Seed->Delete Part Seeds** or **Seed->Delete Instance Seeds**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also delete seeds using the  tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Do one of the following:

   - If you are deleting part seeds, click **Yes** in the prompt area to confirm the deletion.
   - If your assembly contains only one part instance, click **Yes** in the prompt area to confirm the deletion.
   - If your assembly contains more than one part instance, select the instances whose seeds you want to delete and click mouse button 2 to commit your selection.

   Abaqus/CAE deletes the seeds.

Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Deleting edge seeds

Delete edge seeds by selecting **Seed->Delete Edge Seeds** from the main menu bar and then selecting the edges, faces, or cells from which to delete edge seeds.

1. From the main menu bar, select **Seed->Delete Edge Seeds**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also delete seeds using the  tool, located with the seed tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Select the edges whose edge seeds you want to delete.

   By default, Abaqus/CAE allows you to select from all items to delete edge seeds. To limit object selection to only edges, faces, or cells, use the **Selection** toolbar to change the type of object that you can select to **Edges**, **Faces**, or **Cells**. For more information, see *Filtering your selection based on the type of object*.

3. When you have finished selecting edges, faces, or cells, press mouse button 2 to commit your selections. For more information on selecting objects, see *Selecting objects within the viewport*."

   The seeds disappear from all edges of the selected objects.

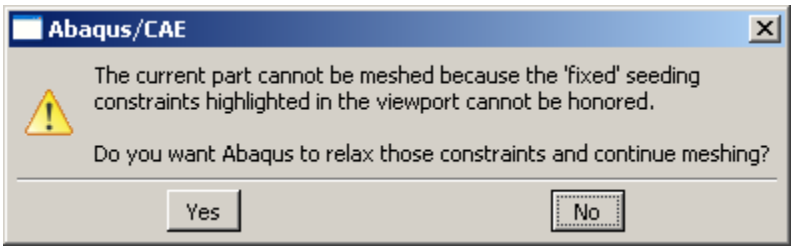
Additional information

- *Seeding a model*
- *Understanding seeding*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Relaxing constraints using the error dialog box

Sometimes mesh generation fails if seeds are overconstrained. Several options are available to handle the overconstraints.

When Abaqus/CAE is unable to create a mesh because of overconstrained seeds, the following dialog box appears:



In addition, the overconstrained seeds are highlighted in the viewport. You can choose one of the following options:

- Click **Yes** to relax the seed constraints and to continue meshing the region.
- Click **No** to save the seed constraints and to cancel the meshing procedure.

Additional information

• *Applying constraints to edge seeds*

Creating and deleting meshes

This section explains how to use the mesh tools to create and delete meshes throughout a part instance or region.

In this section:

- *Creating a mesh*
- *Deleting a mesh*

Creating a mesh

You can create meshes throughout entire parts or part instances or just within selected regions.
To create a mesh, select either **Mesh->Part**, **Mesh->Instance**, or **Mesh->Region** from the main menu bar.

1. From the **Object** field in the context bar, select the part to mesh or select the assembly.
2. From the main menu bar, select **Mesh->Part**, **Mesh->Instance**, or **Mesh->Region**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also mesh parts, part instances, or regions using the  and  tools, located with the mesh tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

3. If you are meshing the assembly, you must select the instances or regions to mesh. Click mouse button 2 to indicate that you have finished selecting. (For more information on selecting objects, see *Selecting objects within the viewport*.)

   Select only those part instances or regions that are colored green, pink, or yellow, indicating they are meshable. To make orange regions meshable, you must either subdivide them using the partitioning tools, assign the bottom-up meshing technique, or assign tetrahedral-shaped elements to them. Regions that you have already assigned the bottom-up meshing technique are colored light tan; you must either mesh them using the bottom-up technique or assign another meshing technique. (For more information on creating bottom-up meshes, see *Bottom-up meshing*.) Parts colored white cannot be meshed, because they are associated with independent instances.

   Note:

   Part instances are colored according to their meshability only when the **Mesh defaults** color mapping is selected. Apply this color mapping if the colors in your viewport do not match those described in this step.

4. Do one of the following:

   - From the prompt area, click **Yes** to generate the mesh on those regions that are meshable.
   - If the assembly includes a solid region that will be meshed with tetrahedral elements, Abaqus/CAE asks if you want to preview the triangular mesh on the exterior faces of the region.

     1. To preview the mesh, toggle on **Preview boundary mesh** from the prompt area and click **Yes** to create the boundary mesh.

        Abaqus/CAE generates the triangular boundary mesh on the mesh on those regions that are meshable.

     2. If some regions fail to mesh or if the boundary mesh is not acceptable, Abaqus/CAE provides a variety of tools that will help you generate a mesh in all regions and create an acceptable mesh. For more information, see *What can I do with a boundary mesh?*.

     3. If the boundary mesh is acceptable, click **Yes** from the prompt area to continue meshing the interior of the parts, instances, or regions.

Abaqus/CAE generates the mesh on those regions that are meshable.

Additional information

- *Understanding mesh generation*
- *Verifying and improving meshes*
- *Advanced meshing techniques*
- *Using the prompt area during procedures*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Deleting a mesh

You can delete Abaqus native meshes throughout entire parts, part instances, or just within selected regions. To delete a mesh, select either **Mesh->Delete Part Native Mesh**, **Mesh->Delete Instance Native Mesh**, or **Mesh->Delete Region Native Mesh** from the main menu bar.



Note: Deleting a mesh does not cause the underlying seeds to be deleted, so you can modify the seeding pattern and regenerate the mesh.

1. From the main menu bar, select either **Mesh->Delete Part Native Mesh**, **Mesh->Delete Instance Native Mesh**, or **Mesh->Delete Region Native Mesh**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also delete meshes using the  or  tools, located with the mesh tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Do one of the following:

   - If you are deleting a part mesh, click **Yes** from the prompt area to confirm that you want to delete the mesh.
   - If you are deleting region meshes from a part or an assembly, select the regions from which to delete the mesh. Click mouse button 2 to indicate that you have finished selecting. (For more information on selecting objects, see *Selecting objects within the viewport*.)
   - If you are deleting instance meshes from the assembly, select the instances from which to delete the mesh. Click mouse button 2 to indicate that you have finished selecting. (For more information on selecting objects, see *Selecting objects within the viewport*.)

   Abaqus/CAE deletes the mesh.

Additional information

- *Understanding seeding*
- *Using the prompt area during procedures*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Controlling mesh characteristics

This section explains how you control the overall characteristics of your mesh.

In this section:

- *Assigning mesh controls*
- *Choosing an element shape*
- *Selecting a meshing technique*
- *Redefining region corners*
- *Setting the mesh algorithm*
- *Specifying the sweep path*
- *Sweep meshing a solid, revolved region whose profile touches the axis of revolution*
- *Applying a mesh stack orientation*
- *Changing mesh controls for previously meshed regions*
- *Associating Abaqus elements with mesh regions*
- *Changing the labels of all nodes and elements*
- *Adding layers of wedge elements to tetrahedral mesh boundaries*

Assigning mesh controls

The **Mesh Controls** dialog box allows you to specify the shape of the elements in a mesh as well as the meshing technique that Abaqus/CAE uses to create the mesh. In some cases, you can also select transition options and redefine region corners.

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the  tool, located in the Mesh module toolbox.

2. If your part or assembly contains more than one region, select the regions whose mesh controls you want to view or modify and then press mouse button 2. All the selected regions must have the same dimensionality.

   Note:

   To view or modify the mesh controls for faces of a region to which the free meshing technique and tetrahedral element shape are assigned or for faces of a bottom-up region, you must change the entity selection type in the prompt area to **faces of solid regions**.

   The **Mesh Controls** dialog box appears.

3. Select the mesh controls of your choice. For information on specific mesh controls, see the following:

   - *Choosing an element shape*
   - *Selecting a meshing technique*
   - *Redefining region corners*
   - *Setting the mesh algorithm*
   - *Adding layers of wedge elements to tetrahedral mesh boundaries*

4. If desired, click **Defaults** to change the settings in the **Mesh Controls** dialog box back to the default values.
5. Click **OK** to save your settings and to close the **Mesh Controls** dialog box.

Additional information

- *Assigning Abaqus element types*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Choosing an element shape

You can control the shape of the elements in your mesh by selecting **Mesh->Controls** from the main menu bar. The **Element Shape** options are located at the top of the **Mesh Controls** dialog box that appears.

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can set the element shape using the tool, located in the Mesh module toolbox.

2. If your part or assembly contains more than one region, select those regions whose element shapes you want to view or modify and then press mouse button 2. All the selected regions must have the same dimensionality.

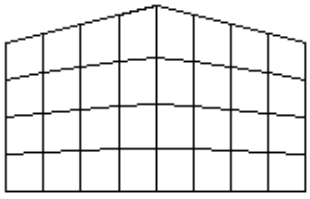
   The **Mesh Controls** dialog box appears.

3. From the list of **Element Shape** options, select the element shape of your choice.

   If you selected a two-dimensional region, you can choose from the following element shape options:

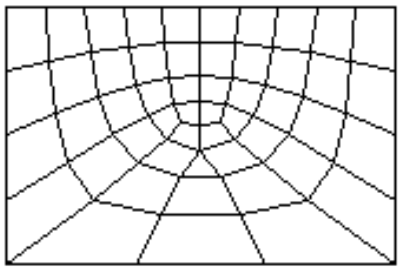
Quad

Use exclusively quadrilateral elements. The following figure shows an example of a mesh that was constructed using this setting:



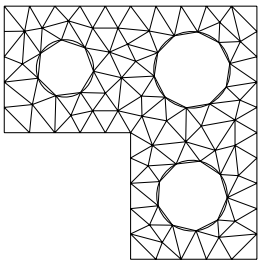
Quad-dominated

Use primarily quadrilateral elements, but allow triangles in transition regions. This setting is the default. The following figure shows an example of a mesh that was constructed using this setting:



Tri

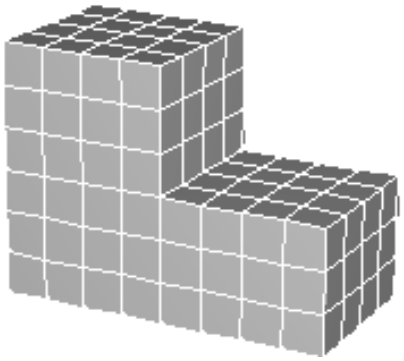
Use exclusively triangular elements. This setting is the only one available when you apply mesh controls to faces of solid regions since the triangular face mesh will be used to generate a tetrahedral solid mesh. The following figure shows an example of a mesh that was constructed using this setting:



If you selected a three-dimensional region, you can choose from the following element shape options:

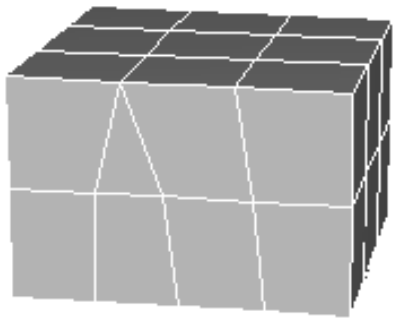
Hex

Use exclusively hexahedral elements. This setting is the default. The following figure shows an example of a mesh that was constructed using this setting:



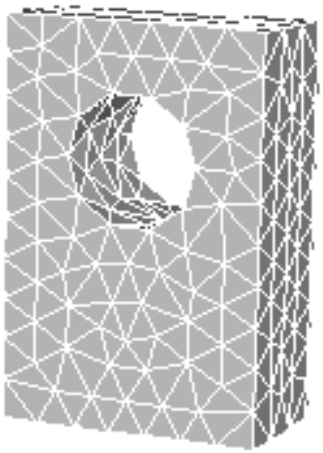
Hex-dominated

Use primarily hexahedral elements, but allow some triangular prisms (wedges) in transition regions. The following figure shows an example of a mesh that was constructed using this setting:



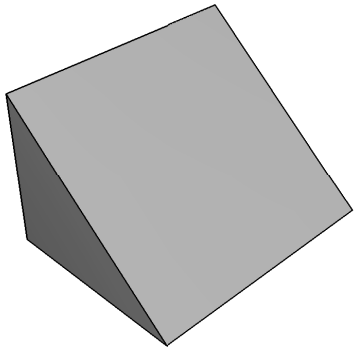
Tet

Use exclusively tetrahedral elements. The following figure shows an example of a mesh that was constructed using this setting:



Wedge

Use exclusively wedge elements. The following figure shows an example of a single-element mesh that was constructed using this setting:



4. Click **OK**.

   The next time you generate a mesh on the selected regions, your selections will be honored.

   If the selected regions are already meshed, you will be prompted to delete the mesh or to cancel the mesh control procedure.

Additional information

- *Controlling mesh characteristics*
- *Understanding mesh generation*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Selecting a meshing technique

Abaqus/CAE assigns a default top-down meshing technique to each meshable region of your model based on the geometry of the region and the current element shape selection for that region. Abaqus/CAE uses the mesh technique assigned to a region to generate a mesh for the region. You can use the **Mesh Controls** dialog box to select an alternate meshing technique.

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also set the meshing technique using the  tool, located in the Mesh module toolbox.

2. Select **faces of solid regions** to assign mesh controls to boundary faces of solid regions that will be tetrahedral meshed or bottom-up meshed.
3. If your part or assembly contains more than one region, select those regions whose meshing technique you want to view or modify and press mouse button 2. The selected regions must all have the same dimensionality.

   The **Mesh Controls** dialog box appears.
4. From the list of **Technique** options, select the meshing technique of your choice. (Some techniques are available only if they are valid for the selected region.)

   - Abaqus/CAE selects **As is** if, in the previous step, you selected multiple regions that have different meshing techniques already assigned to them.
   - Select ***Free*** to create a free mesh.
   - Select ***Structured*** to create a structured mesh.
   - Select ***Sweep*** to create a swept mesh.
   - Select ***Bottom-up*** to create a bottom-up mesh.
   - Abaqus/CAE selects **Multiple** if a change in element shape assignment results in multiple techniques being assigned automatically to the selected regions. For example, suppose the free meshing technique is applied to all the regions of a solid part instance. If you change the element shape assignment of these regions from **Tet** to **Hex** or **Hex-dominated**, Abaqus/CAE automatically changes the meshing technique assigned to each region from the free meshing technique to whatever technique is appropriate for each region; for example, structured meshing for some regions and swept meshing for others.

     Note:

     The bottom-up meshing technique must be manually assigned to or removed from a region. When you modify the geometry of a region to which the bottom-up meshing technique is assigned, the resulting region or regions will also be assigned the bottom-up meshing technique. You can click **Defaults** in the **Mesh Controls** dialog box to assign a top-down technique to a bottom-up region.

   For detailed information on each meshing technique, see *Understanding mesh generation*.

   While assigning mesh controls to faces of a region, Abaqus/CAE color codes the faces according to the mesh technique that will be used on the faces. The face colors may not be the same as the color for the region. For example, the faces of a bottom-up region will appear pink by default since they are free meshed. If you assign the structured mesh technique to some faces, they will be colored green.

The solid region color for a bottom-up region is light tan, and the color for a solid region that will be tetrahedral meshed is pink.

5. Click **OK** to close the dialog box and to save your mesh technique selection.

   The next time you generate a mesh on the selected part instance or region, your selections will be honored.

   If the selected region already contains a mesh, you will be prompted to delete the mesh or to cancel the mesh control procedure.

Additional information

- *Controlling mesh characteristics*
- *Understanding mesh generation*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Redefining region corners

Structured meshing patterns exist for regions with particular topologies. For example, Abaqus/CAE applies a particular structured pattern to quadrilateral regions and another pattern to pentagonal regions. In some cases, however, you can change the structured pattern assigned to a surface region by redefining the region's corners. You can redefine region corners only for surface regions to which the structured meshing technique has been assigned.

If you click **Redefine Region Corners** in the **Mesh Controls** dialog box, you can select which corners of the region you want Abaqus/CAE to consider when creating the mesh. If you leave a corner unselected, Abaqus/CAE internally "combines" the edges on either side of the unselected corner into a single logical edge (though the actual topology of the region remains unchanged). For example, if you leave one corner of a pentagonal region unselected, Abaqus/CAE considers that region to have only four edges instead of five. As a result, the structured meshing pattern for quadrilateral regions will be applied to the region rather than the pattern for pentagonal regions. For more information, see *Two-dimensional structured meshing*.

A surface region can be meshed using the structured meshing technique only if the region is bounded by three to five logical sides. However, if the region contains virtual topology, Abaqus/CAE can apply structured meshing only if the region has four corners. As a result, to redefine the corners of a region with virtual topology, the region must be bounded by more than four corners, and you must select four of the existing corners.

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   
   Tip: You can also click the  tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. If your part or assembly contains more than one region, select those regions whose corners you want to redefine and then press mouse button 2. The regions that you select should have three or more vertices.

   The **Mesh Controls** dialog box appears.

3. Select **Structured** as the meshing technique if it is not already selected.

   The **Redefine Region Corners** button appears on the right side of the **Mesh Controls** dialog box.

4. Click **Redefine Region Corners**.

   If you selected multiple regions, this procedure considers each of the selected regions in turn. (Abaqus/CAE skips selected regions that a structured pattern cannot be applied to or that contain three or fewer vertices.) The region currently being considered becomes highlighted in magenta. The currently selected corners for the region are highlighted in yellow.

5. In the prompt area, select an option for determining region corners.

   - If you click **Accept Highlighted**, Abaqus/CAE accepts the currently highlighted corners. If multiple regions are selected, options are presented for the next region. If only one region is selected, the procedure is complete, and the **Mesh Controls** dialog box reappears.
   - If you click **Select New**, the currently selected vertices turn red. You must go on to the next step.
   - If you click **Revert to Defaults**, the default corners of the region are highlighted. You are prompted to select either **Accept Highlighted** or **Select New**, described above.

6. If you clicked **Select New** in the previous step, select the vertices of the region that you want as region corners. You can select between three and five vertices.

   - **[Shift] + Click** to select a vertex without unselecting all the other vertices.

- **[Ctrl] + Click** to unselect an individual vertex without unselecting all the other vertices.
- When you have finished selecting vertices, click mouse button 2.

Selected vertices are red, and unselected vertices are yellow.

If multiple regions are selected, the procedure starts over with the next region. If only one region is selected, the procedure is complete and the **Mesh Controls** dialog box reappears.

Additional information

- *Controlling mesh characteristics*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Setting the mesh algorithm

The mesh algorithm options that are available depend on the element shape and the meshing technique that you have selected. If the mesh algorithm option is applicable to the type of mesh you are creating, an **Algorithm** field appears on the right side of the **Mesh Controls** dialog box.

Abaqus/CAE provides the following mesh algorithm options:

Choose the mesh algorithm

Choose either **Medial axis** or **Advancing front**. It is difficult to predict which algorithm will produce a better mesh for a particular region; you may have to experiment with the two algorithm settings. For more information, see *What is the difference between the medial axis algorithm and the advancing front algorithm?*.

Minimize the mesh transition

You can control whether Abaqus/CAE will try to minimize the mesh transition when it moves from a coarse mesh to a fine mesh. In most cases, toggling on **Minimize the mesh transition** will reduce mesh distortion. However, if you toggle off **Minimize the mesh transition**, the mesh may move closer to the specified mesh seeds. For more information, see *What is a mesh transition?*.

Use mapped meshing where appropriate

Some models that appear very complex actually contain faces with relatively simple geometry. By default, Abaqus/CAE uses the mapped meshing technique where appropriate to generate elements on simple faces. You can toggle off **Use mapped meshing where appropriate** to prevent Abaqus/CAE from using mapped meshing. However, if you mesh a model with simple faces and do not allow mapped meshing, the resulting element quality can be poor on the simple faces. For more information, see *What is mapped meshing?*, and *When can Abaqus/CAE apply mapped meshing?*.

Insert boundary layer

When you are creating a free mesh of tetrahedral elements, you can add a boundary layer of wedge elements along the faces of region boundaries. The boundary layer is composed of a series of wedge elements stacked normal to the solid wall of the region with the thinnest elements against the wall. Boundary layers create a mesh with a high density at the wall and decreasing density as it progresses toward the tetrahedral mesh in the interior of the region. For detailed instructions on creating a boundary layer, see *Adding layers of wedge elements to tetrahedral mesh boundaries*.

Use the default algorithm

When you are creating a free mesh of tetrahedral elements, you can choose the default mesh generation algorithm or the algorithm that was included with Abaqus/CAE 6.4 and earlier. In most cases the default algorithm is more robust, particularly when meshing complex shapes and thin solids. For more information, see *Free meshing with triangular and tetrahedral elements*.

Increase the size of the interior elements

If you choose the default mesh generation algorithm to create a free mesh of tetrahedral elements, you can toggle on **Non-standard interior element growth** and use either the slider control or the text field to specify a growth rate for interior elements. The growth rate must be between `1.0` (no or minimal growth) and `2.0` (maximum growth).

If the mesh density is adequate for the model being analyzed and the areas of interest are on the mesh boundary, increasing the size of the interior elements will increase the computational efficiency. To view the internal elements generated by Abaqus/CAE, you can use a view cut or use display groups to remove exterior elements from the view.

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the  tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. If your part or assembly contains more than one region, select the regions of interest and click mouse button 2.

   The **Mesh Controls** dialog box appears. If the mesh algorithm option is applicable to the selected element shape and meshing technique, an **Algorithm** field appears on the right side of the **Mesh Controls** dialog box.

3. Select the desired algorithm options, and click **OK** to save your data and to close the dialog box.

Additional information

- *Free meshing*
- *Controlling mesh characteristics*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Specifying the sweep path

If you apply the swept meshing technique to a particular region, a **Redefine sweep path** button appears in the **Mesh Controls** dialog box for that region. If the region has more than one valid sweep path, you can click **Redefine sweep path** to select the path of your choice. (For more information on sweep paths, see *Swept meshing*.)

This option is of particular interest if you are assigning mesh controls to a gasket, continuum shell, cylindrical, or cohesive region. Unless you assign another orientation (for more information, see *Applying a mesh stack orientation*), the directional properties of these region types are dependent on the sweep path. When you mesh a gasket region, the axis of each gasket element will be coincident with the sweep path direction. (For more information, see *Gaskets*," and *About Gasket Elements*.) When you mesh a continuum shell or cohesive region, the sweep path is aligned with the stack direction. (For more information, see *Meshing parts with continuum shell elements*, and *Creating a model with cohesive elements using geometry and mesh tools*.) When you mesh a cylindrical region with cylindrical elements, the sweep path of the selected sweep/revolve region is aligned along the circumferential direction of the cylindrical geometry. (For more information, see *Swept meshing of cylindrical solids*.)

1. From the main menu bar, select **Mesh->Controls**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the  tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. If your part or assembly contains more than one region, select the sweep-meshable regions of interest and then press mouse button 2.

   The **Mesh Controls** dialog box appears.

3. In the **Mesh Controls** dialog box, select **Sweep** as the meshing technique if it is not already selected.

   If more than one valid sweep path exists for the regions that you selected, a **Redefine Sweep Path** button appears toward the bottom of the dialog box.

4. Click **Redefine Sweep Path**.

   If you selected multiple regions, this procedure considers each of the selected regions in turn. The region currently being considered becomes highlighted in magenta, and the default sweep path is highlighted in red with an arrow indicating the sweep direction.

5. Specify the sweep path and direction of your choice by selecting the appropriate option in the prompt area:

   - Click **Accept Highlighted** to accept the sweep path highlighted in the viewport.
   - Click **Flip** to change the direction of the currently selected sweep path. Then click **Yes** to indicate that the new sweep path direction is correct.
   - Click **Select New** (if applicable) to select a different edge as the sweep path. Then perform the following steps:

     1. Select an edge in the sweep path. You can indicate the desired sweep direction by clicking the mouse button toward the end of the edge that you want to coincide with the end of the sweep path.

        The new sweep path is highlighted in red with an arrow indicating the sweep direction.

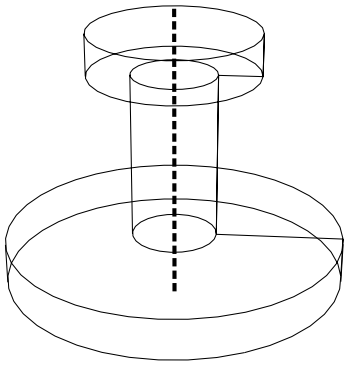
     2. Click **Yes** in the prompt area if the path and direction are correct, or click **Flip** if you want to change the direction of the sweep path. Then click **Yes** to indicate that the new sweep path direction is correct.

If multiple regions are selected with more than one valid sweep path, the procedure starts over with the next region. If only one region is selected, the procedure is complete and the **Mesh Controls** dialog box reappears.

Sweep meshing a solid, revolved region whose profile touches the axis of revolution

You can use the partitioning technique to make a part swept meshable.

In most cases, unless you create strategically placed partitions, you cannot sweep mesh a revolved solid region with all hexahedral elements if the profile that was revolved to create the region touches the axis of revolution. For example, the cylindrical part in the following figure cannot be swept meshed.



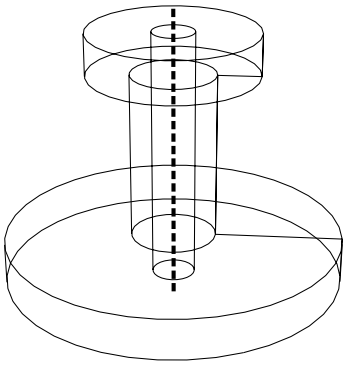
The partitioning technique to make the part swept meshable involves dividing the part or part instance into the following two regions:

- A cylindrical core region that can be meshed using the extruded swept meshing technique.
- An outer region that can be meshed using the revolved swept meshing technique.

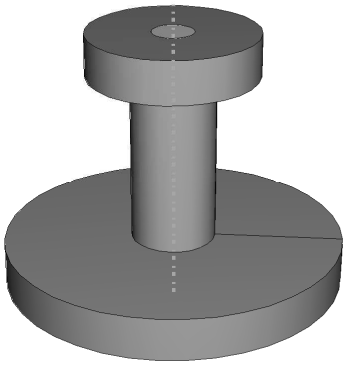
(For detailed information on sweep meshing solid regions, see *Swept meshing of three-dimensional solids*.)

Tip: The following instructions refer to color cues that appear only when the **Mesh defaults** color mapping is selected. Apply this color mapping to your viewport before attempting to partition a solid, revolved region for sweep meshing.

1. Use the Partition toolset to create a cylindrical core at the center of the region. (For detailed information on partitions, see *The Partition toolset*.) The partitions creating the cylindrical core are outlined in red in the figure below.



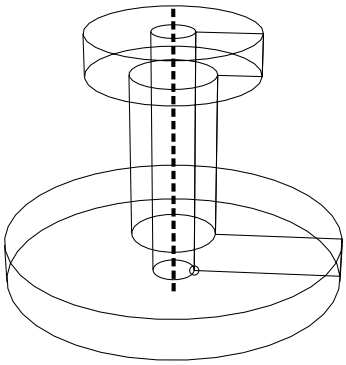
The cylindrical core is meshable using the extruded swept meshing technique and therefore becomes yellow. The outer region remains orange because it is still unmeshable.



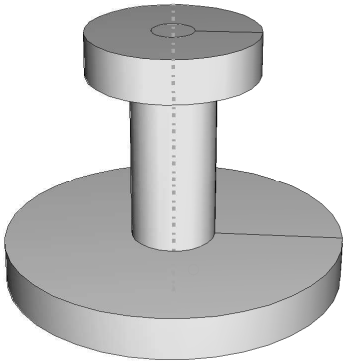
In the next step you will create additional partitions that allow Abaqus/CAE to recognize the outer region as a revolved solid whose profile does not touch the axis of revolution.

2. Use the Partition toolset to create whatever partitions are necessary to outline a revolution profile for the outer region. The profile will serve as the source side of the revolved mesh and will be swept along the circular edge defined by the cylindrical core to create the solid mesh.

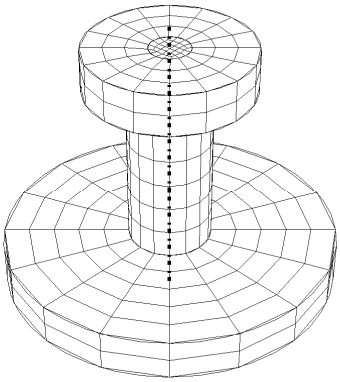
For example, partitions are used to outline the profile (shown in red) of the outer region in the figure below.



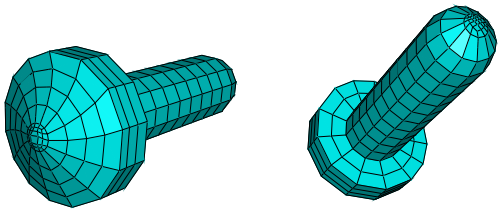
Once the profile has been defined, both the cylindrical inner region and the outer region are colored yellow and are ready for sweep meshing.



The resulting mesh is shown below.



You can apply a similar technique to mesh revolved regions even though the source side and the target side are not planar, as shown below.



Applying a mesh stack orientation

The mesh stack orientation is important when meshing parts with continuum shell, cohesive, cylindrical, or gasket elements because these elements have a distinctive directional behavior. If you use sweep meshing, the default stack orientation follows the direction of the sweep path, and you may be able to change the stack orientation by picking a different sweep path. However, you must assign a sweep direction to each cell in a part separately—a time-consuming process—and the available sweep directions may not include your desired stack direction.

The stack orientation tool allows you to assign a stack orientation to all cells or to selected cells within a solid part or part instance, as long as the cells do not have tetrahedral elements assigned. You can use the tool to apply a stack orientation quickly to a group of cells in a single operation. You assign the stack orientation based on a selected face; the stack is oriented such that the selected face is the top of the stack. Abaqus/CAE applies the orientation regardless of the mesh technique used for the cells; so the sweep direction, if one exists, does not matter.

The stack orientation tool was used to assign an orientation through the thickness of the part in *Figure 1*, as shown by the red arrow; the top faces of the elements are colored brown. The complex profile of the swept solid part did not allow the desired stack orientation using the sweep path—the two available sweep directions are indicated by the black arrow in the figure.

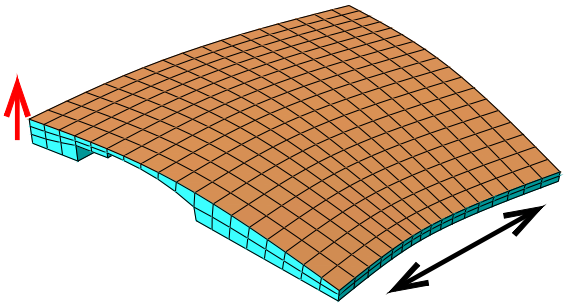


Figure 1: The stack orientation need not follow the sweep direction.

Application of a new stack orientation may vary depending on the cells selected. *Figure 2* illustrates a case where the stack direction changes as more cells are meshed. The upper left face was used for the reference orientation, so the mesh orientations for the two disconnected cells in the center image match this orientation. However, when the center cell is meshed, the orientation of the elements in the lower right cell is updated to align with the other two cells, as shown in the image on the right.

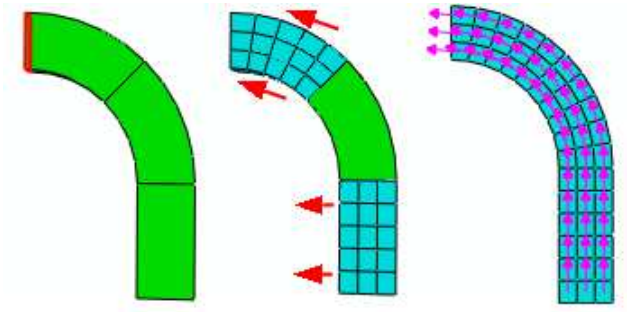


Figure 2: Stack orientation may change if you add cells to a mesh.

1. From the main menu bar, select **Mesh->Orientation->Stack**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

Tip: You can also set a mesh stack direction using the  tool, located in the Mesh module toolbox (for more information, see *Using the Mesh module toolbox*.), and the **Assign stack direction** button, located in the **Mesh Controls** dialog box.

2. If your part contains multiple cells, select the cells for which you want to assign a stack direction.
3. Select a top face to define the reference orientation.

   Abaqus/CAE prompts you to confirm your selections before assigning the orientation.
4. Click **Yes** to accept the selected face and corresponding orientation, or click **No** to return to Step 3.

   When you accept the reference face, Abaqus/CAE applies the stack direction to the selected cells.

   If any of the selected cells contains cylindrical elements, Abaqus/CAE displays a warning message indicating that the cylindrical elements must be deleted to apply a new stack direction. Click **Yes** to delete the cylindrical mesh or **No** to cancel the stack direction procedure.

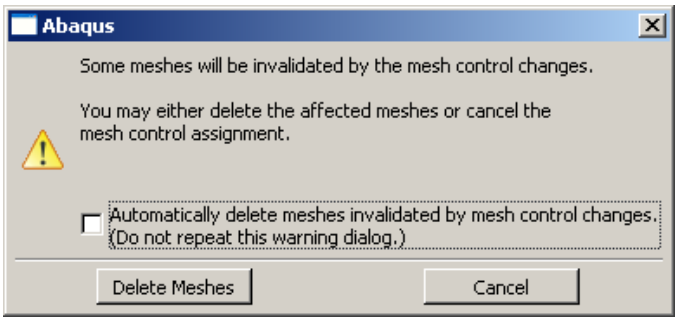
Additional information

- *Controlling mesh characteristics*

Changing mesh controls for previously meshed regions

If you change any of the mesh controls assigned to a previously meshed region, that region's mesh may become invalid.

In these cases, the following warning dialog appears when you click **OK** in the **Mesh Controls** dialog box.



You can delete the mesh by clicking **Delete Meshes**, or you can keep your mesh and cancel the new settings in the **Mesh Controls** dialog box by clicking **Cancel**.

You can also avoid this warning message for the remainder of the current session by toggling on **Automatically delete meshes invalidated by mesh control changes**. The next time you attempt to change the controls assigned to a region that already contains a mesh, the mesh will be deleted immediately without the appearance of the warning dialog box.

Additional information

• *Controlling mesh characteristics*

Associating Abaqus elements with mesh regions

To associate particular Abaqus elements with mesh regions or with orphan elements, select **Mesh->Element Type** from the main menu bar.
Then select those regions whose element type you want to assign, and make the assignment using the **Element Type** dialog box that appears.

You can use the dialog box to specify Abaqus element settings for all the element shapes that could conceivably appear in the regions you select, even if the regions currently contain only a few different element shapes. For example, even though a selected region may contain only quadrilateral elements, you can associate an element type with other shapes, too, such as triangles.

For information on populating the **Element Type** dialog box with preferred element types, see *Preferred element types list*.

The element type setting behaves like a feature. For example, if you assign element types to a region and then later partition that region into several more regions, the new regions will inherit the element type settings of the original parent region.

1. From the main menu bar, select **Mesh->Element Type**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also choose element types using the 

   tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. If your assembly or part contains both orphan elements and geometric regions, choose **geometry** in the prompt area to assign an element type to the geometry.

   Orphan elements in an assembly belong to dependent part instances; you cannot assign an element type to a dependent part instance. To assign an element type to orphan elements, you must select the mesh part from the parts list in the **Object** field of the context bar and assign the element type to the desired elements of the part.

3. If you are selecting from multiple geometric regions or if you are selecting elements from a part imported from an output database, use the following selection techniques:

Geometry

Use the mouse to select the desired regions in the viewport, and then click mouse button 2 when your selection is complete.

You can select only regions from parts of the same type; for example, you cannot select both a rigid surface and a deformable body. Likewise, the regions that you choose must have the same dimensionality.

Orphan mesh

Use the mouse to select the desired orphan elements, and then click mouse button 2 when your selection is complete.

Alternatively, you can click **Sets** on the right side of the prompt area. A dialog box appears with a list of all of the elements sets associated with the meshed part. Select the element set of your choice and then click **Continue**. For information on creating sets, see *The Set and Surface toolsets*.

All elements that you select, regardless of the selection method, must be of the same order. In addition, the elements must belong to parts of the same type.

The **Element Type** dialog box appears.

4. In the upper left corner of the dialog box, select the **Element Library** option of your choice.

   Select **Standard** to choose from the list of Abaqus/Standard elements, or select **Explicit** to choose from the list of Abaqus/Explicit elements.

5. Select the **Geometric Order** of your choice: **Linear** (first order) or **Quadratic** (second order).
6. From the **Family** list on the right side of the dialog box, select an appropriate element family for the type of analysis you will perform on the model. For example, if you plan to do a heat transfer analysis, select the **Heat Transfer** family.

   The name of the default element for the element library, geometric order, and family that you specified appears in the lower half of the dialog box.

   Note:

   You can set the element type corresponding to only a single family. For example, you cannot set the element type for linear triangles in both the plane strain and heat transfer families; you must select either heat transfer or plane strain. If, after setting element types for one family, you switch to another family, the settings for the first family are lost.

7. Choose the Abaqus element type of your choice for each element shape.
   a. Click the tab corresponding to the element shape of interest.
   b. Select the element characteristics of your choice. For more information on the element control options, see *Section Controls*.

      The name of the Abaqus element that meets all your criteria appears at the bottom of the tabbed page with a brief description.

8. Click **OK** to commit your element type assignments, or click **Defaults** and then **OK** to return all element settings to their default values.

   The element types are changed according to your specifications.

9. To set element types for additional regions, repeat this procedure starting from Step 2.

Additional information

- *How do mesh elements correspond to Abaqus elements?*
- *What kinds of elements must be generated outside the Mesh module?*
- *Element type assignment*
- *Importing parts*
- *Using the prompt area during procedures*
- *Using the Mesh module toolbox*
- *Selecting objects within the viewport*

Changing the labels of all nodes and elements

Select **Mesh->Global Numbering Control** from the main menu to change the node and element labels of a native region or of selected native part instances in the assembly. Enter the start labels, and Abaqus/CAE changes the nodes and elements labels while preserving the original order and incrementation. You can change the labels before or after Abaqus/CAE generates the mesh. (For information on renumbering orphan nodes, see *Renumbering nodes* and *Renumbering elements*.)

1. Enter the Mesh module.
2. From the main menu, select **Mesh->Global Numbering Control**.
3. If you are working with the assembly, select the native geometry part instances to modify.

   You can select independent or dependent part instances to modify. However, any operations applied to dependent part instances are performed at the part level and, therefore, apply to all instances of the part, even if not all instances were selected.

   Abaqus/CAE displays the **Global Numbering Control** dialog box and the start label for the nodes and elements. If you selected a single part instance, the values displayed are the current start label for the nodes and elements of the selected part instance. If you selected multiple part instances that have the same start labels for their nodes and elements, those values are displayed; otherwise, no values are displayed.
4. From the **Global Numbering Control** dialog box, enter the start labels for the nodes and elements. Abaqus/CAE Applies the same start labels to each of the part instances you selected.
5. Click **OK** to renumber the nodes and elements.

   If the mesh exists, Abaqus/CAE renumbers the nodes and elements and preserves the original order and incrementation. If the mesh has not been created, the new start labels are applied when Abaqus/CAE meshes the part or part instances.

Additional information

- *Renumbering nodes*
- *Renumbering elements*

Adding layers of wedge elements to tetrahedral mesh boundaries

If you apply the free meshing technique and tetrahedral element shape to a region, an **Insert boundary layer** toggle and **Assign Controls** button are available in the **Mesh Controls** dialog box.

If you add a boundary layer to multiple meshing cells, Abaqus/CAE temporarily fixes the seeding on any internal faces between the cells to simplify creation of the boundary layer across these faces.

1. From the main menu bar, select **Mesh->Controls**. (For detailed instructions, see *Assigning mesh controls*.)

   Note:

   You must select a free tetrahedral mesh to include a boundary layer of wedge elements.

2. Toggle on **Insert boundary layer** near the bottom of the **Mesh Controls** dialog box, and click **Assign Controls**.

   Abaqus/CAE displays the **Boundary Layer** dialog box.

3. Enter the **Wall element height**.

   The wall element height sets the height or thickness of the first layer of wedge elements against the boundary walls.

4. Enter the **Growth factor**.

   Starting from the wall elements and moving inward, the height of each successive layer is determined by multiplying the height of the previous layer by the growth factor. The growth factor must be between `1.0` and `2.0`, inclusive, where `1.0` results in no growth and `2.0` doubles the thickness of each new layer.

5. Enter the number of wedge element layers in the boundary layer.

   Abaqus/CAE displays the total **Boundary layer thickness** based on your entries in Step 3 through Step 5.

6. If desired, toggle on **Inactive faces**, and click **Edit** to select faces that you do not want to include in the boundary layer.

   The inactive faces are expected to be approximately planar and perpendicular to local flow directions or along a plane of symmetry.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

7. If desired, toggle on **Create set**, and enter a set name to save a set containing all the boundary layer elements.
8. Click **OK** to close the **Boundary Layer** dialog box.

   The boundary layer will be created when you mesh the regions.

   If mesh generation fails due to problems in the boundary layers, Abaqus/CAE displays a preview of the boundary layer mesh along with a warning dialog box. Before closing the warning dialog and deleting the mesh preview, look for problems such as self-intersecting layers near sharp corners; the possible failure modes are similar to those described for offset meshes in *Reducing element distortion*

and collapse during mesh offsetting. Make the necessary corrections to the boundary layer controls and try to mesh the regions again.

Additional information

• *Controlling mesh characteristics*

Obtaining mesh information and statistics

This section explains how you use the verification tool in the Mesh module to verify graphically the quality of the elements used in a mesh. This section also describes how you use the Query toolset in the Mesh module to obtain listings of mesh, element, and node statistics.

In this section:

- *Verifying element quality*
- *Obtaining mesh information*

Verifying element quality

You can verify the quality of the mesh in selected regions as well as the quality of selected elements,

To verify the quality of a mesh, select **Mesh->Verify** from the main menu bar. The mesh verify tool allows you to do the following:

- Select a part, or select one or more part instances or regions; and highlight elements that do not meet specified criteria, such as aspect ratio. You can also obtain mesh statistics for each selected part, part instance, or region, such as the total number of elements, the number of highlighted elements, and the average and worst values of the selection criterion.
- Select a part, or select one or more part instances or regions; and highlight elements that do not pass the mesh quality tests that are included with the input file processor in Abaqus/Standard and Abaqus/Explicit.
- Save sets containing the elements highlighted by the verification tests. For native meshes, you can save sets of cells, faces, or edges related to the highlighted elements.

You can also obtain quality information for individual elements. For more information, see *Verifying your mesh*.

Additional information

• *Verifying your mesh*
• *Obtaining mesh information*

Verify selected elements

1. To verify the quality of selected elements, select **Mesh->Verify** from the main menu bar.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also verify selected elements using the  tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. From the **Select the regions to verify by** field in the prompt area, select **Element**.
3. Select the element that you want to verify. Abaqus/CAE displays the following in the message area:
   - The name of the part or part instance
   - The element index
   - The element shape
   - The shape factor for triangle and tetrahedra elements
   - The minimum and maximum face corner angles
   - The aspect ratio
   - The geometric deviation factor
   - The stable time increment
   - The maximum allowable frequency for acoustic elements
   - The shortest edge and longest edge
   - Whether the element passes the checks found in the input file processor in Abaqus/Standard and Abaqus/Explicit
4. Continue selecting elements, as desired.

5. When you have finished selecting elements, either
   - Click mouse button 2 in the viewport, or
   - Select any other tool from the toolbox, or
   - Click the cancel button  in the prompt area, or
   - Click the verify mesh tool in the Mesh module toolbox.

Verify a part, a part instance, or a region

1. From the **Object** field in the context bar, select a part or select the assembly.
2. From the main menu bar, select **Mesh->Verify** from the main menu bar.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

    **Tip:** You can also verify a mesh using the  tool, located in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

3. From the text field in the prompt area, select the type of region to verify:
   - Select **Part** or **Part Instances** and select the part or part instances whose mesh you want to verify, and press mouse button 2.
   - **Geometric Regions**. Select the cells, faces, or edges whose mesh you want to verify, and press mouse button 2.

   Abaqus/CAE displays the **Verify Mesh** dialog box.
4. From the top of the **Verify Mesh** dialog box, click the tab corresponding to the desired verification checks. The following verification types are available:
   - **Shape metrics**
   - **Size metrics**
   - **Analysis checks**

   You can specify verification checks on multiple tabbed pages. Abaqus/CAE refers to the verification checks specified on all three tabbed pages when you click **Highlight**.
5. To save a set containing results of the selected verification checks, toggle on **Create set** near the bottom of the **Verify Mesh** dialog and either accept the default set name or enter a new name for the set.

   If there are any results displayed, Abaqus/CAE creates the set when you click **Highlight**, as described in the following steps.
6. If you want to specify verification checks on the **Shape metrics** tabbed page, do the following:
   a. From the **Shape factor** options, specify the shape factor criterion for triangular elements and tetrahedral elements in your selection. If your selection includes both triangular elements and tetrahedral elements, the **Shape factor** options provide separate controls for each type; if your selection includes only triangular elements or only tetrahedral elements, a single control is provided.
   b. If your selection includes triangular elements, you can specify the small face corner angle and the large face corner angle for triangular elements from the **Tri-Face Corner Angle** options.
   c. If your selection includes tetrahedral elements, you can specify the small face corner angle and the large face corner angle for tetrahedral elements from the **Quad-Face Corner Angle** options.
   d. Specify a value for the **Aspect ratio**.

For a detailed description of the selection criteria, see *Verifying your mesh*.

7. If you want to specify verification checks on the **Size metrics** tabbed page, specify failure criteria for any of the following:

   - **Geometric deviation factor**
   - **Shortest edge**
   - **Longest edge**
   - **Stable time increment**
   - **Maximum allowable frequency for acoustic elements**

   Stable time increment is available only for elements in the Abaqus/Explicit element library. **Maximum allowable frequency for acoustic elements** is available only for acoustic elements in the Abaqus/Standard element library.

   For a detailed description of the selection criteria, see *Verifying your mesh*.

8. If you want to specify analysis checks, click the **Analysis checks** tab, and toggle **Errors** and **Warnings** to select which elements will be highlighted.
9. Click **Highlight**.

   Abaqus/CAE highlights elements that fail the element checks specified in the **Shape Metrics** or **Size Metrics** tabbed pages as warnings. In addition, any elements that generated errors or warnings using the checks found in the input file processor in Abaqus/Standard and Abaqus/Explicit are highlighted in the appropriate colors. If you selected **Create set** in Step 5, Abaqus/CAE saves a set containing the highlighted results. In addition, Abaqus/CAE displays information in the message area, such as the name of the part instance, the total number of elements, the number of highlighted elements, and the average and worst value of the selection criterion.

   Regardless of your selection of **Errors** and **Warnings** in the **Analysis checks** tabbed page, Abaqus/CAE also displays in the message area the total number of elements tested and the number of errors and warnings. In most cases, it will be obvious from the element shape why the input file processor issued an error or a warning. If necessary, you can submit a datacheck analysis from the Job module and review the messages that Abaqus writes to the data file. Abaqus/CAE does not support analysis checks for beam, gasket, or cohesive elements.

10. From the buttons along the bottom of the **Verify Mesh** dialog box, do the following:

    - Click **Reselect** to select different part instances or regions.
    - Click **Defaults** to restore the default element failure criteria on all of the tabs.
    - Click **Dismiss** to close the **Verify Mesh** dialog box.

Your changes to the mesh verification criteria are saved for use in future Abaqus/CAE sessions.

Obtaining mesh information

To obtain information on the mesh, select **Tools->Query** from the main menu bar. You can request information on the following:

- The total number of nodes and elements in a selected part, part instance, or region along with the number of elements of each element shape.
- The type and connectivity of a selected element.
- The positive and negative sides of shell and membrane faces.
- The direction of beam and truss tangents.
- The mesh stack orientation.
- Whether any edges of boundary faces have incompatible interfaces, cracks, or gaps and whether any edges intersect other faces.
- Whether there are free or non-manifold exterior element edges.
- Whether there are unmeshed regions.

1. From the **Object** field in the context bar, select a part or select the assembly.
2. From the main menu bar, select **Tools->Query**.

    **Tip:** You can also select the  tool in the Query toolset.

   Abaqus/CAE displays the **Query** dialog box.
3. From the **Query** dialog box, select one of the following queries:

Shell element normals

Abaqus/CAE displays the part or the assembly using the shaded render style. The side of the shell where the surface normal coincides with the shell normal (top face) is shaded brown; the opposite side (bottom face) is shaded purple.

Beam element tangents

Abaqus/CAE displays cyan arrows indicating the direction of the beam tangents.

Mesh stack orientation

For hexahedral and wedge elements, Abaqus/CAE colors the top face brown and the bottom face purple. Similarly, arrows indicate the orientation of quadrilateral elements. In addition, Abaqus/CAE highlights any element faces and edges that have inconsistent orientation.

Mesh

For an assembly, part or part instance, geometric region, or element, Abaqus/CAE displays the following:

- The total number of nodes and elements in the selected area
- The number of elements for each element shape

By default, Abaqus/CAE displays mesh information in the message area, but you can display this information in a tabular format in the **Mesh Statistics** dialog box by toggling on **Display detailed report** in the prompt area. The **Mesh Statistics** dialog box also enables you to display mesh information by part instance or by element type.

Element

Select an element. Abaqus/CAE displays the following in the message area:

- The element label
- The element topology
- The element type that Abaqus/CAE will use for the analysis
- Nodal connectivity

Mesh gaps/intersections

Select a part instance. Abaqus/CAE highlights any edges of boundary faces of the model in the current viewport with the following:

- Incompatible interfaces
- Cracks or gaps
- Intersections with other faces

In addition, if intersecting elements are found, Abaqus/CAE displays a dialog box that allows you to save a set containing elements that share the highlighted edges. If the model geometry is available, you can save a set containing the cells, faces, or edges related to the highlighted element edges.

Free/Non-manifold edges

If the current viewport contains an assembly with more than one part instance, select one or more solid or shell instances for the query. Abaqus/CAE highlights two types of exterior element edges:

- Free element edges are exterior edges that belong to a single exterior element face.
- Non-manifold element edges are exterior edges shared by more than two adjacent exterior element faces.

If Abaqus/CAE finds free or non-manifold element edges, it displays the combined total number of these edges in the message area. Abaqus/CAE also displays a dialog box from which you can save an element set containing all the elements with the highlighted edges. If there are no free or non-manifold edges, the message area indicates that all exterior element edges are shared by exactly two exterior element faces. *Figure 1* shows an extruded shell part with the free and non-manifold edges highlighted in red. The element edges along outer boundaries of the three planar faces are free edges. The element edges at the intersection of the three faces are non-manifold edges—each shared by three exterior element faces.

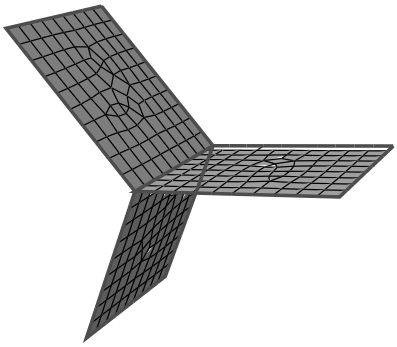


Figure 1: Free element edges make up the exterior edges, and non-manifold edges form the center where the three planar shells are joined.

Shell parts should contain free edges and non-manifold edges where appropriate for the part design. Solid parts should not contain any free or non-manifold edges.

Unmeshed regions

Abaqus/CAE displays a warning dialog and highlights any regions of the model that are not meshed. Toggle on **Save regions in a set** and select a set name to save a set containing the unmeshed regions.

If there are no unmeshed regions, Abaqus/CAE displays a message in the message area indicating that all regions are fully meshed. The query does not consider regions that do not require a mesh, such as display bodies and analytical rigid surfaces.

Unassociated geometry

Abaqus/CAE displays the **Query Unassociated Geometry** dialog box, which enables you to highlight the cells, faces, edges, or vertices in your part or model that are not associated with a mesh. Toggle on **Create set** and select a set name to save a set containing the geometry that is not associated with a mesh.

If all the geometry you select is associated with a mesh, Abaqus/CAE displays a dialog box indicating that all selected geometry is associated with a mesh. The query does not consider regions that do not require a mesh, such as display bodies and analytical rigid surfaces.

4. When you are finished obtaining mesh information, close the **Query** dialog box.

Additional information

- *Controlling mesh characteristics*
- *Verifying element quality*

Creating a mesh part

Select **Mesh->Create Mesh Part** from the main menu bar to create a mesh part with no geometric features from the current meshed part or assembly. You can also create a mesh part from selected part instances that you have meshed. If you have meshed only some regions of the assembly or the selected part instances, Abaqus/CAE creates a mesh part from only those meshed regions; unmeshed regions are not included in the mesh part.

A mesh part contains no feature information and is defined by a collection of nodes, elements, surfaces, and sets. Sets, surfaces, and section assignments are copied from the source parts or part instances when you create a mesh part, so loads and interactions applied to the original parts are also applied to the mesh part. You can add geometric features to a mesh part, and you can use the mesh editing tools to modify its nodes and elements. For more information, see *What can I do with the Edit Mesh toolset?*.

1. From the **Object** field in the context bar, select a meshed part or select the assembly.
2. From the main menu bar, select **Mesh->Create Mesh Part**.
3. If the assembly contains more than one part instance, you must select the part instances that you want to include in the mesh part. In the prompt area, click **Done** to indicate that you have finished selecting part instances.
4. In the prompt area, enter the name of the new part. If you selected part instances from the assembly, you can toggle on **Replace part instances** to replace the assembly instances with an instance of the new mesh part.

   Abaqus/CAE creates the mesh part.

Controlling adaptive remeshing

This section describes how you can define an adaptive remeshing rule and how you can adaptively remesh your model manually.

In this section:

- *Creating a remeshing rule*
- *Selecting the step and error indicator output variables for the remeshing rule*
- *Choosing the remeshing rule sizing method*
- *Choosing remeshing rule constraints*
- *The Remeshing Rules Manager*
- *Manually resizing and remeshing*

Creating a remeshing rule

Remeshing rules enable Abaqus/CAE to adapt your mesh iteratively to meet error indicator targets and sizing criteria that you have specified. For more information, see *What are remeshing rules?*.

1. From the main menu bar, select **Adaptivity->Remeshing Rule->Create**.

   Tip: You can also create a remeshing rule using the tool, located with the mesh tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. Select the regions to which Abaqus/CAE will apply the remeshing rule, or click **Done** to select the entire model. If you assign a remeshing rule to a dependent instance, Abaqus/CAE remeshes the original part and each dependent instance of the part inherits the same mesh. The modeling space of the region must be homogeneous. For example, Abaqus/CAE does not allow you to select a region that contains both solids and shells.

   If necessary, you should use the Partition toolset to isolate regions that will generate stress singularities and to exclude those regions from the remeshing rule. For more information, see *Singularities*.

3. The **Create Remeshing Rule** dialog box appears.
4. If desired, use the **Name** text field to change the name of the new rule.
5. If desired, use the **Description** text field to enter a description of the remeshing rule. You can use the description to help you keep track of the scope and the purpose of your remeshing rules. The **Remeshing Rules Manager** displays the name and the description of a remeshing rule.
6. Click the **Step and Indicator** tab to select the following:

   - The step to which the remeshing rule will be applied.
   - The error indicator output variables that Abaqus/CAE will write to the output database, and the frequency at which they will be written.

   For more information, see *Selecting the step and error indicator output variables for the remeshing rule*.

7. Click the **Sizing Method** tab to select the following:

   - The method that Abaqus/CAE will use to calculate the size of the elements during the remeshing process.
   - Whether to use automatic reduction of error indicator targets for the adaptive remeshing process or whether to specify the error indicator targets.

   For more information, see *Choosing the remeshing rule sizing method*.

8. Click the **Constraints** tab to select constraints on the element size during the remeshing process. For more information, see *Choosing remeshing rule constraints*.
9. Click **OK** to create the remeshing rule and to close the **Create Remeshing Rule** dialog box.

Additional information

- *Understanding adaptivity processes*
- *Understanding adaptive remeshing*
- *Manually resizing and remeshing*
- *About Adaptive Remeshing*

Selecting the step and error indicator output variables for the remeshing rule

When you create a remeshing rule, you must select the step to which the rule will be applied. You must also select the error indicator output variables that Abaqus/CAE will write to the output database during the analysis and the frequency at which they are written. The error indicator output variables are the basis of the adaptive remeshing process. They provide information to Abaqus/CAE that describes where a mesh needs refinement to approach or achieve the desired error indicator goals. In addition, Abaqus/CAE uses the error indicator output variables to determine where the mesh can be coarsened without introducing unacceptable errors. *Selection of Error Indicators Influencing Adaptive Remeshing*.

You can use the adaptivity plotter plug-in to review the history of selected error indicators and the element count during an adaptivity analysis. For more information, see *Displaying the history of the adaptive remeshing error indicator*.

1. From the **Create Remeshing Rule** dialog box, click the **Step and Indicator** tab.
2. Click the arrow to the right of the **Step** field, and select the step of your choice from the list that appears. The remeshing rule will be applied only during this step; however, you can apply a different remeshing rule with the same settings to another step in your model. Adaptive remeshing is available only for procedures that use Abaqus/Standard. In addition, Abaqus/CAE cannot apply adaptive remeshing in frequency extraction and steady-state dynamic procedures. For more information, see *Which procedures can I use with adaptive remeshing?*.
3. Choose the frequency at which the error indicator output variables will be written to the output database. You can choose to write the variables after each increment or after the end of the last increment of the step. Abaqus/CAE remeshes the model based on the value of the error indicators in the last increment of the step. However, if your analysis fails to converge and you saved the error indicator output variables after every increment, you can use the most recent values to remesh your model manually and to continue. For more information, see *Manually resizing and remeshing*.
4. From the list of **Error Indicator Variables**, select one or more variables that will be written to the output database. For each variable that you select, Abaqus/CAE also writes an accompanying base solution variable to the output database.

Additional information

- *Selection of Error Indicators Influencing Adaptive Remeshing*
- *Understanding adaptivity processes*
- *Understanding adaptive remeshing*
- *Creating a remeshing rule*
- *About Adaptive Remeshing*
- *Displaying the history of the adaptive remeshing error indicator*

Choosing the remeshing rule sizing method

You can allow Abaqus/CAE to choose a default sizing method based on the error indicator output variable, or you can specify a sizing method that will be used for all error indicators in the remeshing rule. If you specify the sizing method, you can also specify the adaptivity goal. For more information, see *Solution-Based Mesh Sizing*.

1. From the **Create Remeshing Rule** dialog box, click the **Sizing Method** tab.
2. Click the arrow to the right of the **Method** field, and select one of the following mesh size algorithms:
   - Choose **Default method and parameters** to allow Abaqus/CAE to choose the default calculation method for each error indicator output variable. By default, all of the error indicators except for element energy (ENDENERI) and heat flux (HFLERI) use the **Minimum/maximum control** mesh sizing algorithm. ENDENERI and HFLERI use the **Uniform error distribution** algorithm. The default error target is **Automatic target reduction** with a moderate mesh bias.
   - Choose the **Uniform error distribution** mesh sizing algorithm to force Abaqus/CAE to apply a sizing method that attempts to meet the specified error target in every element in the model. In most cases this approach leads to a globally converging mesh.
   - Choose the **Minimum/maximum control** mesh sizing algorithm to control the mesh density at the location of the minimum and maximum values of the base solution.

   For more information, see *Solution-Based Mesh Sizing*.
3. If you selected **Uniform error distribution**, specify the approach that Abaqus/CAE will use to determine the error indicator target:
   - Choose **Automatic target reduction** to allow Abaqus/CAE to generate successive mesh refinements that attempt to reduce the solution error by a fixed amount from the prior job in the adaptivity process. You select the maximum number of mesh iterations while creating an adaptivity process in the Job module. If you select **Automatic target reduction**, Abaqus/CAE will consider the rule to be satisfied when the error indicator reaches 1%. The criterion of 1% is intended only to provide protection against excessively expensive analysis jobs. In most cases Abaqus/CAE will complete all of the remesh iterations that you specified when you created the adaptivity process in the Job module.
   - Choose **Fixed target**, and enter the percentage error target. Abaqus/CAE uses this value to apply a sizing method that attempts to meet the error target in every element in the model. Such an approach ensures a globally converging mesh. Abaqus/CAE will consider the rule to be satisfied when the error indicator reaches the error indicator target.
4. If you selected **Minimum/maximum control**, do the following from the **Error Indicator Targets** region:
   a. Specify the approach that Abaqus/CAE will use to determine the error target:
      - Choose **Automatic target reduction** to generate successive mesh refinements that attempt to reduce the solution error by a fixed amount from the prior analysis. You select the maximum number of mesh iterations in the Job module. If you select **Automatic target reduction**, Abaqus/CAE will consider the rule to be satisfied when the error indicator reaches 1%. The criterion of 1% is intended only to provide protection against excessively expensive analysis jobs. In most cases Abaqus/CAE will complete all of the remesh iterations that you specified when you created the adaptivity process in the Job module.
      - Choose **Fixed targets**, and enter the percentage error indicator targets at the location of the maximum and minimum value of the base solution. Abaqus/CAE uses these values to apply a heterogeneous sizing method that attempts to meet both targets at their respective locations. Abaqus/CAE will consider the rule to be satisfied when the error indicator corresponding to

the maximum value of the base solution is reached. In addition, both error targets help Abaqus/CAE ensure that mesh refinement is focused on areas of interest. Conversely, they also help Abaqus/CAE ensure that disproportionate refinement is not applied to areas where the values of the base solution results are low.

b. Specify the **Mesh Bias**. The mesh bias further tunes the distribution of sizing between the locations of the maximum and minimum base solutions. When you choose **Strong**, the sizing method acts more aggressively and focuses more elements near locations of high base solution intensity. When you choose **Weak**, the sizing method acts less aggressively and generates fewer elements near locations of high base solution intensity. *Figure 1* illustrates the effect of the bias factor.

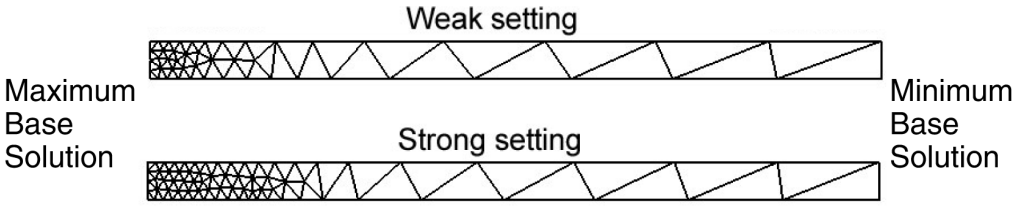


Figure 1: The impact of the bias factor on element size distribution.

Additional information

- *Solution-Based Mesh Sizing*
- *Understanding adaptivity processes*
- *Understanding adaptive remeshing*
- *Creating a remeshing rule*
- *About Adaptive Remeshing*

Choosing remeshing rule constraints

You can define constraints on the element size that Abaqus/CAE will apply during the adaptive remeshing process. You can also specify rate limits that control the introduction of larger and smaller elements and modulate how aggressively Abaqus/CAE applies the sizing method.

The minimum and maximum element sizes that you specify limit the size used in the mesh sizing function, which Abaqus/CAE uses to guide the internal meshing algorithm. The sizes that you specify are not absolute constraints on the element size—the element sizes in the generated mesh only approximate the sizing function. Therefore, some element edges may be larger or smaller than the maximum and minimum element sizes that you specified.

The maximum number of elements that you specify limits the number of elements generated by the mesh sizing function. The mesh sizing function adjusts the target error and the element sizes such that the number of elements generated in a remeshing region does not exceed the specified value. The number of elements in the generated mesh (as with the element sizes) only approximates the sizing function. Therefore, the number of elements generated may be greater than the maximum number of elements that you specified.

1. From the **Create Remeshing Rule** dialog box, click the **Constraints** tab.
2. Specify the minimum and maximum **Element Size**.
   - Specify a **Minimum** value to impose a lower bound on the size of the elements calculated by the remesh algorithm. You can choose either the minimum element size computed by Abaqus/CAE, or you can enter the minimum element size.
   - Specify a **Maximum** value to impose an upper bound on the size of the elements calculated by the remesh algorithm. You can choose either the maximum element size computed by Abaqus/CAE, or you can enter the maximum element size.
3. If desired, specify the **Approximate maximum number of elements** to impose an upper bound on the total number of elements calculated by the remesh algorithm.
4. Specify the **Rate Limits**.
   - The **Refinement** rate limit modulates the aggressivity of the sizing method and controls the introduction of smaller elements. Choose one of the following:
     - **Use default** to specify a refinement limit midway between **High** and **Low**.
     - **Specify** and drag the **Refinement** slider to specify the rate limit. Specifying **Low** indicates minimal reduction in the element size; specifying **High** indicates a maximum reduction of the original element size.

       The refinement factor has a significant effect on the convergence of the adaptive meshing procedure and may help you achieve faster and more efficient mesh convergence.
     - **Do not refine** to prevent reduction in the element size.
   - The **Coarsening** rate limit modulates the aggressivity of the sizing method and controls the introduction of larger elements. Choose one of the following:
     - **Use default** to use the default limit.
     - **Specify** and drag the **Coarsening** slider to specify the rate limit. Specifying **Low** indicates minimal growth in the element size; specifying **High** indicates a maximum growth of the element size.

- **Do not coarsen** to prevent growth in the element size.

Additional information

• *Understanding adaptivity processes*
• *Understanding adaptive remeshing*
• *Creating a remeshing rule*
• *About Adaptive Remeshing*

The Remeshing Rules Manager

The **Remeshing Rules Manager**, which is similar to other managers in Abaqus/CAE, allows you to do the following:

- Create a remeshing rule. For more information, see *Creating a remeshing rule*.
- Edit the selected remeshing rule.
- Copy, rename, suppress, resume, or delete the selected remeshing rule.

You can display the **Remeshing Rules Manager** by selecting **Adaptivity**->**Remeshing Rule**->**Manager** from the main menu bar.

The two columns in the **Remeshing Rules Manager** display the following:

Name

The **Name** column displays the name of the remeshing rule. Click **Rename** to rename the selected remeshing rule.

Description

The **Description** column displays the description of the remeshing rule. You can use the description to help you keep track of the scope and the purpose of your remeshing rules.

Additional information

- *Understanding adaptivity processes*
- *Understanding adaptive remeshing*
- *Creating a remeshing rule*
- *About Adaptive Remeshing*

Manually resizing and remeshing

To learn about the impact of a remeshing rule on the mesh generated by Abaqus/CAE, you can manually apply a rule and view its impact on the resulting mesh. However, before you can remesh your model, you must run an analysis and create an output database that contains error indicator output variables. You can then change the sizing methods and constraints and view the impact of the modified rule on the generated mesh. When you are confident that a remeshing rule is capturing your intent, you can use the same rule to control several iterations that are driven by automatic remeshing. You can also use manual remeshing to continue an adaptivity process that ended prematurely.

For more information, see *What is the difference between automatic adaptive remeshing and manual adaptive remeshing?*, and *When do I need to use manual adaptive remeshing?*.

1. From the main menu bar, select **Adaptivity->Manual Adaptive Remesh**.

   Abaqus/CAE displays the **Manual Adaptive Remesh** dialog box.

   Tip: You can also manually resize and remesh using the tool, located with the mesh tools in the Mesh module toolbox. (For more information, see *Using the Mesh module toolbox*.)

2. In the **ODB** field, enter the name of the output database that contains the error indicator output variables. Abaqus/CAE uses the error indicators to drive the remeshing algorithm.

   Abaqus/CAE displays the following information about each of the remeshing rules in the output database:

   - The name of the rule.
   - A description of the error indicator output variable that was used by the rule.
   - The percentage target error. This is the value you specified when you created the remeshing rule. If you selected fixed targets, Abaqus/CAE displays the maximum and/or minimum base solution targets. Abaqus/CAE displays a dash if you selected **Default methods and parameters** or **Auto** if you selected **Automatic target reduction**.
   - The sizing method used by the rule.

3. If desired, click **Display Error**.

   Abaqus/CAE displays the percentage **Error Indicator Result** and the **Element Count** for each error indicator output variable. The element count is the number of elements in the region to which the remeshing rule is applied.

4. Click **Manual Adaptive Remesh**.

   Abaqus/CAE remeshes the model. All of the active (not suppressed) remeshing rules contribute to the remeshing of the model. If you applied more than one remeshing rule to the same region, the rule that results in the finest mesh takes precedence.

5. If desired, you can select **Adaptivity->Remeshing Rule->Edit->*rule name*** from the main menu and modify the parameters that define the rule. You can then return to the **Manual Adaptive Remesh** dialog box and view the effect of the modified rule on the error estimate and the resulting mesh.

   Note:

   When you edit a remeshing rule, you can modify parameters such as the remeshing rule sizing method and constraints on the element size and view the effect of your changes on the resulting mesh. However, if you change the step or the error indicator output variables, you must run the analysis again.

6. You can modify the remeshing rules and remesh your model manually until you are satisfied with the resulting mesh. You can then use the same rules to drive a sequence of iterative remeshing operations that are controlled by Abaqus/CAE. For more information, see *Creating, editing, and manipulating adaptivity processes*.

Additional information

- *What is the difference between automatic adaptive remeshing and manual adaptive remeshing?*
- *When do I need to use manual adaptive remeshing?*
- *Understanding adaptivity processes*
- *About Adaptive Remeshing*

The Optimization module

You use the Optimization module to create an optimization task that can be used to optimize the topology or shape of your model given a set of objectives and a set of restrictions. For example, an optimization can attempt to remove material from selected regions to satisfy a maximum weight objective while maintaining a minimum stiffness.

In this section:

- *Understanding the role of the Optimization module*
- *Entering and exiting the Optimization module*
- *Understanding optimization*
- *Using the Optimization module toolbox*
- *Viewing and troubleshooting an optimization*
- *Creating and configuring an optimization task*
- *Configuring design responses*
- *Creating objective functions*
- *Creating constraints*
- *Configuring geometric restrictions*
- *Creating local stop conditions*

Understanding the role of the Optimization module

You can use the Optimization module to perform the following tasks:

Create optimization tasks

An optimization task contains the definition of your optimization. You run an optimization in the Job module using an optimization process. An optimization process refers to an optimization task.

Create design responses

A design response is a single scalar value that is extracted from an optimization. A design response can be extracted directly from the output database, such as the volume of the model. Alternatively, the Optimization module can extract data from the output database and calculate the design response, such as the total strain energy of the model, a measure of its flexibility.

Create objective functions

An objective function defines the objective of the optimization and refers to the value of a design response or a combination of design responses. For example, the objective function of the optimization can be to minimize the total strain energy in the model (maximize its stiffness).

Create constraints

Constraints define the changes that the Optimization module can apply to the topology or the shape of the model during the optimization. For example, the volume of the optimized model can be constrained to be 50% of the original volume. If a constraint cannot be satisfied, the optimization is not feasible. A constraint also refers to the value of a design response, but it cannot refer to a combination of design responses.

Create geometric restrictions

A geometric restriction places restrictions on the changes that the Optimization module can make to the topology of the model. Geometrical restrictions include frozen regions from which material cannot be removed and manufacturing constraints, such as restrictions on cavities and undercuts, that would prevent the optimized model from being removed from a mold.

Create stop conditions

A stop condition is an indicator that the optimization has converged to a solution. For example, an optimization can be considered complete after a specified number of iterations or when the change in an optimization function between iterations is less than a specified value.

Entering and exiting the Optimization module

You can enter the Optimization module at any time during an Abaqus/CAE session by clicking **Optimization** in the **Module** list located in the context bar. The **Task**, **Design Response**, **Objective Function**, **Constraint**, **Geometric Restriction**, **Stop Condition**, and **Tools** menus appear on the main menu bar. If the current viewport contains something other than the assembly, the contents of the viewport disappear when you start the Step module.

To exit the Optimization module, select any other module from the **Module** list. You need not save your optimization definition before exiting the module; it will be saved automatically when you save the model database by selecting **File**->**Save** or **File**->**Save As** from the main menu bar.

Understanding optimization

An optimization is an iterative process that modifies the structure of your model while it searches for an optimized solution given a set of objectives and constraints that must be satisfied. This section briefly describes the components of an optimization that you can create with the Optimization module.
For more detailed information, see *About Structural Optimization*.

In this section:

- *About tasks*
- *About design responses*
- *About objective functions*
- *About constraints*
- *About geometric restrictions*
- *About stop conditions*
- *About optimization processes*

About tasks

An optimization task contains the components that define your optimization, such as the design responses, objectives, constraints, and geometric restrictions. To run an optimization, you execute an optimization process. An optimization process refers to an optimization task.

For more information, see *Creating an optimization task*.

About design responses

The inputs to your optimization are called the design responses. Design responses can be read from the Abaqus output database file; for example, stiffness, stresses, eigenfrequencies, and displacements. Alternatively, the Optimization module can extract the design responses from the position of nodes or the layout of elements in your model; for example, its weight, center of mass, or inertia.

A design response is associated with a region of your model; however, it consists of a single scalar value, such as the maximum stress in a region or the total volume of the model. In addition, a design response can be associated with a particular step or load case. For more information, see *Configuring design responses*, and *Design Responses*.

About objective functions

The objective functions define the objective of your optimization. The objective function is extracted from a design response, such as the lowest eigenfrequencies or the minimum stress. The objective function can be formulated from multiple design responses. If you specify that the objective functions minimize or maximize the design responses, the Optimization module calculates the objective function by adding each of the values determined from the design responses. In addition, you can specify a weighting factor (the default weighting factor is 1.0). For the most common optimization formulations you do not need to change the default value of the weighting factor. However, in some cases you may have to change the weighting factor to balance the effect of an objective function that is dominating the optimization. You should be aware that changing the weighting factor can have a significant impact on the final design. In addition, a design response that is dominant at the start of the optimization may have less effect as the Optimization module modifies your model. For more information, see *Creating objective functions*, and *Objectives and Constraints*.

About constraints

Constraints are also extracted from the design responses. Constraints restrict the value of a design response; for example, you can specify that the volume must be reduced by 45% or the absolute displacement in a region must not exceed 1 mm. You can also apply manufacturing and geometric constraints that are independent of the optimization; for example, a structure must be able to be cast or stamped or the diameter of a bearing surface cannot be changed.

When you execute an optimization process, Abaqus generates history output from the constraints you defined in the Optimization module. For volume design responses the history output is always reported as a fraction of the initial value. For all other design responses the history output is reported as an absolute value.

Satisfying the constraints has priority over the minimization or maximization of the objective function. The optimization algorithms start to maximize or minimize the objectives only after the constraints are satisfied.

You can specify only a volume constraint for a condition-based topology or shape optimization, and the volume constraint must be either equal to a fixed value or a fraction of the value before the optimization starts. If the requested volume differs greatly from the initial volume, the Optimization module may need several design cycles to satisfy the volume constraint. A general topology optimization provides more flexibility; and you can select any of the design responses as a constraint, except for the design response that calculates the eigenfrequencies with the Kreisselmeier-Steinhauser formula. The constraint in a general topology optimization can be less than, greater than, or equal to a fixed value or a fraction of the value before the optimization starts. For more information, see *Creating constraints*, and *Objectives and Constraints*.

About geometric restrictions

Geometric restrictions are manufacturing and geometric constraints that are independent of the optimization; for example, a structure must be able to be cast or stamped or the diameter of a bearing surface cannot be changed. For more information, see *Configuring geometric restrictions*, and *Objectives and Constraints*.

About stop conditions

A global stop condition defines the maximum number of iterations an optimization should perform. A local stop condition specifies that the optimization should end when a local minimum (or maximum) has been reached. For more information, see *Creating local stop conditions*, and *Objectives and Constraints*.

About optimization processes

You create optimization processes in the Job module. An optimization process reads an optimization task that you defined in the Optimization module and iteratively searches for an optimized solution based on the objective functions and constraints that you defined in the optimization task. For more information, see *What is an optimization process?*. You can use a view cut in the Visualization module to view the results of an optimization process. For more information, see *Cutting through a model*.

Using the Optimization module toolbox

You can access all the Optimization module tools through either the main menu bar or through the Optimization module toolbox. *Figure 1* shows the icons for all the Optimization module tools in the toolbox.

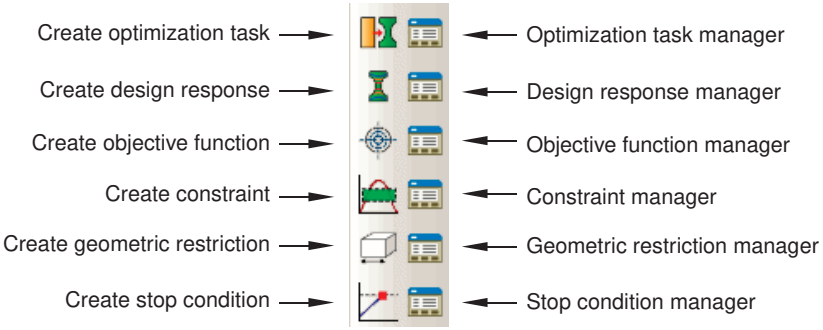


Figure 1: The Optimization module tools.

To see a tooltip containing a brief definition of an Optimization module tool, hold the mouse over the tool for a moment. For information on using toolboxes and selecting hidden icons, see *Using toolboxes and toolbars that contain hidden icons*.

Viewing and troubleshooting an optimization

You can use the field and history output generated by Abaqus to view the results generated by your optimization process. You can also use the output to diagnose any problems with the optimization; for example, to determine if the optimization is converging on an objective or to study the rate of convergence. You view the results by clicking **Results** in the **Optimization Process Manager**.

When you submit an optimization process for analysis, Abaqus creates an output database (`.odb`) file for each design cycle of the optimization. The output database files are stored in the *jobname*`\SAVE.odb` directory. You must combine the separate output database files into a single output database file before you can view the optimization in the Visualization module. The behavior of the Visualization module depends on whether the output database file was created from a topology optimization, a shape optimization, a sizing optimization, or a bead optimization.

Topology optimization

When you view the results of a topology optimization, Abaqus/CAE automatically displays a view cut superimposed on the current view that represents the optimized design surface. The isosurface variable of the view cut is the normalized material property that the Optimization module uses to "add" or "remove" elements from the analysis. By default, Abaqus/CAE displays the view cut with the normalized material property set to 0.3. You can use the view cut manager to modify the value of the isosurface variable and to view the resulting boundary of the isosurface. Boundary conditions are not displayed while an optimization view cut is active. For more information, see *Managing view cuts*.

Shape optimization

When you view the results of a shape optimization, Abaqus/CAE displays the model in its optimized shape using the new position of the surface nodes.

Sizing optimization

When you view the results of a sizing optimization, Abaqus/CAE displays the shell model and the optimized shell thickness, which varies as the optimization progresses. (When you view the results of a nonoptimized analysis of a shell model, the shell thickness is read from the model data and does not change during the analysis.)

Bead optimization

When you view the results of a bead optimization, Abaqus/CAE displays the shell model in its optimized shape using the new position of the surface nodes.

After you combine the separate output database files into a single output database file, each design cycle in the optimization process appears as a frame in the output database, and you can open the **Step/Frame** dialog box to display the results from each design cycle. You can view the results of an optimization process while it is still in progress. As the optimization process moves toward completion, the lists of completed steps and frames are updated every time you close and reopen the **Step/Frame** dialog box. For more information, see *Selecting a specific results step and frame*, and *Stepping through frames*.

You can perform a time history animation of a deformed or contour plot and view the progress of the optimization as Abaqus attempts to satisfy the objective functions while respecting the constraints. For a topology optimization you can view the progressive removal of elements from the analysis and the resulting effect on the mechanical behavior of the model, such as the change in deformation or stress. Turning on translucency allows you to see the progression of the optimization through interior elements; for example, the removal of interior elements to create voids. For more information, see *Changing the translucency*. For a shape optimization you can view the incremental change in the position of the surface nodes as the optimization progresses; and, similar to a topology optimization, you can view the resulting effect on the mechanical behavior of the model.

In addition, the optimization process writes data files (`optimization_report.csv` and `optimization_status_all.csv`) to the *jobname* directory that you can use to track the design variables. For example, you can create an *X–Y* plot showing how the objective functions and constraints change after each design cycle. The resulting plot indicates how the Optimization module tries to meet the specified objective functions while satisfying the constraints. You can use an *X–Y* plot of the objective functions and constraints as a diagnostic tool to view the progression of the optimization after each design cycle and to determine if the optimization is converging on a solution, as shown in *Figure 1*.

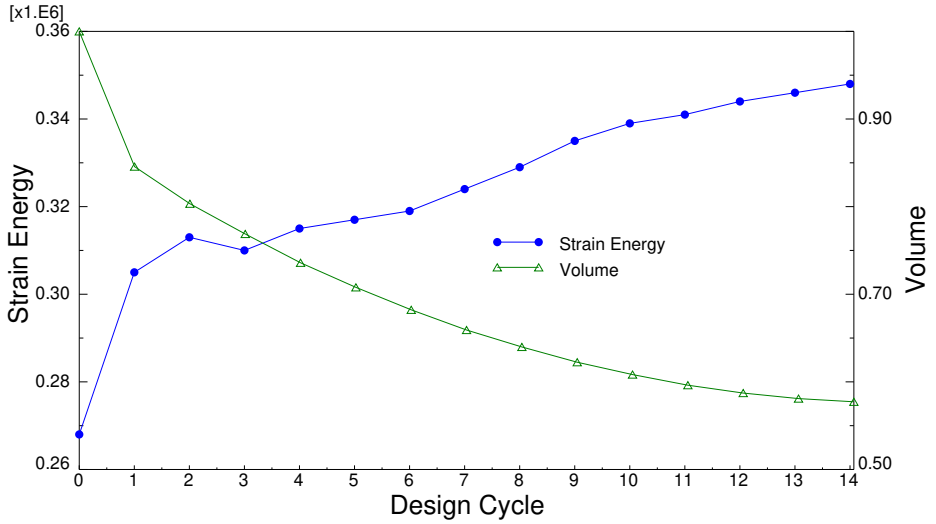


Figure 1: The design objective and constraint during an optimization.

You can also view an *X–Y* plot showing how the objective function and constraints change after each design cycle by monitoring the progress of an optimization process from the optimization process manager. See *Monitoring your optimization process*, for more information.

Additional information

- *Understanding view cuts*
- *Understanding X–Y plotting*

Creating and configuring an optimization task

You can create and configure an optimization task. An optimization task contains the definition of your optimization.

In this section:

- *Creating an optimization task*
- *Configuring a topology optimization task*
- *Configuring a shape optimization task*
- *Configuring a sizing optimization task*
- *Configuring a bead optimization task*

Creating an optimization task

You can create a topology optimization task. An optimization task contains the definition of your optimization, such as the design responses, objectives, constraints, and geometric restrictions. To run an optimization, you create an optimization process in the Job module. An optimization process refers to an optimization task.

1. From the main menu bar, select **Task->Create**.

   The **Create Optimization Task** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Optimization Task Manager**. (You can display the **Optimization Task Manager** by selecting **Task->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Optimization Task** dialog box that appears, enter the name of the task.
3. Select the optimization **Type** (**Topology optimization**, **Shape optimization**, **Sizing optimization**, or **Bead optimization**), and click **Continue**. For more information, see *About Structural Optimization*.
4. From the viewport, select the region that will be optimized or click **Done** to optimize the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   The **Edit Optimization Task** dialog box appears.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click the button—**Select in Viewport** or **Sets**—on the right side of the prompt area.

5. When you have finished selecting the optimization region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."
6. Configure the optimization task, as described in *Configuring a topology optimization task*, *Configuring a shape optimization task*, *Configuring a sizing optimization task*, and *Configuring a bead optimization task*.

Additional information

- *Configuring a topology optimization task*
- *Configuring a shape optimization task*
- *Configuring a sizing optimization task*
- *Configuring a bead optimization task*

Configuring a topology optimization task

The Optimization module provides a variety of settings that allow you to configure a topology optimization task. The configuration settings depend on whether you are configuring an optimization task for a general topology optimization or a condition-based topology optimization.

In this section:

- *Configuring a general topology optimization task*
- *Configuring a condition-based topology optimization task*

Configuring a general topology optimization task

A general topology optimization is a flexible, sensitivity-based optimization that allows you to apply a range of constraints and objective functions to your model. You use the optimization task editor to customize various aspects of a general topology optimization.
To locate the editor, select **Task->Edit->*optimization task name*** from the main menu bar. To specify a general topology optimization, select the **Advanced** tab and choose **General optimization (sensitivity-based)**.

In this section:

- *Configuring basic settings*
- *Configuring the density settings*
- *Configuring the perturbation settings*
- *Configuring convergence options*
- *Configuring advanced options*

Configuring basic settings

You can configure a general topology optimization.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to freeze load or boundary condition regions.

   It is recommended that you freeze regions to which prescribed conditions are applied because you do not want these regions to be removed during the optimization. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.

Configuring the density settings

You can configure a general topology optimization.

1. In the optimization task editor, click the **Density** tab.
2. Select the **Density update strategy**.

   This setting controls the rate at which the Optimization module updates the relative material density of design elements during the optimization. In most cases you should accept the default setting (**Normal**). However, if the design responses are very sensitive and you have problems fulfilling the constraints, you may need a more conservative rate that requires more optimization iterations.
3. Do either of the following to specify the relative density of each element during the initial optimization iteration:

   - Select **Optimization product default** to allow the Optimization module to determine the initial density. If the material volume is selected as a constraint, the Optimization module calculates the initial density such that the volume constraint is fulfilled exactly. If the material volume is selected as an objective function, each element has an initial relative density of 50%.
   - Select **Specify** and enter a value ($0.0 < \textit{initial density} \quad 1.0$). You should use this option only if volume is selected as an objective function and not as a constraint and if you know, prior to the optimization, that setting the initial density to a larger or smaller value will fulfill other constraints; for example, displacement constraints. You can use a value greater that 0.5 in conjunction with volume constraints to stabilize nonlinear or contact problems and to improve the convergence behavior.
4. Enter the **Minimum density**, the **Maximum density**, and the **Maximum change per design cycle**.

   The minimum density must be greater than 0.0, and the maximum density must be less than or equal to 1.0. Changing the density bounds is not recommended, in particular the upper bound. You may need to increase the lower bound if the default value leads to a nearly singular stiffness matrix.

   Numerical experiments indicate that a value of 0.25 (default) is acceptable for the maximum change in density. A lower limit in the change of density, such as 0.1, is recommended for complicated design responses and optimization formulations. However, a lower limit often leads to a higher number of optimization iterations.

Configuring the perturbation settings

You can configure a general topology optimization.

1. In the optimization task editor, click the **Perturbation** tab.
2. Enter the number of eigenmodes to track. The default value is five, which means that the Optimization module tracks the five lowest eigenfrequencies.

   In some cases many local low frequency eigenmodes appear during the optimization iterations, which leads to a high number of modes to track and degrades performance. You can avoid tracking a high number of modes by choosing the lower bound of the eigenfrequencies to be 25% of the eigenfrequency of interest in the first optimization iteration.

   Mode tracking is not required if your design response will use the Kreisselmeier-Steinhauser formulation to evaluate the eigenfrequencies. Your Abaqus model must include an output request for at least the number of eigenfrequencies you are tracking.
3. Select the region over which the Optimization module should track the eigenmodes.

   By default, the Optimization module tracks the eigenmodes of all the nodes in the model, which can degrade performance if you have a large model. You can improve the performance by tracking the eigenmodes over only a selected region; for example, over selected surfaces of your model or over points where lumped or rigid masses are attached.

Configuring convergence options

You can configure a general topology optimization.

1. In the optimization task editor, click the **Convergence** tab.
2. Specify the **Convergence Criteria**. The following options allow you to specify the convergence criteria for a general topology optimization:

Specifying when to start checking for convergence

You can specify the iteration during which the Optimization module will begin to check the two convergence criteria. The optimization will always continue at least until this value has been reached. The default value is 4.

Specifying which convergence criterion to check

You can specify whether the optimization should end when either of the convergence criterion has been fulfilled or both of the criteria have been fulfilled. The default value is that both criteria must be fulfilled.

Convergence based on the change in optimization function

You can specify that the optimization will end based on the change in the objective function from one iteration to the next. The default value is 0.001.

Convergence based on the change in element densities

Element density is the design variable for a topology optimization. You can specify that the optimization will end based on the average change in the element density from one iteration to the next. The default value is 0.005.

Configuring advanced options

You can configure a general topology optimization.

1. In the optimization task editor, click the **Advanced** tab.
2. Select the **General optimization** algorithm.
3. Choose whether to **Delete soft elements in region**.

   During the topology optimization process, the Optimization module distributes a given mass within the design area while it tries to satisfy the constraints and optimize the objective. At the end of the optimization, the structure contains hard (filled) and soft (void) elements. The soft elements have a negligible influence on the stiffness of the structure; but they are still relevant for the number of degrees of freedom of the structure and, hence, influence the speed of the optimization process. The **Delete soft elements** option allows you to select a region from which soft elements that have only soft neighboring elements will be removed. The deleted elements are reactivated if needed; for example, if the force flow changes during the optimization.

   Choosing to delete soft elements can help Abaqus converge on a solution because those elements would otherwise degenerate or collapse and is recommended when you are optimizing a nonlinear model. In addition, selecting a **Conservative** density update strategy and a small change in density per design cycle will improve the accuracy of the results. See *Configuring the density settings*, for more information.
4. If you chose to delete soft elements, you can prevent isolated soft elements from being removed by choosing to delete only soft elements that have neighboring soft elements. You can define a neighboring element as being within the radius specified by the **Average edge length** (default) or specified by a value that you enter. If the element edge length varies considerably within the mesh, the radius calculated from the average edge length can be misleading.
5. If you chose to delete soft elements, you can select the method that the Optimization module will use to delete elements:

Favor continuity (Standard)

Choose **Favor continuity (Standard)** and enter a **Relative material density threshold** to check for continuity before deleting soft elements. If the optimized model contains an “island” of hard elements that are separated from the rest of the model by soft elements, the Optimization module does not remove the soft elements. In addition, the Optimization module retains soft elements that are preventing hard elements from moving with respect to each other; for example, hard elements that share a common edge but not a common face. An element is considered “soft” if its relative material density is less than the threshold value, and the Optimization module removes it from the analysis.

Favor continuity (Aggressive)

Choose **Favor continuity (Aggressive)** and enter a **Relative material density threshold** to remove soft elements regardless of continuity. An element is considered “soft” if its relative material density is less than the threshold value, and the Optimization module removes it from the analysis.

Maximum shear strain

Choose **Maximum shear strain** and enter a **Maximum shear strain threshold**. The Optimization module removes elements from the analysis that have a shear strain larger than the threshold.

Minimum principal strain

Choose **Minimum principal strain** and enter a **Minimum principal strain threshold**. The Optimization module removes elements that have a principal strain lower than the threshold.

Maximum elastoplastic strain

Choose **Maximum elastoplastic strain** and enter a **Maximum elastoplastic strain threshold**. The Optimization module removes elements that have an elastoplastic strain larger than the threshold.

Volume compression

Choose **Volume compression** and enter a **Relative volume compression**. The Optimization module removes elements that are compressing and have a relative volume that is lower than the threshold. The relative volume V_{rel} is defined as $\frac{V_{deform}-V_{org}}{V_{org}}$, where V_{deform} is the deformed element volume and V_{org} is the original element volume.

You should choose **Volume compression** if your model uses shell or membrane elements or if your model is experiencing large deformations.



Note:

The soft delete method that you select is dependent on the material behavior and the element type, and you might have to experiment to determine the best method and its threshold value. The file, `TOSCA.OUT`, contains information about the elements that are being removed and will help you determine the best soft delete method and threshold value. The **Favor continuity** methods provide a default **Relative material density threshold** of 0.05. In contrast, the strain and volume methods do not provide a default threshold because the appropriate value depends on your model; for example, on the properties of the materials.

6. Choose the **Material interpolation technique** and the **Penalty factor**.

   Optimization generates hard elements with a density close to one or void elements with a density close to zero. Topology optimization introduces elements with a density between one and zero, and the material interpolation technique calculates the relationship between density and stiffness for these intermediate elements. The **SIMP** (solid isotropic material with penalization) interpolation scheme defines an exponential relationship between the density and the stiffness of an element and is suitable for static problems. The penalty factor should be greater than `1`, and numerical experiments indicate that the default value of 3 produces good results. The **RAMP** (rational approximation of material properties) interpolation scheme is suitable for dynamic problems. The penalty factor should be greater than 0, and numerical experiments indicate that the default value of 3 produces good results. The **MIMP** material interpolation is a combination of the SIMP for the constitutive material interpolation and a new material interpolation for the physical density.

   By default, the Optimization module selects the **SIMP** interpolation scheme for static problems and the interpolation scheme is switched to **PEDE** automatically by TOSCA if at least one dynamic load case appears in your model.

7. You can choose **Use Abaqus sensitivities when possible** to use Abaqus to compute the design responses and their sensitivities whenever possible. This workflow modification improves the optimization process performance.

8. Select **Use Group Operator when possible** to use large groups of more than 5000 elements or nodes in the design response definition in an efficient way. This workflow modification uses a new algorithm based on Abaqus sensitivities.

Configuring a condition-based topology optimization task

A condition-based topology optimization uses a strain energy objective function and a volume constraint. You use the optimization task editor to customize various aspects of the condition-based topology optimization. To locate the editor, select **Task**->**Edit**->***optimization task name*** from the main menu bar. To specify a condition-based topology optimization, select the **Advanced** tab and choose **Condition-based optimization**.

In this section:

- *Configuring basic settings*
- *Configuring advanced options*

Configuring basic settings

You can configure a condition-based topology optimization.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to freeze load or boundary condition regions.

   It is recommended that you freeze regions to which prescribed conditions are applied because you do not want these regions to be removed during the optimization. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.

Configuring advanced options

You can configure a condition-based topology optimization.

1. In the optimization task editor, click the **Advanced** tab.
2. Select the **Condition-based optimization** algorithm.
3. Select the rate at which the Optimization module will modify the element properties during a topology optimization. You can select the rate (**Very small**, **Small**, **Moderate**, **Medium**, or **Large**) and allow the Optimization module to calculate the number of design cycles required to meet this rate.

   Alternatively, you can select **Dynamic** and enter the maximum number of design cycles. The minimum number of design cycles is 10, and the default value is 15. A reduction in the number of design cycles can lead to undesired effects in the optimization. Although the resulting structures have the same stiffness (the sum of the strain energy is almost equal for the different results), changing the optimization speed can cause a different truss configuration in the solution.
4. Select the volume deleted after the first cycle. You can enter a percentage or an absolute value.

   By default, the Optimization module removes 5% of the optimization region volume in the first iteration. In some cases increasing this starting value will accelerate the optimization without influencing the solution, especially for models where relatively low stresses are present in large areas. Conversely, the Optimization module may remove too many elements in the first iteration if the starting value is too high, leading to a failure in the optimization or a coarse structure.

Configuring a shape optimization task

Shape optimization determines the displacement of each surface node in an effort to homogenize the stress on the surface and satisfy the objective function and any constraints.
You can use the optimization task editor to customize various aspects of the shape optimization and to incorporate durability analysis into the optimization. You can specify a condition-based shape optimization (default) or a general (sensitivity-based) shape optimization.

To locate the editor, select **Task**->**Edit**->***optimization task name*** from the main menu bar. The default settings for shape tasks provide reasonable results across a variety of optimization models; and, in most cases, you do not need to modify the default settings.

In this section:

- *Configuring basic settings*
- *Configuring mesh smoothing quality*
- *Configuring advanced options*
- *Configuring durability options*

Configuring basic settings

During a shape optimization the Optimization module modifies the surface of your model. If the Optimization module modifies only the surface nodes without adjusting the inner nodes, the layer of surface elements will become distorted. Therefore, the results of the Abaqus analysis will no longer be reliable, and the quality of the optimization will suffer. To maintain the quality of the surface elements, the Optimization module applies mesh smoothing to selected regions, which adjusts the position of the inner nodes in relation to the surface nodes. For more information, see *Applying Mesh Smoothing to a Shape Optimization*.

The Optimization module can apply mesh smoothing only to triangular, quadrilateral, and tetrahedral elements. Other element types are ignored during the mesh smoothing.



Note:

It is important to have a good quality finite element mesh before you start the shape optimization, especially in areas where you expect the shape to change.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to freeze boundary condition regions.

   Regions to which you have applied a displacement boundary condition in the Load module have the same displacement boundary condition during the optimization. The coordinate system that governs the displacement boundary condition in the Load module is used to govern the boundary condition in the optimization.
3. By default, the Optimization module freezes boundary conditions for the whole model. If desired, click  and select the region in which the boundary conditions should be frozen.
4. Select the region to which the mesh smoothing will be applied:

   - Select **Specify smoothing region** (default), and click  to select the region. You can apply mesh smoothing to cells or faces.
   - Select **Specify first layer**, and click  to select the faces representing the first layer of elements to smooth. Enter the number of element layers to smooth.
   - Select **Smooth six layers using the task region** to smooth six layers of elements of the design region.

   It is recommended that you accept the default selection and manually select the region to which mesh smoothing will be applied.
5. Select the number of layers of nodes adjacent to the design region that should be allowed to move during the mesh smoothing operation:

   - Select **Fix all** (default) to prevent free surface nodes from moving.
   - Select **Fix none** to allow all the free surface nodes to move.
   - Select **Specify** and enter the number of adjacent layers of free surface nodes that should be allowed to move.

Configuring mesh smoothing quality

Mesh smoothing attempts to improve the quality of the mesh despite the mesh distortion that results from the displacement of the design nodes during the shape optimization. You can specify the relative quality of the smoothed mesh, and you can specify the range of angles (quadrilateral and triangular elements) or the range of aspect ratios (tetrahedral elements) that define an element that is considered good quality. Elements that are considered poor are given a quality rating. The poorer an element is rated, the greater the consideration it will be given in improving the element quality.

1. In the optimization task editor for a shape optimization, click the **Mesh Smoothing Quality** tab.
2. Do either of the following:
   - Toggle on **Target mesh quality**, and select a setting (**Low**, **Medium**, or **High**).

     In most cases you should accept the default setting of **Low**. You should select a higher convergence level only after you have determined the mesh quality is not satisfactory. Even though it is computationally expensive, you may want to select a higher convergence level if your mesh contains a large number of tetrahedral elements; otherwise, the mesh quality may not be acceptable.

     If you are unable to obtain a satisfactory mesh quality, even with a convergence level of **High**, you should consider reducing the amount of displacement during the shape optimization by reducing the **Growth scale factor** and the **Shrink scale factor**, as described in *Configuring advanced options*.
   - Toggle off **Target mesh quality** to deactivate the algorithm that calculates the element quality.
3. Toggle on **Report poor quality elements** to generate a list of elements that fall outside the ranges defined in the element quality table.
4. Toggle on **Report solver quality criteria violation** to report elements that Abaqus considers to be of poor quality.
5. If you toggled on **Report solver quality criteria violation**, you can choose to stop the optimization process if Abaqus encounters elements of poor quality. It is possible that the Optimization module will generate a poor quality mesh that will not allow the Abaqus analysis to complete successfully, especially as the number of design cycles increases. If Abaqus stops the analysis prematurely, no results are available to the Optimization module, and the optimization ends prematurely. If you allow the Optimization module to stop the optimization because the Abaqus element quality criteria is violated, it will be easier for you to troubleshoot the optimization and determine why it failed.
6. If you chose to allow the Optimization module to adjust mesh quality, you can use the table to specify the range of angles (quadrilateral and triangular elements) or the range of aspect ratios (tetrahedral elements) that define an element that is considered good quality. You can also enter the maximum skew angle for quadrilateral and tetrahedral elements and the maximum taper for quadrilateral elements. In most cases, you should not modify the default values. Modifying the range of angles or aspect ratios has a minimal effect on the quality of the mesh. You should try to match the acceptable mesh quality in the Optimization module with the acceptable mesh quality in Abaqus. It is preferable to have your optimization process end because of degrading mesh quality rather than allowing Abaqus to end the optimization process or generate meaningless results.
7. Choose the strategy or algorithm that the mesh smoothing operation will use. By default, the Optimization module uses the **Constrained Laplacian** mesh smoothing algorithm. If you have a relatively small model (less than 1000 nodes in the mesh smooth area), you can select the **Local gradient** mesh smoothing algorithm.
8. If you selected a strategy of **Constrained Laplacian**, do the following:
   a. Select the **Convergence level**, a measure of the amount of time the Optimization module should spend trying to improve the quality of the mesh. In most cases you should accept the default value of **Low**, which results in the Optimization module applying a few iterations with large increments. Selecting **Medium** or **High** will result in more iterations with smaller increments; however, the

computational time will increase significantly. You should use the **Mesh Smoothing Quality** tabbed page to adjust the target mesh quality before you modify the convergence level.

b. Select the **Frequency of evaluating geometric restrictions**, which determines how often the Optimization module applies any geometric restrictions while the mesh smoothing algorithm is executing. In most cases you should accept the default value of **Low**. Selecting **Medium** or **High** will result in the Optimization module applying the geometric restrictions more often, and the computation time will increase significantly.

9. If you selected a strategy of **Local gradient**, enter the **Feature recognition angle**, which is the angle that the Optimization module uses during the mesh smoothing operation to recognize features by detecting edges and corners. The default value is 30°, which provides good results in most cases.

Configuring advanced options

You can configure a shape optimization task.

1. In the optimization task editor, click the **Advanced** tab.
2. Choose the shape optimization algorithm.
   - Select **General optimization (sensitivity-based)** to use a sensitivity-based shape optimization that allows you to apply a range of constraints and objective functions to your model.
   - Select **Condition-based optimization** (default) to use a condition-based shape optimization that uses a strain energy objective function and volume constraint.
3. Enter values specifying the **Growth scale factor** and the **Shrink scale factor**. The growth scale factor is applied to the displacement of nodes that are growing (increasing the volume of the model) as a result of the shape optimization. The shrink scale factor is applied to the displacements of nodes that are shrinking (decreasing the volume of the model) as a result of the shape optimization.

   It is recommended that you perform an optimization with default scale factors of 1.0 and examine the results before you attempt an optimization with modified scale factors. A value greater than 1.0 increases the incremental displacement of nodes and accelerates the optimization. Conversely, a value less than 1.0 decreases the incremental displacement of nodes and slows down the optimization.

   You should consider increasing the scale factors if the first few iterations of the optimization produce little change in the position of the surface nodes; for example, if you have a dense mesh with small element edge lengths. Conversely, if the scale factor is too large, mesh quality will suffer, individual elements might collapse, and the optimization might not be able to converge on the optimal solution.

   You should consider decreasing the scale factors if the original model is close to being optimal. Decreasing the scale factor and slowing down the optimization is also beneficial when the optimization includes many geometric restrictions and when the beginning mesh quality is poor.

   To optimize regions that are in contact, you might want to enter a negative value to reverse the direction of the optimization. As a result, areas of high stress will shrink and areas of low stress will grow.
4. Choose whether to update the optimization shape vectors after every optimization cycle (default) or only after the first cycle.

   The Optimization module determines an optimization displacement vector for every node in the design area. The vector lies along the normal to the outer surface at the node and indicates the direction of displacement during the optimization. If you choose to update the optimization shape vectors after every optimization cycle, the Optimization module adjusts the vectors to account for changing conditions, such as changes in the shape of the structure, the mesh quality, and design variable restrictions. If you choose to update the optimization shape vectors only after the first optimization cycle, the vectors remain fixed in subsequent cycles.

   In most cases, the default value of updating the optimization shape vectors after every optimization cycle provides better results because the mesh smoothing algorithm is less restricted, resulting in an improved mesh quality.
5. Choose whether the step size should be determined by the minimum displacement of the nodes in the design area during the optimization or the average displacement.

   The Optimization module examines your mesh and limits the amount of displacement of the nodes in the design area during each optimization cycle. This limit prevents the large displacement of one node from causing the collapse of a neighboring element. In addition, the condition-based optimization algorithm provides control of the displacement of the nodes in the design area after every design cycle—the step size. The step size depends on the limit that the Optimization module has applied to the nodes. For example, if the Optimization module decreases the allowed displacement, the condition-based optimization algorithm decreases the increment size.

This option allows you to choose which displacement is used by the condition-based optimization algorithm to determine the step size. You can choose the average value of the allowed displacement of the nodes in the design area during the optimization or the minimum value (default). Selecting the average value results in a larger step size and a faster calculation of the optimum solution. However, selecting the average value can result in limited displacement of nodes for which only small displacements are allowed causing undesirable corners in the design area.

6. Choose the method that the Optimization module will use to interpolate the midside nodes.

   If you select **Linearly by position** (default), the optimization linearly interpolates the position of the midside node from the optimized position of the connected corner nodes. If you select **By optimization displacement of corner nodes**, the optimization interpolates the position of the midside node from the optimization displacement of the connected corner nodes.

   If the nodes are in their original position, the midside node sits on the line between the corner nodes and there is no difference between the two interpolation methods. However, to prevent element bending, you must select **By optimization displacement of corner nodes**.

7. If desired, toggle on **Edge length for movement vector** and enter a value.

   The Optimization module modifies the displacement of nodes in areas of high curvature to prevent the mesh from collapsing because of a large volume change. In effect, sharp corners are smoothed out. The default value of the minimum element edge length that triggers smoothing is 5.0. A larger value results in a larger radius for the smoothed region.

8. The Optimization module can use a filter to smooth out local stress peaks. You can define the filter function by toggling on **Max. influence radius for equivalent stress** and entering the following:

   - A value for the maximum distance between nodes that are influenced by the filter.
   - A value that determines how much the local surface curvature will be used to adjust the maximum distance between nodes that are influenced by the filter. The default value is 0.2; a smaller value increases the effect of the surface curvature.
   - A weighting value that controls the effect of the filter depending on the distance from the node.

9. Volume is the only constraint you can apply to a shape optimization, and you can specify that the volume be reduced to a specified value or to a fraction of the initial value. The **Equality constraint tolerance** specifies the minimum difference between the specified volume constraint and the calculated volume that results in the Optimization module assuming the solution has converged. The Optimization module compares the absolute value of the difference with the tolerance value you enter. The default value is 0.001.
10. If you selected **General optimization (sensitivity-based)**, you can select **Use Abaqus sensitivities when possible** to use Abaqus to compute the design responses and their sensitivities whenever possible. This workflow modification improves the optimization process performance.
11. If you selected **General optimization (sensitivity-based)**, you can choose **Use Group Operator when possible** to use large groups of more than 5000 elements or nodes in the design response definition in an efficient way. This workflow modification uses a new algorithm based on Abaqus sensitivities.

Configuring durability options

Typically, you use shape optimization to modify the surface geometry of a component to minimize stress concentrations. In most cases reducing the stress levels leads to a significant increase in durability. However, it is possible that the regions of peak stress identified by a static analysis differ from the regions of maximum damage identified from a durability (or damage) analysis and using shape optimization alone to modify the surface geometry might decrease the durability. To avoid this situation, you can incorporate a durability solver in the optimization loop to ensure that you are both reducing stress levels and increasing durability.

To include durability in your optimization, you must enable durability analysis in the optimization task editor and configure the selected durability solver. In addition, you must create a damage design response that will be used as an objective. The objective must attempt to minimize the maximum damage in the critical areas.

1. In the optimization task editor, click the **Durability** tab.
2. Select **Optimize based on durability analysis**.
3. From the **Durability solver** options, select one of the following:

fe-safe	Uses the **fe-safe** durability solver.
FEMFAT	Uses the **FEMFAT** durability solver.
Custom	Allows you to use a Tosca Structure optimization neutral file generated by a durability analysis (ONF 600 or ONF 601). Contact your SIMULIA support office for more information about the format of this file.

4. Select the durability input files that will be read by the durability solver. The durability input files must be located in the working directory.
5. Enter the name of any additional files in the working directory that will be read by the durability solver. If a file is stored outside the working directory, you must provide the path to the file along with the name of the file.

Configuring a sizing optimization task

A sizing optimization is a flexible, sensitivity-based optimization that allows you to apply a range of constraints and objective functions to your model. You use the optimization task editor to customize various aspects of a sizing optimization.
To locate the editor, select **Task**->**Edit**->***optimization task name*** from the main menu bar.

In this section:

- *Configuring basic settings*
- *Configuring the thickness settings*
- *Configuring the perturbation settings*
- *Configuring convergence options*
- *Configuring advanced options*

Configuring basic settings

You can configure a sizing optimization task.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to freeze load or boundary condition regions.

   It is recommended that you freeze regions to which prescribed conditions are applied because you do not want these regions to be removed during the optimization. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.

Configuring the thickness settings

You can configure a sizing optimization task.

1. In the optimization task editor, click the **Thickness** tab.
2. Select the **Thickness update strategy**.

   This setting controls the rate at which the Optimization module updates the shell thickness of design elements during the optimization using the method of moving asymptotes. In most cases you should accept the default setting (**Normal**). However, if the design responses are very sensitive and you have problems fulfilling the constraints, you may need a more conservative rate that requires more optimization iterations. Selecting an aggressive rate may lead to unstable optimization or prevent the optimization from converging on a solution.
3. Enter the **Maximum change per design cycle**.

   This setting controls the limit on the change in shell element thickness during each design cycle.

Configuring the perturbation settings

You can configure a sizing optimization task.

1. In the optimization task editor, click the **Perturbation** tab.
2. Enter the number of eigenmodes to track. The default value is five, which means that the Optimization module tracks the five lowest eigenfrequencies.

   In some cases many local low frequency eigenmodes appear during the optimization iterations, which leads to a high number of modes to track and degrades performance. You can avoid tracking a high number of modes by choosing the lower bound of the eigenfrequencies to be 25% of the eigenfrequency of interest in the first optimization iteration.

   Mode tracking is not required if your design response will use the Kreisselmeier-Steinhauser formulation to evaluate the eigenfrequencies. Your Abaqus model must include an output request for at least the number of eigenfrequencies you are tracking.
3. Select the region over which the Optimization module should track the eigenmodes.

Configuring convergence options

You can configure a sizing optimization task.

1. In the optimization task editor, click the **Convergence** tab.
2. Specify the **Convergence Criteria**. The following options allow you to specify the convergence criteria for a sizing optimization:

Specifying when to start checking for convergence

You can specify the iteration during which the Optimization module will begin to check the two convergence criteria. The optimization will always continue at least until this value has been reached. The default value is 4.

Specifying which convergence criterion to check

You can specify whether the optimization should end when either of the convergence criterion has been fulfilled or both of the criteria have been fulfilled. The default value is that both criteria must be fulfilled.

Convergence based on the change in optimization function

You can specify that the optimization will end based on the change in the objective function from one iteration to the next. The default value is 0.001.

Convergence based on the change in element thickness

Element thickness is the design variable for a sizing optimization. You can specify that the optimization will end based on the average change in the element thickness from one iteration to the next. The default value is 0.005.

Configuring advanced options

You can configure a sizing optimization task.

1. In the optimization task editor, click the **Advanced** tab.
2. Select **Use Abaqus sensitivities when possible** to use Abaqus to compute the design responses and their sensitivities whenever possible. This workflow modification improves the optimization process performance.
3. You can choose **Use Group Operator when possible** to use large groups of more than 5000 elements or nodes in the design response definition in an efficient way. This workflow modification uses a new algorithm based on Abaqus sensitivities.

Configuring a bead optimization task

The Optimization module provides a variety of settings that allow you to configure a bead optimization task. The configuration settings depend on whether you are configuring an optimization task for a condition-based bead optimization (default) or for a general bead optimization.

In this section:

- *Configuring a condition-based bead optimization task*
- *Configuring a general bead optimization task*

Configuring a condition-based bead optimization task

A condition-based bead optimization is based upon a special bending hypothesis and uses special filters to generate beads along the bending trajectories. You use the optimization task editor to customize various aspects of a condition-based bead optimization.
To locate the editor, select **Task->Edit->*optimization task name*** from the main menu bar. To specify a condition-based bead optimization, select the **Advanced** tab and choose **Condition-based optimization**.

In this section:

- *Configuring basic settings*
- *Configuring advanced options*

Configuring basic settings

You can configure a condition-based bead optimization task.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to respect boundary conditions that have been applied to the model.

   It is recommended that you freeze regions to which boundary conditions are applied because you do not want these regions to be moved during the optimization. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.
3. By default, the Optimization module freezes boundary conditions for the whole model. If desired, click  and select the region in which the boundary conditions should be frozen.

Configuring advanced options

You can configure a condition-based bead optimization task.

1. In the optimization task editor, click the **Advanced** tab.
2. Toggle on **Growth direction opposite to shell normal** to reverse the direction of the nodal displacement that forms the bead.
3. By default, the direction in which the nodes are moved to form the bead is determined from the stress state of the model at the start of the optimization—the first cycle. Alternatively, you can specify that the optimization determines the direction in which the nodes are moved after every design cycle.
4. By default, an internally computed value will be used for the bead width. Alternatively, you can specify the absolute value of the width of the bead.
5. Specify the number of iterations the bead optimization will perform. The number of iterations modifies the step size of the optimization. The default value is 2.
6. Specify the following **Penalty Conditions**:

Minimum stress ratio

Enter the value of the minimum von Mises stress ratio to prevent Abaqus/CAE from optimizing regions with very low stresses. Abaqus/CAE does not apply bead optimization in the regions where the von Mises stress is less than the value computed from the specified ratio multiplied by the highest von Mises stress in the design area ($0.0 < \textit{Minimum stress ratio} < 1.0$). The default value is 0.001.

Maximum membrane stress ratio

Enter the value of the maximum membrane stress ratio to prevent Abaqus/CAE from optimizing regions in a predominately membrane, or inplane, stress state (the introduction of a bead in a region under a predominately membrane stress state may make the structure softer).

Abaqus/CAE does not apply bead optimization in regions where the membrane stress is greater than the constant value computed from the maximum bending stress in the original model divided by the specified ratio ($0.0 < \textit{Maximum membrane stress ratio}$). The default value is 1.0.

7. Specify the following **Mesh Smoothing Parameters**:

Curve smooth

Enter the relative value of the radius defining a region of high curvature.

The introduction of a bead during the optimization can squeeze nodes together and result in small elements. For even higher degrees of curvature and large bead heights, the nodes can begin to overlap causing the analysis to fail. To prevent the collapse of the mesh, Abaqus/CAE can modify how it moves nodes while creating a bead in a region of high curvature. High curvature is defined as the radius calculated by multiplying the **Curve smooth** value and the average element edge length in the design area. High values of **Curve smooth** and, hence, a large radius encompassing many elements, can be computationaly expensive. The default value is five times the average element length.

Node smooth

By default, the value for **Node smooth** is $0.25 \times \textit{bead width}$. Alternatively, you can specify the absolute minimum in-plane distance between neighboring nodes during the creation of a bead. Values between 0.0 and $0.5 \times \textit{bead width}$ are allowed.

Node smoothing is applied to prevent sudden changes in displacement of neighboring nodes, especially near the boundary between the design area and the rest of the model or where active design variable constraints are restraining the displacement of nodes.

Configuring a general bead optimization task

A general bead optimization is a flexible, sensitivity-based optimization that allows you to apply a range of constraints and objective functions to your model. The sensitivity-based algorithm does not implement a bead filter; therefore, the optimization may not generate a distinct bead pattern. You use the optimization task editor to customize various aspects of a bead optimization.
To locate the editor, select **Task**->**Edit**->***optimization task name*** from the main menu bar. To specify a general bead optimization, select the **Advanced** tab and choose **General optimization (sensitivity-based)**.

In this section:

- *Configuring basic settings*
- *Configuring the nodal move settings*
- *Configuring the perturbation settings*
- *Configuring advanced options*

Configuring basic settings

You can configure a general bead optimization task.

1. In the optimization task editor, click the **Basic** tab.
2. Choose whether to respect boundary conditions that have been applied to the model.

   It is recommended that you freeze regions to which boundary conditions are applied because you do not want these regions to be moved during the optimization. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.
3. By default, the Optimization module freezes boundary conditions for the whole model. If desired, click  and select the region in which the boundary conditions should be frozen.

Configuring the nodal move settings

You can configure a general bead optimization task.

1. In the optimization task editor, click the **Nodal Move** tab.
2. Select the **Nodal update strategy**.

   This setting controls the rate at which the Optimization module updates the shell thickness of design elements during the optimization using the method of moving asymptotes. In most cases you should accept the default setting (**Conservative**). However, if the design responses are very sensitive and you have problems fulfilling the constraints, you may need a more aggressive rate that requires more optimization iterations. Selecting an aggressive rate may lead to unstable optimization or prevent the optimization from converging on a solution.
3. Enter the **Nodal move limit**.

   This setting limits the nodal displacement per iteration relative to the maximum displacement prescribed ($0.0 < \textit{Nodal move limit} < 1.0$). The default value is 0.1. If you encounter a difficult optimization problem that is slow to converge, you can reduce the size of each optimization iteration by reducing the **Nodal move limit**.

   Note:

   You prescribe the maximum displacement of a node by specifying the height of the stiffening bead when you create a bead optimization constraint.

Configuring the perturbation settings

You can configure a general bead optimization task.

1. In the optimization task editor, click the **Perturbation** tab.
2. Enter the number of eigenmodes to track. The default value is five, which means that the Optimization module tracks the five lowest eigenfrequencies.

   In some cases many local low frequency eigenmodes appear during the optimization iterations, which leads to a high number of modes to track and degrades performance. You can avoid tracking a high number of modes by choosing the lower bound of the eigenfrequencies to be 25% of the eigenfrequency of interest in the first optimization iteration.

   Mode tracking is not required if your design response will use the Kreisselmeier-Steinhauser formulation to evaluate the eigenfrequencies. Your Abaqus model must include an output request for at least the number of eigenfrequencies you are tracking.
3. Select the region over which the Optimization module should track the eigenmodes. Selecting a region other than the whole model may result in increased performance.

Configuring advanced options

You can configure a general bead optimization task.

1. In the optimization task editor, click the **Advanced** tab.
2. Specify the **Filter Radius** of the Sigmund filter, which smoothes the resulting optimization solution. Changing this value might help you to avoid known problems from fluctuations in sensitivity values. The following options allow you to specify the filter radius for a general bead optimization:

   ### Relative to average edge length

   Enter the relative filter radius. Abaqus/CAE computes the filter radius as the value specified multiplied by the average element length in the design area (the region that will be optimized). The default value is four. A value of zero turns off the filter; the resulting bead optimization might generate an unsmooth result that is numerically optimal but not a realistic physical solution.

   ### Absolute value

   Enter the absolute value of the radius.

3. For the sensitivity calculation, choose whether the optimization should consider only the nodes in the region that will be optimized (default) or all the nodes in the model.
4. Enter the **Bead perturbation**. Abaqus/CAE uses this value to compute a semianalytical sensitivity analysis using a finite difference on the element matrices. The finite difference is computed as the perturbation value specified multiplied by the average element edge length. The default value is 0.0001, which is suitable for most bead optimization problems.
5. Select **Use Abaqus sensitivities when possible** to use Abaqus to compute the design responses and their sensitivities whenever possible. This workflow modification improves the optimization process performance.
6. You can choose **Use Group Operator when possible** to use large groups of more than 5000 elements or nodes in the design response definition in an efficient way. This workflow modification uses a new algorithm based on Abaqus sensitivities.

Configuring design responses

This section describes the design response editor and the options that appear in the design response editor.

In this section:

- *Creating and editing a design response*
- *Selecting the data source of a design response*
- *Combining design responses*

Creating and editing a design response

You use the design response editor to create and configure your design responses. You associate a design response with a region of your model; however, the design response consists of a single scalar value, such as the maximum stress in the region or the total volume of the model. In addition, you can associate a design response with a particular step or a load case within a step. Design responses are used by objective functions and constraints. The design responses that are available depend on the type of optimization task you created; for more information, see *Design Responses*.

A design response can cover multiple models. You can incorporate multiple models into your optimization when linear perturbations about a base state are no longer sufficient as load cases. For example, you can simulate nonlinear load cases (which are not supported by Abaqus/CAE) by creating multiple copies of your nonlinear model and by creating a step in each model during which different loads and boundary conditions are applied. Each model must have the same mesh and the same section assignments, and the models must be open in your Abaqus/CAE session.

1. From the main menu bar, select **Design Response->Create**.

   The **Create Design Response** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Design Response Manager**. (You can display the **Design Response Manager** by selecting **Design Response->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the prompt area, select the region to which the design response will be applied:

   - Select **Whole Model** (default) to apply the design response to the entire model.
   - Select **Body (elements)**, and select the region to which the design response will be applied. During the optimization, the design response will be applied to the elements in the selected region.
   - Select **Point (nodes)**, and select the region to which the design response will be applied. During the optimization, the design response will be applied to the nodes in the selected region.

   By default, Abaqus/CAE allows you to select all regions of the model. Use the **Selection** toolbar to change the type of object that you can select to **Vertices**, **Edges**, **Faces**, or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click the button—**Select in Viewport** or **Sets**—on the right side of the prompt area.

3. When you have finished selecting the design response region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Design Response** dialog box appears.

4. By default, the Optimization module assumes a design response, such as a displacement along an axis, is defined in the global coordinate system. To change the coordinate system in which the design response is defined, click  for the **CSYS** option and do one of the following:

   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   Alternatively, click  to define a new datum coordinate system.
5. From the **Edit Design Response** dialog box, select the **Variable** tabbed page.
6. Select the variable and, if applicable, its component.

   Note:

   By default, the Optimization module displays all of the variables that are available for the selected optimization task. To avoid creating a design response that cannot be used as expected, you can display only the variables that are available for an objective function or only the variables that are available for a constraint.
7. If you are creating an internal force or internal moment design response, you must select the nodal subset region. This region contains the nodes that form a cross section of the elements in the region in which the internal force or moment will be maximized or minimized.
8. Select the **Steps** tabbed page, and specify the model and step or load case of interest. In addition, if you are performing an eigenfrequency optimization, you must select the modes of interest from the **Steps** tabbed page. For more information, see *Selecting the data source of a design response*.
9. Select the operator that will be applied on the selected variable across the design area.

   - **Sum of values**: The sum of the values across the design area. For some variables (such as volume, weight, moment of inertia, and gravity) **Sum of values** is the only operator, and it is selected by default.
   - **Minimum value**: The minimum value across the design area.
   - **Maximum value**: The maximum value across the design area. For some variables (such as stress, contact stress, and strain) **Maximum value** is the only operator, and it is selected by default.
10. If applicable, select the operator that will be applied on the selected variable across steps and load cases.

    - **Sum of values**: The sum of the values across the selected steps or load cases.
    - **Minimum value**: The minimum value across the selected steps or load cases.
    - **Maximum value**: The maximum value across the selected steps or load cases.
11. Click **OK** to save your data and to exit the editor.

Additional information

- *Selecting the data source of a design response*

- *Combining design responses*

Selecting the data source of a design response

By default, the Optimization module uses the data from the last load case, if any, and the last step of your Abaqus model to define the design response. Alternatively, if your model contains multiple steps or load conditions, you can select which steps or load conditions will be used to define the design response. Finally, if you have multiple models open in your session, you can select which models and which steps or load conditions within each model will be used to define the design response.

In addition, if the design response is calculating an eigenfrequency, you can choose which mode or range of modes to examine.

1. From the **Create Design Response** dialog box, select the **Steps** tabbed page.
2. Do either of the following:
   - Select **Use last step and last load case** to define the design response using data from the last step or last load case.
   - Select **Specify** to select the step or load condition that will be used to define the design response.
3. If you selected **Specify**, do the following to select the models, steps or load cases (within a step) that will be used to define the design response.
   - Select  to add a step from the current model to the list of steps.
   - Select  to add all of the valid steps from the current model to the list of steps.
   - Select  to add all of the valid steps from all of the open models in your session to the list of steps. (Abaqus/CAE displays an asterisk next to the name of the current model.)
   - Select  to delete a step from the list of steps.
4. If desired, click on a cell in the table, and select an alternate model, step, or load case from the menu that appears.
5. If the design response will calculate an eigenfrequency, you can select a mode or a range of modes from which the eigenfrequency will be calculated.
6. If the design region is a shell, you can select the location in the shell section from which the Optimization module will calculate the shell stresses. You can choose from the following:
   - The value of the shell stress at the top, middle, or bottom layer of the shell. (The middle layer experiences no bending and behaves as a membrane.)
   - The maximum value of the shell stress from the top, middle, or bottom layer of the shell.
   - The minimum value of the shell stress from the top, middle, or bottom layer of the shell.
7. Select how the Optimization module will extract the design response from the selected region. You can choose from the following:
   - The maximum value from the selected region.
   - The minimum value from the selected region.
   - The sum of the values from the selected region.
8. Select how the Optimization module will extract the design response from the steps and load cases. You can choose from the following:

- The maximum value from all the selected steps and load cases.
- The minimum value from all of the selected steps and load cases.
- The sum of the values from all of the selected steps and load cases.

9. Click **OK** to save your data and to exit the editor.

Additional information

• *Creating and editing a design response*
• *Combining design responses*

Combining design responses

The simplest approach for combining design responses is to create an objective function with a weighted sum of design responses. Alternatively, you can combine design responses using the design response editor.
You should be careful how you combine design responses to avoid creating a meaningless optimization task. In addition, you should understand how condition-based and general optimization combine terms and how they differ.

In this section:

- *Combining design responses for a condition-based topology or shape optimization*
- *Combining design responses for a general topology optimization*
- *Filtering a design response for a shape optimization*
- *Applying a cutoff filter to a design response for a shape optimization*
- *Normalizing a design response for a shape optimization*

Combining design responses for a condition-based topology or shape optimization

For both condition-based topology and shape optimization the following methods are available for combining up to four design responses (R_1, R_2, R_3, and R_4):

Add	$R_1 + R_2 + R_3 + R_4$
Multiply	$R_1 \times R_2 \times R_3 \times R_4$
Minimum	$Min\,(R_1, R_2, R_3, R_4)$
Maximum	$Max\,(R_1, R_2, R_3, R_4)$

For both condition-based topology and shape optimization the following methods are available for combining two design responses (R_1 and R_2):

Subtract	$R_1 - R_2$
Divide	$\frac{R_1}{R_2}$

For both condition-based topology and shape optimization the following methods are available for operating on a single design response:

Absolute value	$abs\,(R_1)$
Sine	$Sin\,(R_1)$
Cosine	$Cos\,(R_1)$
Tangent	$Tan\,(R_1)$
Common logarithm	$Log\,(R_1)$
Natural logarithm	$Ln\,(R_1)$
Square root	$\sqrt{R_1}$
Exponential	$exp\,(R_1)$
N[th] Root	$\sqrt[n]{R_1}$
N[th] Power	${R_1}^n$
Integer	$Int\,(R_1)$
Nearest integer number	$Nint\,(R_1)$
Sign	$Sign\,(R_1)$
Δ_{ij}, the difference between two iterations	$R_1 i - R_1 j$

Additional information

- *Creating and editing a design response*
- *Selecting the data source of a design response*

Combining design responses for a general topology optimization

For a general topology optimization you can create a design response calculated as the difference or the absolute difference between two design responses of the same type or a weighted combination of several (up to 10) displacement design responses.

A typical example uses the absolute difference in the displacement between two design responses to constrain the displacement of two vertices relative to each other. The following table shows which design responses can be combined:

Design responses	Absolute difference	Weighted sum	Subtract
Displacements and rotation	√	√	√
Absolute displacements and rotations			
Reaction forces	√	√	√
Absolute reaction forces			
Internal forces	√	√	√
Absolute internal forces			
Modal eigenfrequencies	√		√

Although a design response is a single scalar value, you can use the appropriate weighting to combine design responses that are defined in different directions or in different coordinate systems.

Additional information

- *Creating and editing a design response*
- *Selecting the data source of a design response*

Filtering a design response for a shape optimization

You can apply a filter to a design response that will smooth out local peaks for a shape optimization. The filter is defined as

$$\Phi_j = \frac{\sum_{i-1}^{N} \Phi_i B_{ij}}{\sum_{i-1}^{N} B_{ij}},$$

$$B_{ij} = (r_j - d_{ji})^p,$$

$$r_j = r_{\max} e^{-0.5(k_{\max} \int \sigma)},$$

$$k_{\max} = \max(|n_j \times n_k|),$$

where $\Phi_j$is the filter function for node *j*. The main filter function, B_{ij}, decreases with the distance, *d*, between nodes *i* and *j*. The maximum radius of influence, r_{max}, defines the maximum distance from node *j* for the nodes *i* to influence the filter value. The local curvature, κ_{max}, is approximated by the vector product of the node normal, n_j, and the neighboring nodes, n_k. The curvature dependency, r_j, defines a weight function that reduces the radius at higher local curvature.

You can specify the following:

- The maximum radius of influence, r_{max}.
- The exponent, *p*, which defines the weight function that controls the influence of the distance between nodes *i* and *j*. The default value is 1.0.
- The surface bending radius reduction, σ. The default value of σ is 0.2. A larger value of σ decreases the influence of the surface curvature.

Additional information

• *Creating and editing a design response*
• *Selecting the data source of a design response*

Applying a cutoff filter to a design response for a shape optimization

You can apply a filter that cuts off local peaks of the design response for a shape optimization. You can specify the following:

- The lower bound of the design response. All values less than this lower bound are assumed to be zero.
- The upper bound of the design response. All values greater than this upper bound are assumed to be the value of the upper bound.

Additional information

- *Creating and editing a design response*
- *Selecting the data source of a design response*

Normalizing a design response for a shape optimization

You can normalize the vectors used as terms for the algorithm that calculates a shape optimization. You may want to normalize design responses before you combine them using a weighted combination; for example, if different loads are applied to different regions. If desired, you can apply either of the following nomalizations:

- A normalization based on the maximum value in each design cycle (in effect, a normalization of 1.0 for each cycle).
- A normalization based on the initial value of the design response (the maximum value during the first design cycle).

Additional information

- *Creating and editing a design response*
- *Selecting the data source of a design response*

Creating objective functions

An objective function defines the objective of your optimization. To configure an objective function, you select a design response or a combination of design responses and specify the target; for example, to minimize the total strain energy in the model.

1. From the main menu bar, select **Objective Function->Create**.

   The **Create Objective Function** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Objective Function Manager**. (You can display the **Objective Function Manager** by selecting **Objective Function->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Objective Function** dialog box that appears, enter the name of the objective function.

   The **Edit Objective Function** dialog box appears.

3. Select the **Target** of the objective function. You can select one of the following:

   - **Minimize design response values**: create an optimized model that tries to minimize the sum of the weighted differences between the design response and the reference value.
   - **Maximize design response values**: create an optimized model that tries to maximize the sum of the weighted differences between the design response and the reference value.
   - **Minimize the maximum design response values**: find the maximum weighted difference between the design response and the reference value and create an optimized model that minimizes that maximum difference.

4. From the menu in the **Name** column, select the design response that will be used to define this objective function.

   The objective function editor displays the type of the design response along with its default weight and reference value.

5. To create an objective function that is a weighted sum of multiple design responses, choose from the following:

   - Select to add a design response to the list of design responses.
   - Select to add all of the valid design responses to the list of design responses.
   - Select to delete a design response from the list of design responses.
   - Select to create a new design response and add it to the list of design responses.

6. If desired, click on the weight or reference value of an objective function and enter a new value.
7. Click **OK** to save your data and to exit the editor.

Creating constraints

A constraint constrains the Optimization module from making changes to the topology of the model during the optimization. If a constraint cannot be satisfied, the optimization is not feasible. To configure a constraint, you select a design response or a combination of design responses and specify the value of the constraint; for example, the volume of the optimized model can be constrained to be 50% of the original volume.

When you execute an optimization process, Abaqus generates history output from the constraints you defined in the Optimization module. For volume design responses the history output is always reported as a fraction of the initial value. For all other design responses the history output is reported as an absolute value.

1. From the main menu bar, select **Constraint->Create**.

   The **Create Constraint** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Constraint Manager**. (You can display the **Constraint Manager** by selecting **Constraint->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Constraint** dialog box that appears, enter the name of the constraint.

   The **Edit Constraint** dialog box appears.

3. From the menu, select the design response that will be used to define this constraint. Alternatively, you can select to create a new design response and add it to the list of design responses.

   The constraint editor displays the type of the design response, which is the variable you are constraining. If you are performing a condition-based optimization, you must select a volume constraint.

4. Define the constraint. The editor provides different options, depending on the type of optimization task you selected.

   - If you are defining a constraint for a condition-based topology or shape optimization, the volume constraint can be either of the following:
     - Equal to a fixed value.
     - Equal to a fraction of the value before the optimization starts.
   - If you are defining a constraint for a general topology optimization, the constraint can be one of the following:
     - Less than, greater than, or equal to a fixed value.
     - Less than, greater than, or equal to a fraction of the value before the optimization starts.

     Note:

     During a condition-based topology or shape optimization, the volume is reduced incrementally until the volume constraint is satisfied during the last step. In contrast, a general topology optimization satisfies the volume constraint in the first design cycle. As a consequence, a general topology optimization starts with a much weaker structure, and the element density is scaled to satisfy the remaining constraints and objectives.

5. Click **OK** to save your data and to exit the editor.

Additional information

• *About constraints*

Configuring geometric restrictions

This section describes the geometric restriction editor and the options that appear in the geometric restriction editor.

In this section:

- *Creating and editing geometric restrictions*
- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*
- *Creating a geometric restriction in a bead optimization*

Creating and editing geometric restrictions

You can create geometric restrictions that place boundaries on the changes that the Optimization module can make to the topology of the model.

Geometrical restrictions include frozen regions from which material cannot be removed and manufacturing constraints, such as restrictions on undercuts that would prevent the optimized model from being removed from a mold.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select the type of geometric restriction to apply, and click **Continue**.
4. From the viewport, select the region to which the geometric restriction will be applied, or click **Done** to apply the geometric restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*.

   The **Edit Geometric Restriction** dialog box appears. The content of the dialog box depends on the type of geometric restriction you are creating.

6. Click **OK** to save your data and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a geometric restriction in a topology or sizing optimization

This section describes how to create a geometric restriction in a topology or sizing optimization.

In this section:

- *Creating a frozen area restriction*
- *Creating a member size restriction*
- *Creating a cyclic symmetry restriction*
- *Creating a planar symmetry restriction*
- *Creating a rotational symmetry restriction*
- *Creating a point symmetry restriction*
- *Creating a minimum width restriction*
- *Creating a thickness control restriction*
- *Creating a clustering restriction*
- *Creating a demold restriction*
- *Creating an overhang control restriction*
- *Creating a milling control restriction*
- *Creating a rib design restriction*

Creating a frozen area restriction

You can select regions of your model that will not be modified during a topology or sizing optimization. The regions continue to contribute to variables such as the weight and the inertia of the model. You can freeze regions that are contacting other parts, such as regions that are used to support the model and regions that form bearing surfaces. It is recommended that you freeze regions to which prescribed conditions are applied. You can freeze these regions using a geometric restriction; or you can request that the Optimization module freeze them automatically when you create a topology optimization task, as described in *Creating an optimization task*. Freezing these regions stabilizes the optimization and often leads to a significantly lower number of iterations.

1. From the main menu bar, select **Geometric Restriction->Create**. The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Frozen area** or **Frozen area (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the regions that will be frozen.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the region to freeze, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Click **OK** to freeze the selected regions and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a member size restriction

For a topology optimization you can specify the minimum or maximum size of selected regions of your model; for example, the minimum size of a region in which the optimization creates trusses.
You can also specify the minimum distance between trusses. Specifying a maximum member size forces the optimization to split thick regions into several smaller regions and prevents it from creating large contiguous regions that might be difficult to cast. Specifying a minimum member size avoids thin trusses that may be difficult to manufacture. The minimum member size must be greater than twice the average element size to prevent the results from being dependent on the mesh size.

In most cases you can specify the same value for both the minimum and maximum member size, and the Optimization module creates trusses that are approximately equal to the specified value. To prevent the structure from collapsing, the Optimization module tries to avoid creating thin trusses in regions where prescribed conditions are applied.

Specifying the minimum or maximum member size is very computationally expensive and should be used only where necessary. You should perform an optimization without member size restrictions to identify any regions where the restrictions should be applied.



Note: You can combine a member size constraint and a demold constraint only for a general topology optimization.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can start the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Member size (Topology)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region to which the member size restriction will be applied or click **Done** to apply the member size restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   To select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Do one of the following:
   - Select **Minimum thickness**, and enter the minimum member thickness.
   - Select **Maximum thickness**, and enter the maximum member thickness.
   - Select **Envelope**, and enter the following:
     - The **Minimum** member thickness.
     - The **Maximum** member thickness.
     - The **Minimum gap** between members.
7. Select a **Method** algorithm: **Filter** or **Local Volume** when the restriction type is **Maximum thickness** or **Envelope**.
8. Click **OK** to create the frozen area geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a cyclic symmetry restriction

You can specify a cyclic symmetry geometric restriction for a topology or sizing optimization.

A cyclic symmetry geometric restriction replicates a selected region over a specified distance, as shown in *Figure 1*.



Figure 1: Cyclic symmetry from a topology optimization.

The selected region is replicated along an axis of a coordinate system. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information). You can choose to remove frozen regions from the symmetry restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Cyclic symmetry** or **Cyclic symmetry (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region that will be replicated by the cyclic symmetry or click **Done** to apply the cyclic symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system, and select the axis along which the region will be replicated.
7. Enter the distance that the selected region will be translated along the axis.
8. If desired, toggle on **Ignore frozen area** to remove any frozen regions from the region that will be translated.
9. Click **OK** to create the cyclic symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a planar symmetry restriction

You can specify a planar symmetry geometric restriction for a topology or sizing optimization.

A planar symmetry geometric restriction forces the optimized model to be symmetric about a specified plane, as shown in *Figure 1*:

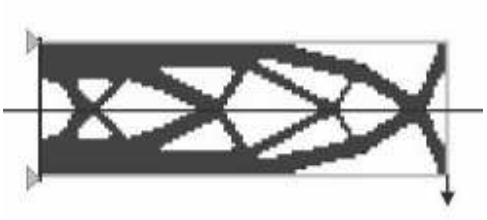


Figure 1: Planar symmetry from a topology optimization.

You specify the plane of symmetry by selecting the axis of a coordinate system that is the normal to the plane of symmetry. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information). You can choose to remove frozen regions from the symmetry restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Planar symmetry** or **Planar symmetry (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the planar symmetry will be enforced or click **Done** to apply the planar symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system, and select the axis of the coordinate system that represents the normal to the plane of symmetry.
7. If desired, toggle on **Ignore frozen area** to remove any frozen regions from the symmetry restriction.
8. Click **OK** to create the planar symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a rotational symmetry restriction

You can specify a rotational symmetry geometric restriction for a topology or sizing optimization.

A rotational symmetry geometric restriction forces the optimized model to be symmetric about a specified axis, as shown in *Figure 1*:



Figure 1: Rotational symmetry from a topology optimization.

You specify the axis of symmetry by selecting the axis of a coordinate system. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information). In addition, you must enter an angle that specifies the size of the repeating segment. You can choose to remove frozen regions from the symmetry restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Rotational symmetry** or **Rotational symmetry (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the rotational symmetry will be enforced or click **Done** to apply the rotational symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.
6. Select the coordinate system, and select the axis of the coordinate system that represents the axis of symmetry.
7. Enter the **Repeating segment size**, which is the angle (in degrees) specifying the size of the repeating segment. The value must be greater than or equal to 2°.
8. If desired, toggle on **Ignore frozen area** to remove any frozen regions from the symmetry restriction.
9. Click **OK** to create the rotational symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a point symmetry restriction

You can specify a point symmetry geometric restriction for a topology or sizing optimization.

A point symmetry geometric restriction forces the optimized model to be symmetric about a specified point, as shown in *Figure 1*.



Figure 1: Point symmetry from a topology optimization.

You specify the point of symmetry by selecting a coordinate system (the point of symmetry is assumed to be the origin of the coordinate system). You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information). You can choose to remove frozen regions from the symmetry restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Point symmetry** or **Point symmetry (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the point symmetry will be enforced or click **Done** to apply the point symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system. The point of symmetry is assumed to be the origin of the selected coordinate system.
7. If desired, toggle on **Ignore frozen area** to remove any frozen regions from the symmetry restriction.
8. Click **OK** to create the point symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a minimum width restriction

A sizing optimization determines the optimal element thickness when modeling a sheet metal structure with shell elements. Specifying the minimum width of a region containing elements of the same thickness prevents narrow slivers of elements with equal thickness from appearing in the solution after a sizing optimization. Specifying a minimum width will also prevent oscillations in the shell thickness or a "checkerboard" pattern of element thicknesses. The minimum width must be larger than the average length of the element edges.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Member size (Sizing)** from the list of geometric restrictions and click **Continue**.
4. From the viewport, select the region to which the minimum width restriction will be applied or click **Done** to apply the minimum width restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select **Minimum width** and enter the minimum width.
7. Click **OK** to create the minimum width geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a thickness control restriction

You can specify upper and lower bounds on the thickness of shell elements when you are configuring a sizing optimization. The value can be an absolute thickness or a fraction of the initial thickness.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Thickness control (Sizing)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region to which the thickness control restriction will be applied or click **Done** to apply the thickness control restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Do one of the following:

   - Select **Thickness by value** to enter the absolute value of the shell element thickness.
   - Select **Thickness by fraction** to enter a fraction of the shell element thickness relative to the initial value.

7. Select **Minimum** and enter the minimum thickness.
8. Select **Maximum** and enter the maximum thickness.
9. Click **OK** to create the thickness control geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*

- *Creating a geometric restriction in a shape optimization*

Creating a clustering restriction

You can specify that selected regions contain clusters of shell elements of equal thickness after a sizing optimization or clusters of elements of equal pseudodensities after a topology optimization.

You can use clustering to

- generate strengthening ribs or rings in the sheet metal structure you are optimizing,
- define borders between regions of equal thickness, or
- obtain an extrudable profile in the solid structures.

Clustered regions can be reproduced in manufacturing using sheets of constant thickness; for example, a vehicle “body in white” formed by welding and stamping individual sheet metal structures. To allow for maximum design flexibility, you should first optimize your structure without specifying clustering and use the initial design to decide which regions to cluster in your final optimization.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Cluster areas (Sizing)** or **Cluster areas (Topology)** from the list of geometric restrictions, and click **Continue**.

   The **Edit Geometric Restriction** dialog box appears.
4. From the list of **Eligible Sets**, select the sets specifying regions that will contain clusters of elements of equal thickness or of equal pseudo densities after the optimization.

   Note:

   If the desired set does not exist, click  to create it.
5. Click the arrow to move the selected sets to the list of **Selected Sets**.
6. Click **OK** to create the thickness clustering geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a demold restriction

You can specify a demold geometric restriction for a topology optimization. A demold geometric restriction forces the optimized model to satisfy specified manufacturing requirements; for example, it can prevent undercuts and hollow regions in a part that must be extracted from a mold.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Demold** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the demold restriction will be enforced or click **Done** to apply the demold symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. By default, the region in which the Optimization module checks for collisions is the same as the region in which it enforces the demold restriction. If desired, select from the top of the **Edit Geometric Restriction** dialog box, and select the collision check region. To avoid cavities, the collision check region should include the demold region and any cells that are adjacent to the demold region.
7. Choose one of the following demold techniques:

   - Choose **Demolding with a central plane** and select how the Optimization module will determine the central plane:
     - Choose **Determine automatically**. The Optimization module determines the optimal location of the central plane such that the optimization adds or removes material only in the pull direction

or in the opposite direction. Toggle on **Prevent hole formation** to prevent the optimization from removing material from the central plane.

- Choose **Specify**, and click  to select a point on the central plane. The Optimization module creates a structure that can be released from a mold in both pull directions away from the central plane.

- Choose **Forging (deform only in the pull direction)**. The central plane, the plane at which two halves of a mold meet, is assumed to lie at the back of the model and perpendicular to the pull direction. The optimization adds or removes material only in the pull direction.
- Choose **Stamping**. If the Optimization module decides to remove an element from the structure, it also removes all the elements behind or in front of the element (relative to the vector indicating the pull direction).
- Choose **Demolding at the region surface** to force the optimization to add or remove material only from the surface of the constrained region.

8. Click , and select two points specifying the vector along the pull direction.
9. Enter the **Draft angle**. The draft angle is a small angle relative to the pull direction that will be introduced to the optimized model to ensure the part can be removed from its mold. Normally the value is between 0° and 10°.
10. Click **OK** to create the demold geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating an overhang control restriction

You can specify a geometric restriction for overhang control in a topology optimization.

An overhang control geometric restriction forces the optimized model to satisfy specified additive manufacturing requirements. The restriction can prevent overhanging features in a model that exceed a critical angle. In general, an overhanging feature with an angle greater than 45° is difficult to print and usually requires the addition of support structures.

The base plane defines the printing bed. Only parts of the model situated above the base plane need support. By default, the overhang restriction tries to remove all overhanging features so that support structures are not required. Depending on the geometry, this can lead to unreasonable results. You can specify that there is no base plane, in which case boundary features are always printable. Alternatively, you can specify the base plane with a user-defined point in space on the plane perpendicular to the print direction. The point should be above overhanging features that are mandatory and under overhanging features that should be removed.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. Enter the name of the geometric restriction.
3. Select **Overhang control (Topology)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the overhang control restriction is enforced, or click **Done** to apply the overhang control restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*.

   The **Edit Geometric Restriction** dialog box appears.

6. By default, the region in which the Optimization module checks for critical overhanging features is the same as the region in which it enforces the overhang control restriction. If desired, select from the top of the **Edit Geometric Restriction** dialog box, and select the check region.
7. Select the base plane to define the printing bed.

- Select **Determine automatically** to allow the Optimization module to determine the optimal location of the base plane such that all overhanging features are removed.
- Select **Specify**, and click  to select a point on the plane perpendicular to the print direction.
- Select **None** to indicate that all boundary features should be printed.

8. By default, the Optimization module assumes the print direction is defined in the global coordinate system. To change the coordinate system, click  for the **CSYS** option and do one of the following:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   Alternatively, click  to define a new datum coordinate system.
9. Click , and select two points specifying the vector along the print direction.
10. Enter the **Overhang angle**. The overhang angle is an angle relative to the print direction that is introduced to the optimized model to ensure the part does not contain unacceptable features. The default value is 45°.
11. Enter the **Radius**. The radius defines the size of the cones that are used in the internal check for the overhang criteria. The default value in 1.5 times the average element length.
12. Click **OK** to create the overhang control geometric restriction and to exit the editor.

Additional information

• *Creating a geometric restriction in a topology or sizing optimization*

Creating a milling control restriction

You can specify a milling control geometric restriction for a topology optimization.
A milling geometric restriction forces the optimized model to satisfy specified manufacturing requirements; for example, it can arrange undercuts and hollow regions in a part to facilitate material removal by milling.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Milling control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the milling restriction will be enforced or click **Done** to apply the milling symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. By default, the region in which the Optimization module checks for milling is the same as the region in which it enforces the milling restriction. If desired, select  from the top of the **Edit Geometric Restriction** dialog box, and select the milling check region.
7. By default, the Optimization module assumes the milling directions are defined in the global coordinate system. To change the coordinate system, click  for the **CSYS** option and do one of the following:

   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.

     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.

- Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

Alternatively, click  to define a new datum coordinate system.

8. Click , and select two points specifying the vector along the milling direction. It specifies the direction in which the model is supposed to be milled. Repeat the step to specify multiple directions in case of multiaxis milling.
9. Enter the **Radius**. The radius value is used internally for the collision check during the removal of the elements.
10. Click **OK** to create the milling geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a rib design restriction

You can specify a rib design geometric restriction for a topology optimization.

The rib pattern restriction forces the optimizer to prevent material accumulation and to generate substructures in the form of ribs.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Rib design** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the rib design restriction will be enforced or click **Done** to apply the rib design restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

5. **Optional:** If you would rather select from a list of existing sets, do the following:
   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

6. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

7. **Optional:** By default, the region in which the Optimization module checks for rib design is the same as the region in which it enforces the rib design restriction. If desired, select from the top of the **Edit Geometric Restriction** dialog box, and select the rib design check region.
8. **Optional:** By default, the Optimization module assumes the rib direction is defined in the global coordinate system. To change the coordinate system, click for the **CSYS** option and do one of the following:

   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name:

     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.

2. Select a name from the list, and click **OK**.

- Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

Alternatively, click  to define a new datum coordinate system.

9. Click , and select two points specifying the out-of-plane growth direction of the ribs.
10. Enter the **Rib thickness**.

    The rib thickness is the averaged thickness of the ribs.
11. Enter the **Rib distance**.

    The rib distance is the averaged distance between the rib centers. The distance must be larger than twice the average element edge length.
12. Click **OK** to create the rib design geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a geometric restriction in a shape optimization

This section describes how to create a geometric restriction in a shape optimization.

In this section:

- *Creating a member size restriction*
- *Creating a planar symmetry restriction*
- *Creating a rotational symmetry restriction*
- *Creating a point symmetry restriction*
- *Creating a stamp control restriction*
- *Creating a turn control restriction*
- *Creating a demold control restriction*
- *Creating a drill control restriction*
- *Creating a penetration check restriction*
- *Creating a fixed region restriction*
- *Creating a growth restriction*
- *Creating a design direction restriction*
- *Creating a slide region control restriction*

Creating a member size restriction

You can specify the minimum or maximum size of selected regions of your model; for example, the minimum thickness of a rib that is modified during the shape optimization. The value that you enter can be thought of as a radius. During the optimization the Optimization module restrains the thickness of a member to be twice the value that you entered (the diameter). If you enter a value for the minimum thickness, the Optimization module adjusts surface nodes such that the minimum diameter must be achieved along a normal to the surface of the model. If the structure is smaller than the specified value in certain regions, the shape optimization permits only growth until the member size in the regions areas satisfies the restriction. Conversely, if you enter a value for the maximum thickness, the Optimization module adjusts surface nodes such that a maximum diameter must be achieved along a normal to the surface of the model. If the structure is larger than the specified value in certain regions, the shape optimization permits only shrinkage until the member size in the regions areas satisfies the restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create**. procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Member size (Shape)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region to which the member size restriction will be applied or click **Done** to apply the member size restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Do either of the following:

   - Select **Minimum thickness**, and enter the minimum radius of a member.
   - Select **Maximum thickness**, and enter the maximum radius of a member.

7. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the selected regions already include members larger than the minimum thickness. If the regions include smaller members, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle**, and the regions include smaller members, the Optimization module issues an error message and stops execution.
8. If desired, toggle on **Check node group**, and select the node group region.
9. Click **OK** to create the member size geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a planar symmetry restriction

You can specify a planar symmetry geometric restriction for a shape optimization. A planar symmetry geometric restriction forces selected faces of the optimized model to be symmetric about a specified plane. You specify the plane of symmetry by selecting the axis of a coordinate system that is the normal to the plane of symmetry. The origin of the coordinate system is a point on the plane of symmetry. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh must be approximately symmetric across the plane of symmetry before the optimization starts somast that the Optimization module can identify pairs of nodes on either side of the plane of symmetry—the main node and the secondary node. By default, the main node is the node that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the symmetry condition applies an equal displacement to the secondary node so that it remains symmetrical to the main node. Alternatively, if you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Planar symmetry** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces in which the planar symmetry will be enforced. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system, and select the axis of the coordinate system that represents the normal to the plane of symmetry.
7. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.
8. Enter the tolerance that will be used to determine symmetric points in the *X*-, *Y*-, and *Z*-axes.

The Optimization module uses the tolerance to identify the pairs of symmetric nodes across the symmetry plane.

9. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the faces are already symmetric about the plane. If the faces are not symmetric, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle**, and the faces are not symmetric, the Optimization module issues an error message and stops execution.
10. If desired, toggle on **Allow nonsymmetric mesh** (default) to use the planar symmetry geometric restriction for nonsymmetric meshes as well. Toggle off **Allow nonsymmetric mesh** to use the planar symmetry geometric restriction with symmetric meshes only.
11. Click **OK** to create the planar symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a rotational symmetry restriction

You can specify a rotational symmetry geometric restriction for a shape optimization. A rotational symmetry geometric restriction forces selected faces of the optimized model to be symmetric about a specified axis. You choose the axis of symmetry by specifying the starting and ending coordinates of a vector representing the axis. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh must be approximately symmetric around the axis of symmetry before the optimization starts so that the Optimization module can identify the group of nodes that sit on a selected face in a plane normal to the axis of symmetry. By default, the main node is the node in the group that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the symmetry condition applies an equal displacement to the rest of the nodes in the group (the secondary nodes) so that they remain symmetrical about the axis. If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes. The optimization determines how much the main node is displaced, and all other nodes are moved the same amount so that they remain symmetric about the selected axis.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Rotational symmetry (Shape)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces in which the rotational symmetry will be enforced. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Enter the coordinates of the starting point and the ending point of a vector representing the axis of symmetry.
7. Toggle on **Create a repeating pattern**, and enter the angle over which the pattern created by the optimization will be repeated. The value must be between 0° and 360°. A value of 0° implies the

secondary nodes are symmetric about the axis of symmetry, but the optimization does not create a repeating pattern.

8. If desired, toggle on **Allow nonsymmetric mesh** (default) to use the rotational symmetry geometric restriction for nonsymmetric meshes as well. Toggle off **Allow nonsymmetric mesh** to use the rotational symmetry geometric restriction with symmetric meshes only. When you allow nonsymmetric meshes, you can pick the **Start point** of the repeating pattern.
9. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.

   Alternatively, you can select **Region** and select a vertex that will be used to represent the main node.
10. Enter the tolerance that will be used to determine symmetric points in the *X*-, *Y*-, and *Z*-axes.

    The Optimization module uses the tolerance to identify the nodes on the surface that lie on a plane normal to the axis of symmetry and are equidistant from the axis of symmetry.
11. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the faces are already symmetric about the axis. If the faces are not symmetric, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the faces are not symmetric, the Optimization module issues an error message and stops execution.
12. Click **OK** to create the rotational symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a point symmetry restriction

You can specify a point symmetry geometric restriction for a shape optimization. A rotational symmetry geometric restriction forces selected faces of the optimized model to be symmetric about a specified point of symmetry. You specify the point of symmetry by selecting a coordinate system (the point of symmetry is assumed to be the origin of the coordinate system). You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh must be approximately symmetric about the point before the optimization starts so that the Optimization module can identify pairs of nodes on either side of the point of symmetry—the main node and the secondary node. By default, the main node is the node that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the symmetry condition applies an equal displacement to the secondary node so that it remains symmetrical to the main node. Alternatively, if you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Point symmetry (Shape)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces in which the point symmetry will be enforced. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system. The point of symmetry is assumed to be the origin of the selected coordinate system.
7. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.
8. Enter the tolerance in the *X*-, *Y*-, and *Z*-planes that will be used to determine symmetric points .

The Optimization module uses the tolerance to identify the nodes that are symmetric to the point of symmetry.

9. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the faces are already symmetric about the point. If the faces are not symmetric, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the faces are not symmetric, the Optimization module issues an error message and stops execution.
10. Click **OK** to create the point symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a stamp control restriction

You can specify a stamp control geometric restriction for a shape optimization. A stamp control geometric restriction results in an optimized model that can be manufactured by a tool and die stamping operation along a specified direction. You choose the direction by specifying the starting and ending coordinates of a vector representing the direction. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh should define a stampable model before the optimization starts; otherwise, the mesh may become distorted when the optimization creates a stampable model in the first iteration. The main node can lie anywhere in the region that you select to be governed by the stamp restriction. By default, the main node is the node in the region that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the stamp condition applies an equal displacement to the rest of the nodes in the region (the secondary nodes) so that the model remains stampable. If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes. The optimization determines how much the main node is displaced, and all other nodes are moved the same amount so that the model remains stampable.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Stamp control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the stamp control will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Enter the coordinates of the starting point and the ending point of a vector representing the direction along which the stamping tool moves.
7. Enter the draw angle, which represents the angle of the tool that is creating the stamped model. The value must be between 0° and 45°.

8. Enter a positive value for the amount of undercut that can be tolerated in the stamping region.
9. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.

   Alternatively, you can select **Region** and select a vertex that will be used to represent the main node.
10. Enter the tolerance in the *X*-, *Y*-, and *Z*-axes.

    The Optimization module uses the tolerance to identify the nodes on the surface that lie on a plane normal to the axis of symmetry and are equidistant from the axis of symmetry.
11. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the model is stampable. If the model is not stampable, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the model is not stampable, the Optimization module issues an error message and stops execution.
12. Click **OK** to create the stamp control geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a turn control restriction

You can specify a turn control geometric restriction for a shape optimization. A turn control geometric restriction results in an optimized model that can be manufactured by a tool on a lathe cutting into the model along a specified direction. You choose the direction by specifying the starting and ending coordinates of a vector representing the direction. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh should define a turnable model before the optimization starts; otherwise, the mesh may become distorted when the optimization creates a turnable model in the first iteration. The main node can lie anywhere in the region that you select to be governed by the turn restriction. By default, the main node is the node in the region that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the stamp condition applies an equal displacement to the rest of the nodes in the region (the secondary nodes) so that the model remains turnable. If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes. The optimization determines how much the main node is displaced, and all other nodes are moved the same amount so that the model remains turnable.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Turn control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the turn control will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Enter the coordinates of the starting point and the ending point of a vector representing the direction along which the cutting tool moves.

7. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.
8. Enter the tolerance in the *X*-, *Y*-, and *Z*-axes.

   The Optimization module uses the tolerance to identify the nodes on the surface that lie on a plane normal to the axis of symmetry and are equidistant from the axis of symmetry.
9. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the model is turnable. If the model is not turnable, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the model is not turnable, the Optimization module issues an error message and stops execution.
10. Click **OK** to create the turn control geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a demold control restriction

You can specify a demold geometric restriction for a shape optimization. A demold geometric restriction forces the optimized model to satisfy specified manufacturing requirements; for example, it can prevent undercuts and hollow regions in a part that must be extracted from a mold. You choose the demolding direction by specifying the starting and ending coordinates of a vector representing the axis. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh should define a demoldable model before the optimization starts; otherwise, the mesh may become distorted when the optimization creates a demoldable model in the first iteration. The main node can lie anywhere in the region that you select to be governed by the demold control restriction. By default, the main node is the node in the region that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the stamp condition applies an equal displacement to the rest of the nodes in the region (the secondary nodes) so that the model remains demoldable. If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes. The optimization determines how much the main node is displaced, and all other nodes are moved the same amount so that the model remains demoldable.

1. From the main menu bar, select **Geometric Restriction->Create**

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Demold control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the demold control will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. By default, the region in which the Optimization module checks for collisions is the same as the region in which it enforces the demold restriction. If desired, select from the top of the **Edit Geometric Restriction** dialog box, and select the collision check region.

Faces inside the collision check region cannot be penetrated by faces outside the region. If a node attempts to penetrate an element in the collision check region during the shape optimization, the Optimization module scales back the displacement of the node. The collision check region must include the faces to which the demold control will be applied.

7. Enter the coordinates of the starting point and the ending point of a vector representing the direction along which a mold is withdrawn from the demold region.
8. Enter the draw angle, which represents the angle of a mold that is being withdrawn from the demold region. The value must be between 0° and 45°.
9. Enter a positive value for the amount of undercut that can be tolerated in the demold control region.
10. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.
11. Enter the tolerance in the *X*-, *Y*-, and *Z*-axes.

    The Optimization module uses the tolerance to identify the nodes on the surface that lie on a plane normal to the axis of symmetry and are equidistant from the axis of symmetry.
12. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes a mold could be withdrawn from the demold region. If the mold could not be withdrawn, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle**, and the mold could not be withdrawn, the Optimization module issues an error message and stops execution.
13. Click **OK** to create the demold geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a drill control restriction

You can specify a drill control geometric restriction for a shape optimization. A drill control geometric restriction results in an optimized model that can be manufactured by a tool drilling into the model along a specified direction. The hole created by the tool is symmetric about the axis of the tool. In addition, the tool can be withdrawn from the hole. You choose the axis of the tool (and the drilling direction) by specifying the starting and ending coordinates of a vector representing the axis. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

The mesh should define a drillable model before the optimization starts; otherwise, the mesh may become distorted when the optimization creates a drillable model in the first iteration. The main node can lie anywhere in the region that you select to be governed by the drill control restriction. By default, the main node is the node in the region that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the stamp condition applies an equal displacement to the rest of the nodes in the region (the secondary nodes) so that the model remains drillable. If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes. The optimization determines how much the main node is displaced, and all other nodes are moved the same amount so that the model remains drillable.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Stamp control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the drill control will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Enter the coordinates of the starting point and the ending point of a vector representing the direction along which the drilling tool moves.

7. Enter the draw angle, which represents the angle of the tool that is drilling the hole. The value must be between 0° and 45°.
8. Enter a positive value for the amount of undercut that can be tolerated in the drill control region.
9. Select the method that the optimization will use to determine the main point. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.

   Alternatively, you can select **Region** and select a vertex that will be used to represent the main node.
10. Enter the tolerance in the *X*-, *Y*-, and *Z*-axes.

    The Optimization module uses the tolerance to identify the nodes on the surface that lie on a plane normal to the axis of symmetry and are equidistant from the axis of symmetry.
11. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the model is drillable. If the model is not drillable, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the model is not drillable, the Optimization module issues an error message and stops execution.
12. Click **OK** to create the drill control geometric restriction and to exit the editor.

Creating a penetration check restriction

A penetration check geometric restriction in a shape optimization results in an optimized model with faces that cannot penetrate selected regions.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Penetration check (Shape)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the penetration check will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the regions that cannot be penetrated by faces of the optimized model.
7. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the model is not already penetrating the selected regions. If the model is penetrating the selected region, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the model is penetrating the selected region, the Optimization module issues an error message and stops execution.
8. Click **OK** to create the penetration check geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a fixed region restriction

You can specify a fixed region restriction for a shape optimization. A fixed region is restrained in selected degrees of freedom (**1-**, **2-**, or **3-direction**. The degrees of freedom are defined relative to a selected coordinate system.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Fixed region** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the fixation will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system that defines the degrees of freedom. You can select the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).
7. Toggle on the degrees of freedom that you want to restrain.
8. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the region is already restrained in the selected degrees of freedom. If the region has displacement in the selected degrees of freedom, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the region has displacement in the selected degrees of freedom, the Optimization module issues an error message and stops execution.
9. Click **OK** to create the fixed region geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a growth restriction

You can specify a growth restriction for a shape optimization. A growth restriction limits how much a face can grow (surface nodes are moved out) or shrink (surface nodes are moving in) relative to the initial design. For example, if you are optimizing a model that will be cast, you can use a growth restriction to control the maximum and minimum wall thickness in a region.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Growth** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the growth restrictions will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Toggle on **Maximum in shrink direction**, and enter a positive value specifying the maximum inward displacement of a surface node.
7. Toggle on **Maximum in growth direction**, and enter a positive value specifying the maximum outward displacement of a surface node.
8. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the growth of the face is already limited. If the growth of the face exceeds the specified value, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the growth of the face exceeds the specified value, the Optimization module issues an error message and stops execution.
9. Click **OK** to create the growth geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a design direction restriction

You can specify a design direction restriction for a shape optimization. You can use a design direction restriction to keep selected nodes of your model on a planar or circular surface during the optimization. The optimization displaces the main node, and the design direction restriction applies an equal displacement (either magnitude or direction or both) to the rest of the nodes in the region (the client nodes). In addition, you can specify the axes of a coordinate system (rectangular, cylindrical, or spherical) that are applied.

The mesh should define nodes that can be moved along the design direction before the optimization starts; otherwise, the mesh may become distorted when the optimization moves nodes in the first iteration. The main node can lie anywhere in the region that you select to be governed by the design direction restriction. By default, the main node is the node in the region that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). If you are trying to optimize surfaces that are in contact, you can force the Optimization module to select the main node as the node to which the optimization is applying the least growth or the most amount of shrinkage. Alternatively, you can select a single point that will be used as the main node by all other nodes.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Design Direction** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the design direction restriction will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Choose whether a client node should follow the main node by moving in the same direction or with the same magnitude or both.
7. If you chose to move a client node in both the same direction and with the same magnitude as the main node, toggle on the axes along which the movement will be applied.

8. Select the method that the optimization will use to determine the main node. In most cases you should select **Determine from most growth and least shrinkage**. You should select **Determine from least growth and most shrinkage** only if you are trying to optimize faces that are involved in contact.

   Alternatively, you can select **Region** and select a vertex that will be used to represent the main node.

9. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the nodes can be moved along the specified design direction. If the nodes cannot be moved along the design direction, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the nodes cannot be moved along the design direction, the Optimization module issues an error message and stops execution.
10. Click **OK** to create the design direction geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a slide region control restriction

You can specify a slide region control, or contact, geometric restriction for a shape optimization. A slide region control geometric restriction results in an optimized model with a face that contacts a specified surface and follows the contours of the surface. You can select the specified surface from the model. Alternatively, the specified surface can be a surface of revolution generated by a rotating a selected face about a selected axis. You choose the axis of rotation by specifying the starting and ending coordinates of a vector representing the axis. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

If you select the face from the model, the definition of the surface slide control restriction is complete. If you select a face that will be used to form a surface of revolution, the Optimization module selects the main node from the nodes on the face. By default, the main node is the node on the face that the optimization moves out the most (the most growth) or moves in the least (the least shrinkage). The optimization displaces the main node, and the surface slide restriction forces an equal displacement to the rest of the nodes on the face (the secondary nodes) so that the contact conditions are satisfied.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Slide region control** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the surface slide control will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the approach that you will use to define the contact surface.
7. If you selected the **Free-form** approach, select the contact surface.
8. If you selected the **Conserve a turnable surface** approach, do the following:

   a. Select the **Master Node Criteria**.

   b. Select the starting point and the ending point of a vector representing the axis of rotation.

c. Select the coordinate system that defines the vector. You can select the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

d. Enter the tolerance in the *X*-, *Y*-, and *Z*-axes.

The Optimization module uses the tolerance to identify the nodes on the contact surface.

9. If desired, toggle on **Ignore in first design cycle** (default). When the optimization starts, it assumes the surfaces are in contact. If the surfaces are not in contact, the Optimization module issues a warning and tries to continue. If you toggle off **Ignore in first design cycle** and the surfaces are not in contact, the Optimization module issues an error message and stops execution.

10. Click **OK** to create the slide region control geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a topology or sizing optimization*
- *Creating a geometric restriction in a shape optimization*

Creating a geometric restriction in a bead optimization

This section describes how to create a geometric restriction in a bead optimization.

In this section:

- *Creating a filter restriction*
- *Creating a fixed region restriction*
- *Creating a growth restriction*
- *Creating a penetration check restriction*
- *Creating a planar symmetry restriction*
- *Creating a point symmetry restriction*
- *Creating a rotational symmetry restriction*

Creating a filter restriction

You can specify a filter restriction for a bead optimization.

A filter restriction creates a more consistent workflow between bead and other optimization types. In addition, it usually leads to better optimization convergence behavior.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Filter (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the fixation will be applied.

   For more information, see *Using the face curvature method to select multiple faces*.

   To select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. **Optional:** By default, the region in which the Optimization module checks for the filter restriction is the same as the region in which it enforces the filter restriction. If desired, select from the top of the **Edit Geometric Restriction** dialog box, and select the filter region check region.
7. Enter the **filter radius** in terms of value and whether absolute or relative units are used.

   If this value is omitted, double the average edge length of the model is used.

8. **Optional:** Select **Relative radius** to input the relative radius value.

   By default, this check box is off.

9. Click **OK** to create the filter geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating a fixed region restriction

You can specify a fixed region restriction for a bead optimization. A fixed region is restrained in selected degrees of freedom (**1-**, **2-**, or **3-direction**. The degrees of freedom are defined relative to a selected coordinate system.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Fixed Region (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the fixation will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system that defines the degrees of freedom. You can select the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).
7. Toggle on the degrees of freedom that you want to restrain.
8. Click **OK** to create the fixed region geometric restriction and to exit the editor.

Additional information

• *Creating a geometric restriction in a bead optimization*

Creating a growth restriction

You can specify a growth restriction for a bead optimization. A growth restriction limits how much a face can grow (shell nodes at the location of the beads are moving out) relative to the initial design during the creation of the bead.

For a general bead optimization, you must limit the bead height by creating a growth geometric restriction that limits the displacement in the growth direction. The growth geometric restriction is required because it is the only restriction on the displacement of the nodes.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Growth (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the growth restrictions will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Toggle on **Maximum in shrink direction**, and enter a positive value specifying the maximum inward displacement of a node. You can specify the maximum inward displacement of a node only for a general bead optimization.
7. Toggle on **Maximum in growth direction**, and enter a positive value specifying the maximum outward displacement of a node relative to the initial location.
8. Click **OK** to create the growth geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating a penetration check restriction

A penetration check geometric restriction in a bead optimization results in an optimized model with faces that cannot penetrate selected regions.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Penetration check (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the faces to which the penetration check will be applied. For more information, see *Using the face curvature method to select multiple faces*.

   If you would rather select from a list of existing face sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting faces, click **Done** in the prompt area.

   The **Edit Geometric Restriction** dialog box appears.

6. Select the regions that cannot be penetrated by faces of the optimized model.
7. Click **OK** to create the penetration check geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating a planar symmetry restriction

You can specify a planar symmetry geometric restriction for a condition-based bead optimization. A planar symmetry geometric restriction forces the optimized model to be symmetric about a specified plane. You specify the plane of symmetry by selecting the axis of a coordinate system that is the normal to the plane of symmetry. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information). You can choose to remove frozen regions from the symmetry restriction.

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Planar symmetry (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the planar symmetry will be enforced or click **Done** to apply the planar symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system, and select the axis of the coordinate system that represents the normal to the plane of symmetry.
7. Click **OK** to create the planar symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating a point symmetry restriction

You can specify a point symmetry geometric restriction for a condition-based bead optimization. A point symmetry geometric restriction forces the optimized model to be symmetric about a specified point. You specify the point of symmetry by selecting a coordinate system (the point of symmetry is assumed to be the origin of the coordinate system). You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Point symmetry (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the point symmetry will be enforced or click **Done** to apply the point symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system. The point of symmetry is assumed to be the origin of the selected coordinate system.
7. Click **OK** to create the point symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating a rotational symmetry restriction

You can specify a rotational symmetry geometric restriction for a condition-based bead optimization. A rotational symmetry geometric restriction forces the optimized model to be symmetric about a specified axis of a coordinate system. You can use the global coordinate system, or you can create a datum coordinate system (see *Methods for creating a datum coordinate system*, for more information).

1. From the main menu bar, select **Geometric Restriction->Create**.

   The **Create Geometric Restriction** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Geometric Restriction Manager**. (You can display the **Geometric Restriction Manager** by selecting **Geometric Restriction->Manager** from the main menu bar.)
   - Click the tool in the Optimization module toolbox.

2. From the **Create Geometric Restriction** dialog box that appears, enter the name of the geometric restriction.
3. Select **Rotational symmetry (Bead)** from the list of geometric restrictions, and click **Continue**.
4. From the viewport, select the region in which the rotational symmetry will be enforced or click **Done** to apply the rotational symmetry restriction to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

5. When you have finished selecting the geometric restriction region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Geometric Restriction** dialog box appears.

6. Select the coordinate system, and select the axis of the coordinate system that represents the axis of symmetry.
7. Enter the **Repeating segment size**, which is the angle (in degrees) specifying the size of the repeating segment. The value must be greater than or equal to 2°.
8. Click **OK** to create the rotational symmetry geometric restriction and to exit the editor.

Additional information

- *Creating a geometric restriction in a bead optimization*

Creating local stop conditions

A local stop condition is an optional setting that indicates to the Optimization module when your optimization has converged on a solution. For example, you can specify that an optimization is complete when the change in an optimization function drops below a specified value between iterations. You can choose the variable that will be compared, such as displacement or equivalent stress. You can choose the comparison operator—greater than, equal, to, less than, etc. The change in optimization function can be calculated from the previous iteration (default) or from the first iteration.

In addition, when you create an optimization process in the Job module, you can enter the global stop condition—the maximum number of optimization cycles that should be completed before the optimization process ends. For more information, see *Creating and editing optimization processes*. In most optimization problems you will use only the global stop condition. Local stop conditions are supported by only shape optimization and are rarely required.

1. From the main menu bar, select **Stop Condition->Create**.

   The **Create Stop Condition** dialog box appears.

   Tip: You can initiate the **Create** procedure in two other ways:

   - Click **Create** in the **Stop Condition Manager**. (You can display the **Stop Condition Manager** by selecting **Stop Condition->Manager** from the main menu bar.)
   - Click the  tool in the Optimization module toolbox.

2. From the **Create Stop Condition** dialog box that appears, enter the name of the stop condition and click **Continue**.
3. From the viewport, select the region to which the stop condition will be applied or click **Done** to apply the stop condition to the entire model.

   By default, Abaqus/CAE allows you to select all of the model. To select faces or cells, use the **Selection** toolbar to change the type of object that you can select to **Face** or **Cells**. For more information, see *Filtering your selection based on the type of object*.

   If you would rather select from a list of existing sets, do the following:

   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.

   b. Select the set of interest, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click the button—**Select in Viewport** or **Sets**—on the right side of the prompt area.

4. When you have finished selecting the stop condition region, click **Done** in the prompt area. For more information on selecting objects, see *Selecting objects within the viewport*."

   The **Edit Stop Condition** dialog box appears.

5. Choose one of the following operators for calculating the objective function:

   - **Maximum value**
   - **Minimum value**
   - **Sum of values**

- **Number of values**

6. Select the variable that will be used by the stop condition. You can choose either a displacement or an equivalent stress for a shape optimization.
   - The displacement of each node from the original shape. You can select the overall displacement, only the displacement due to the addition of material, or only the displacement due to the reduction of material. In addition, you can select the relative value of the displacement or the absolute value.
   - The equivalent stress in a shape optimization. You can select the equivalent stress of the whole model or the equivalent stress calculated from the following:
     - Only the nodes in the region selected for the task.
     - Only restricted nodes in the region selected for the task. Restricted nodes are restrained by a constraint or an Abaqus boundary condition.
     - Only free or unrestricted nodes in the region selected for the task. Free or unrestricted nodes are unconstrained.
     - Only surface nodes.
7. Select the comparison operator.
8. Select the iteration that will be used to define the reference value. The Optimization module compares the current value of the optimization function with the reference value.
9. If desired, enter a value and the operation that will be used to modify the reference value. In effect, you are normalizing the value of the variable that is defining the stop condition. For a discussion of the reference value, see *Objectives and Constraints*.
10. Click **OK** to save your data and to exit the editor.

The Job module

You can use the Job module to create and manage analysis jobs and to view a basic plot of the analysis results. You can also use the Job module to create and manage adaptivity analyses and co-executions.

See the tutorial in *Creating and Analyzing a Simple Model in Abaqus/CAE* for examples of how to submit and monitor a job.

In this section:

- *Understanding the role of the Job module*
- *Understanding analysis jobs*
- *Understanding adaptivity processes*
- *Understanding co-executions*
- *Understanding optimization processes*
- *Restarting an analysis*
- *Creating, editing, and manipulating jobs*
- *Using the job editor*
- *Creating, editing, and manipulating adaptivity processes*
- *Using the adaptivity process editor*
- *Creating, editing, and manipulating co-executions*
- *Creating, editing, and manipulating optimization processes*

Understanding the role of the Job module

Once you have finished all of the tasks involved in defining a model (such as defining the geometry of the model, assigning section properties, and defining contact), you can use the Job module to analyze your model. The Job module allows you to create a job, to submit it for analysis, and to monitor its progress. If desired, you can create multiple models and jobs and run and monitor the jobs simultaneously.

In addition, you have the option of creating only the analysis input file for your model. This option allows you to view and edit the input file before submitting it for analysis. For an Abaqus/Standard or Abaqus/Explicit analysis, you can also view and edit the analysis keywords for a model by selecting **Model->Edit Keywords->*model name*** from the main menu bar.

You can create and submit a job based on an existing input file instead of an Abaqus/CAE model; for example, if you have created an input file outside Abaqus/CAE but would like to run the analysis, monitor its progress, and view the results in Abaqus/CAE. Jobs can be created only for input files that do not contain references to other results files; for example, you cannot create a job based on an input file that contains a restart analysis.

If you have defined adaptive remeshing rules in the Mesh module, you can submit a mesh adaptivity process. Abaqus/CAE submits a succession of jobs, each of which attempts to improve solution accuracy and reduce error indicators over the previous job.

For an Abaqus co-simulation, you can create a co-execution to execute two analysis jobs in synchronization with one another.

Understanding analysis jobs

This section provides an overview of the Job module.

In this section:

- *Basic steps for analyzing a model*
- *Entering and exiting the Job module*
- *The Job Manager*
- *The job editor*
- *Selecting a job type*
- *Monitoring the progress of an analysis job*
- *Submitting a job remotely*

Basic steps for analyzing a model

After you have defined your model, you are ready to analyze it.
Analyzing a model involves the following steps, each of which can be performed using either the **Job** menu on the main menu bar or the **Job Manager**:

Create and configure an analysis job

You create an analysis job by selecting **Job->Create** from the main menu bar. Abaqus/CAE asks you to name the new job and to associate it with a model selected from the model database or with an existing input file. You can select any model that exists in the database; you are not limited to the current model. The job editor allows you to configure the job settings.

Write the input file

When you submit a job associated with a model for analysis, Abaqus/CAE first generates an input file representing your model and then Abaqus/Standard or Abaqus/Explicit performs the analysis using the contents of this file. Alternatively, you can ask Abaqus/CAE to generate only the input file; Abaqus/CAE writes the input file in ASCII format, and you can view and edit it in your working directory.



Warning:

If you edit the input file for a model using a text editor outside Abaqus/CAE and then submit the job for that model in the Job module, your changes to the input file will be lost. Instead, you must submit the modified input file directly for analysis by creating a new job and selecting **Input file** as the job **Source**. However, if you use the **Keywords Editor** to modify the generated keywords for a model, those modifications are retained in the model and apply to any jobs associated with that model.

Submit the job for analysis

You submit the job for analysis by selecting **Job->Submit** from the main menu bar. As the analysis progresses, Abaqus/CAE displays information from the status, data, log, and message files in the job monitor dialog box. After your job is completed, you can display results from the output database in the Visualization module by selecting **Job->Results** from the main menu bar.

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*
- *Using the job editor*

Entering and exiting the Job module

You can enter the Job module at any time during a session by clicking **Job** in the **Module** list located in the context bar. The **Job** menu appears in the main menu bar.

To exit the Job module, select any other module from the main menu. If your job completed successfully, you can also exit the Job module by selecting **Job**->**Results** from the main menu bar; you will enter the Visualization module, and the output database for your analysis job will be opened automatically.

You need not save your job before exiting the module; it will be saved automatically when you save the entire model by selecting **File**->**Save** or **File**->**Save As** from the main menu bar.

The Job Manager

The **Job Manager**, which is similar to other managers in Abaqus/CAE, allows you to do the following:

- Create an analysis job and associate the new job with a selected model or input file.
- Edit the selected analysis job.
- Copy, rename, or delete the selected analysis job.



Note:

A job associated with a model can be copied only to a job associated with a model. A job associated with an input file can be copied only to a job associated with an input file.

In addition, the **Job Manager** allows you to do the following:

- *Write* an input file for a model-based job without submitting it for analysis.
- Perform a *data check* on a model.
- *Submit* a job for analysis.
- *Continue* an analysis to completion after performing a data check.
- *Monitor* the analysis as it progresses.
- View the *results* from a job.
- *Terminate* a job that is currently running.

You can display the **Job Manager** by selecting **Job->Manager** from the main menu bar. *Figure 1* shows the layout of the **Job Manager**.

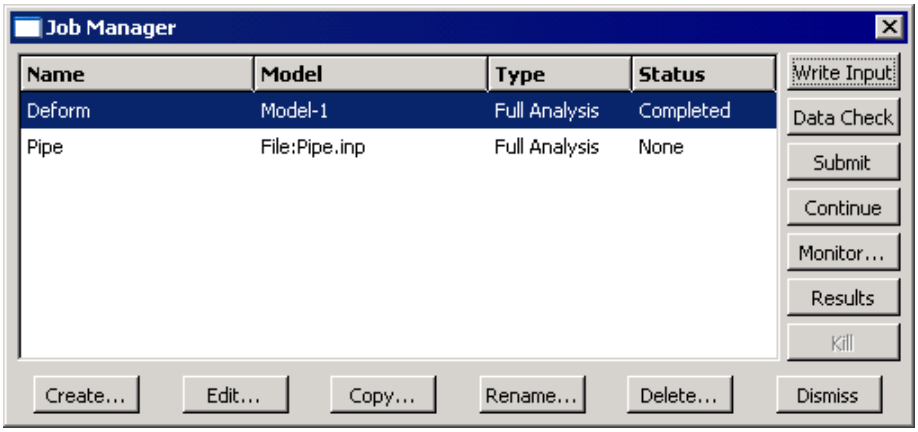


Figure 1: The Job Manager.

The four columns of the **Job Manager** display the following:

Name

The **Name** column displays the name of the job. Click **Rename** to rename the selected job.

Model

The **Model** column displays the name of the model or input file associated with the job.

Type

The **Type** column displays the job type that you selected when you configured the job using the job editor. The job type can be one of the following:

- **Full Analysis**
- **Recover**
- **Restart**

(See *Selecting a job type* for more information.) You can use the job editor to change the job type as long as the job is not running.

Status

The **Status** column displays the current status of the analysis job and is updated continually while your job is running. The status can be one of the following:

None

The job has not been submitted for analysis.

Check Submitted

The input file has been written, and the model is being submitted for a data check.

Check Running

The data check of the model is running.

Check Completed

The data check of the model has completed successfully; you can now continue with the full analysis.

Submitted

The input file has been written, and the job is being submitted for a full analysis.

Running

The job has been submitted for a full analysis and is running.

Completed

The analysis is complete. You can click **Results** to view the contents of the output database and graphically verify your results.

Aborted

The job has been aborted due to problems such as fatal errors in the input file or lack of disk space.

Terminated

The job has been killed by the user.

For detailed instructions on using the **Job Manager** to create, edit, and manipulate jobs, see the following sections:

- *Creating a new analysis job*
- *Writing the input file only*
- *Performing a data check on a model*
- *Submitting an analysis job*
- *Continuing an analysis job after a data check*
- *Terminating an analysis job*
- *Viewing the results of your job*

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*
- *Using the job editor*

The job editor

You use the job editor to customize the settings for a new job or to edit the settings for an existing job. You can display the job editor by selecting **Job->Create** or **Job->Edit->*job name*** from the main menu bar. (You can also click **Create** or **Edit** in the job manager.)

The job editor contains the following tabbed pages:

Submission

Use the **Submission** tabbed page to configure the submission attributes of your job, such as job type, license type for the SimUnit license model, run mode, and submit time. You can also use the submission tabbed page to specify that the job is to be submitted to remote queues that have been configured by your local Abaqus environment file or by your system administrator.

General

Use the **General** tabbed page to configure job settings such as the analysis input file processor printout, the name of the directory used for scratch files, and the output format of the results.

Preprocessor printout options are not available for a job associated with an input file; you must specify these options in the input file itself.

Memory

Use the **Memory** tabbed page to configure the amount of memory allocated to an Abaqus analysis.

Parallelization

Use the **Parallelization** tabbed page to configure the parallel execution of an Abaqus analysis job, such as the number of processors to use.

Precision

Use the **Precision** tabbed page to specify either single or double precision for Abaqus/Explicit analyses. You can also choose the precision of nodal output that is written to the output database during the analysis.

For detailed instructions on using the job editor to define jobs, see the following sections :

- *Navigating the job customization options*
- *Configuring job submission attributes*
- *Choosing the job type*
- *Choosing the job run mode*
- *Setting the job submit time*
- *Specifying general job settings*
- *Controlling job memory settings*
- *Controlling job parallel execution*

- *Controlling job precision*

Additional information

- *Understanding analysis jobs*
- *Using the job editor*

Selecting a job type

The **Submission** tabbed page in the job editor allows you to choose between the following job types:

Full analysis

Submit a job with this option selected to generate (or regenerate) the input file (if the job is associated with a model), perform a complete analysis of your model, and write the results to the output database. This option is the default.

Recover (Explicit)

This option is available only when you are running Abaqus/Explicit. Submit a job with this option selected to complete your analysis after Abaqus/Explicit stopped unexpectedly; for example, after filling a disk or after a network problem.

You cannot use this job type to recover an Abaqus/Standard job that terminated prematurely. Instead, you should use the Abaqus/Standard restart capabilities as described in *Recovering an Abaqus/Standard analysis*.

Restart

Submit a job with this option selected to start the analysis using data from a previous analysis of a specified model. You must use the **Edit Model Attributes** dialog box to specify the job to read data from and to specify the step from which to restart the analysis. When you create a job that refers to a model with restart data attributes, Abaqus/CAE by default selects the job type to be **Restart**.

Restart analyses are described in more detail in *Restarting an analysis*. You cannot create a restart job for a job that was associated with an input file.

The **Job Type** settings are analogous to parameters of the Abaqus execution procedure; for more information, see *Abaqus/Standard and Abaqus/Explicit Execution*. For detailed instructions on choosing a job type, see *Choosing the job type*.

Additional information

- *Restarting an analysis*
- *Understanding analysis jobs*
- *Using the job editor*

Monitoring the progress of an analysis job

The **Job Manager** and **Co-execution Manager** continually update the status of analysis jobs in the model database. In addition, Abaqus/CAE prints error messages from the analysis products to the message area and creates diagnostic files in your current working directory.

You can monitor information concerning a submitted job by selecting **Job->Monitor->*job of your choice*** from the main menu bar or by selecting the job of your choice and clicking **Monitor** in the **Job Manager**. The job monitor dialog box for that job appears, as shown in *Figure 1*. You can display as many job monitors as necessary to view information on multiple jobs.

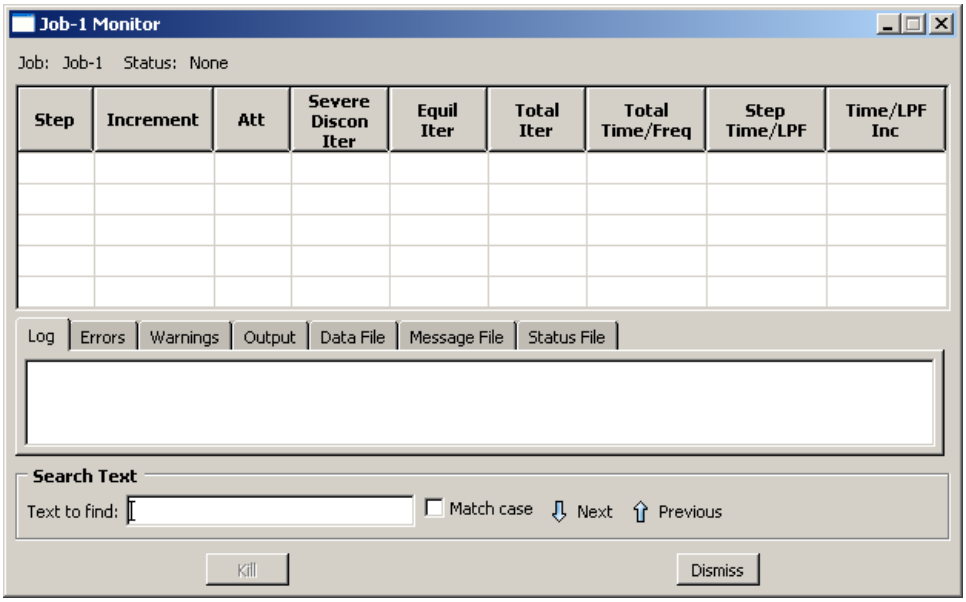


Figure 1: The job monitor.

The jobs submitted for a co-execution appear in the Model Tree in the **Jobs** container under the co-execution in the **Co-executions** container. You can monitor these jobs by clicking mouse button 3 on the job in the Model Tree and selecting **Monitor**.

The top half of the job monitor dialog box displays the information available in the status file that Abaqus creates for the analysis. The table headings are customized for each job based on the settings sent from the analysis. The bottom half of the dialog box displays the following information:

- Click the **Log** tab to display the start and end times that appear in the log file for the analysis.
- Click the **Errors** and **Warnings** tabs to display the errors or the warnings associated with the analysis. Abaqus/CAE indicates the presence of errors or warnings by prepending an exclamation point before the **Errors** and **Warnings** tab. If a particular region of the model is causing the error or warning, a node or element set will be created automatically that contains that region. The name of the node or element set appears with the error or warning message, and you can view the set using display groups in the Visualization module. (For more information on display groups, see *Using display groups to display subsets of your model*.)

  Abaqus/CAE may not perform all consistency checks when it creates an input file, which can result in warnings or error messages during an analysis. If your analysis generates warning or error messages, consider running a data check analysis (see *Performing a data check on a model*) to diagnose and fix possible problems in your model.

  The number of error and warning messages that appear in the job monitor is limited by the environment parameters **cae_error_limit** and **cae_warning_limit**, respectively (see *Job customization parameters*, for details). If the number of errors or warnings exceeds the job monitor limit, consult the data, message, or status file for a complete list of messages.

- Click the **Output** tab to display a record of each output data entry as it is written to the output database. In addition, if you requested that Abaqus monitor the values of a degree of freedom of a particular node to the message and status files, the **Output** tabbed page records each time this information is written and the value of the degree of freedom at that point of the analysis. (For more information, see *Understanding output requests*, and *Degree of freedom monitor requests*.)
- As the analysis proceeds, Abaqus creates the following files:
  - Abaqus/Standard: Data file, message file, and status file
  - Abaqus/Explicit: Data file, message file, and status file

  Abaqus/CAE activates the **Data File**, **Message File**, and **Status File** tabs accordingly. You can click any of these tabs to browse or search the corresponding file for additional error and warning messages.

  Note:

  Abaqus/CAE populates the **Data File**, **Message File**, and **Status File** tabbed pages only for locally submitted analyses; this information will not be displayed in the job monitor for remote jobs. In addition, although Abaqus/CAE updates the contents of these pages periodically as the analysis runs, the data might not always be synchronized with the latest data in the files.

For detailed information on the different output files that Abaqus creates during an analysis, see *About Output*.

You can search for specific error or warning messages within any of the files displayed in tabbed pages in the job monitor. Select the desired file tab, enter a search string in the **Text to find** field, and click **Next** or **Previous** to step through the file from one hit to the next. Toggle on **Match case** to perform a case-sensitive search.

The information presented in the job monitor dialog box is updated continually as the analysis progresses. If the job fails, the **Errors** tabbed page appears in front of the other tabbed pages automatically to help you determine the cause of the failure. In addition, an exclamation point appears on the tab if any error or warning messages are output.

If you start monitoring a job and then exit Abaqus/CAE, close the current model database, or open a new model database, Abaqus/CAE will stop updating the job monitor. The job will continue to run; however, the job monitor will not report the status of the job or update the increment information.

If you requested **DOF Monitor** output on a particular degree of freedom for a particular node, Abaqus/CAE provides another opportunity to monitor the job by plotting the values of the degree of freedom over time. The plot appears in a new viewport that is generated automatically when you submit the job. If the visible part of the canvas is already filled with one or more viewports, the new viewport may be placed on a part of the canvas that is not visible; in this case you should tile or cascade the viewports or enlarge the canvas to bring the viewport into view. (For information on requesting output for a particular degree of freedom for a particular node, see *Degree of freedom monitor requests*.)

If necessary, you can terminate the analysis job by clicking **Kill** at the bottom of the job monitor dialog box.

Additional information

- *Understanding analysis jobs*
- *Understanding co-executions*
- *Creating, editing, and manipulating jobs*
- *Creating, editing, and manipulating co-executions*

Submitting a job remotely

When you configure a job, you can request that Abaqus/CAE route the job to a specified queue on a remote Linux host computer. You can specify the remote queue by selecting an associated queue name in the **Submission** tabbed page of the job editor.



Note: The job editor displays default settings for memory usage, parallelization, and precision based on the environment files effective for the current session on the local machine. When you submit a job to a remote machine, Abaqus replaces the default settings with those from the environment files on the remote machine. Nondefault settings in the job editor are saved with the job and will be used regardless of where you run the analysis.

Each queue name that appears in the job editor refers to an entry in your Abaqus environment file in which you specify how you want the job to be run on the host computer. In other words, when you select a queue name in the job editor, you specify not only the desired queue on the host computer but also other options, such as the directory on the host computer in which you want to run the job and the files you want copied back to your local directory when the job is complete.

You can specify your preferences for running a job remotely by adding the following to your Abaqus environment file:

```
def onCaeStartup():
    import os

    def makeQueues(*args):
        session.Queue(name,
                      queueName,
                      hostName,
                      fileCopy,
                      directory,
                      driver,
                      remotePlatform,
                      filesToCopy,
                      deleteAfterCopy,
                      description)

    addImportCallback('job', makeQueues)
```

This entry is written using the Abaqus command language. The following list describes each argument in the entry above.

name

The queue name that you want to appear in the job editor.

queueName

The name of an existing queue on the host computer. (For information on creating queues on the host computer, refer to the *Abaqus Configuration Guide*.)

hostName

The name of the host computer. The default is an empty string.

fileCopy

When the analysis is complete, the value of this argument determines whether or not the analysis files will be copied back to the directory from which the job was submitted. The default value is `ON`.

directory

The name of the directory on the host computer where you want the job run. You must have write privileges to this directory. The default is an empty string.

driver

The name of the command on the host computer to execute Abaqus/Standard or Abaqus/Explicit. The default is `abaqus`.

remotePlatform

The platform on the remote machine. The default is `UNIX` (for Linux).

filesToCopy

The extensions of the analysis files that you want copied back to the local directory when the job is complete or `ALL`. By default, all of the files generated during job execution are copied.



Note:

The restart (`.res`) file, the Abaqus/Explicit state (`.abq`) file, and the packaging (`.pac`) file are platform-dependent; if your local platform and remote platform settings differ, you will not be able to use these files without some kind of translation. All of the other files can be copied across platforms without any difficulty.

deleteAfterCopy

The value of this argument determines whether remote files are to be deleted after they are copied to the local machine. The default value is `OFF`.

description

A short description of the queue.

The `name` and `queueName` arguments must be included in each queue definition. However, if you do not include any of the other arguments in a queue definition, default values will be supplied automatically. An example queue definition is shown below:

```
def onCaeStartup():
    import os

    def makeQueues(*args):
        session.Queue(name='long',
                      queueName='aba_long',
                      hostName='jobserver',
                      directory='/scratch/' + os.environ['USER'])
    addImportCallback('job', makeQueues)
```

The commands in the example above configure the following:

name

The queue name displayed in the job editor is `long`.

queueName

The queue name on the host computer is `aba_long`.

hostName

The name of the host computer is `jobserver`.

directory

The directory on the host computer where Abaqus will store the input file and all other files associated with the job is `/scratch/`*your user name*.

Since the *fileCopy*, *driver*, *remotePlatform*, *filesToCopy*, and *deleteAfterCopy* arguments have been left out of the entry above, the default options for these parameters are assigned to this queue automatically.

If you want to create two or more queues, you can repeat the line containing the `session.Queue` command as many times as necessary. For example, the following Abaqus environment file entry specifies two queues, one named `long` and the other named `job`:

```
def onCaeStartup():
    import os

    def makeQueues(*args):
        session.Queue(name='long',
                      queueName='aba_long',
                      hostName='jobserver',
                      directory='/scratch/' + os.environ['USER'])
        session.Queue(name='job',
                      queueName='aba_job',
                      hostName='jobserver',
                      fileCopy=OFF)

    addImportCallback('job', makeQueues)
```

The monitoring functions described in *Monitoring the progress of an analysis job*, are available for jobs run remotely just as they are for jobs run locally. However, the output database for the job, like any other analysis files that you may have requested, is not copied to your local directory until after the job is complete. As a result, you must create and start a network output database connector if you want to use the Visualization module to view the results being generated by an analysis running on a remote system. For more information, see *Accessing an output database on a remote computer*.

Additional information

- *Environment File Settings*
- *Abaqus Scripting User's Guide*

Understanding adaptivity processes

This section provides an overview of adaptivity processes.
You use an adaptivity process to control a succession of analysis jobs that are adaptively remeshed by Abaqus/CAE based on the contents of your remeshing rules. For more information about adaptive remeshing and the other adaptivity techniques that are available in Abaqus, see *About Adaptivity Techniques*.

In this section:

- *What is an adaptivity process?*
- *When will my mesh adaptivity stop iterating?*
- *Manual mesh adaptivity*
- *Using a combination of automatic and manual mesh adaptivity*
- *The Adaptivity Process Manager*

What is an adaptivity process?

An adaptivity process is a succession of analysis jobs where Abaqus/CAE remeshes selected regions of the model between each job. Abaqus/CAE modifies the mesh to be used in each analysis job in response to error estimates that were computed during the previous analysis and written to the output database. For more information, see *About Adaptive Remeshing*.

You have considerable flexibility in how you execute this succession of jobs. Adaptivity processes are stored in the model database and are maintained between sessions.

When will my mesh adaptivity stop iterating?

When you create an adaptivity process, you define a sequence of jobs to be run successively. You can specify the number of jobs in the sequence by specifying the maximum number of analysis iterations allowed, $iter_{max}$. The maximum number of remeshing operations will be $iter_{max}-1$, because no remeshing will be performed after the final analysis job is run. When you submit the remesh process, Abaqus/CAE runs each job in the sequence until one of the following conditions is met:

- All active remeshing rules are satisfied. For more information, see *What are remeshing rules?*.
- All jobs are completed, indicating that Abaqus/CAE has reached the maximum number of analysis iterations that you specified.
- A job fails to complete. This failure can be due to convergence difficulties or to machine resource problems.

Manual mesh adaptivity

As an alternative to automatic mesh adaptivity, you can submit a job using the Job manager. When you define a remeshing rule in the Mesh module, you request error indicator output variables that are written to the output database when the job executes. After the job has completed, you can return to the Mesh module and ask Abaqus/CAE to create a new mesh using a combination of your remesh rules and the error indicators in the output database. You can perform this process indefinitely to adapt your mesh successively until the goals of the remesh rules are met. For more information, see *What is the difference between automatic adaptive remeshing and manual adaptive remeshing?*.

Using a combination of automatic and manual mesh adaptivity

In many cases you will want to transition between automatic and manual use of adaptive remeshing. Some examples include:

- You must keep the current Abaqus/CAE session active for automatic remeshing to occur between analysis jobs. If you exit Abaqus/CAE and start a new session, you must use manual remeshing to remesh the model.
- You start an automatic sequence of jobs, but a machine problem (such as a full disk) interrupts the analysis prematurely. You can restart the analysis manually and then continue it in a fully automatic manner. Variations of this case include an analysis ending because it failed to converge or an analysis ending because you terminated the job during a particular iteration.
- You can experiment with sizing function parameters and remesh your model manually until you find a set of remeshing rules that creates a mesh that appears to be responding actively to your variables of interest. When you are comfortable with your remeshing rules, you can switch to automatic mesh adaptivity and complete a specified number of analysis iterations.
- The mesh generated by the automatic mesh adaptivity procedure converges to an acceptable configuration. You can now modify the loads applied to the model and perform an additional manual iteration that makes a minor meshing adjustment to compensate for the new load.

You can transition directly between automatic and manual mesh adaptivity. To transition from automatic to manual mesh adaptivity, you enter the Mesh module and select **Adaptivity->Manual Adaptive Remesh** from the main menu bar and remesh the model. You then return to the Job module and submit the job for analysis. Conversely, to transition from manual to automatic mesh adaptivity, you create an adaptivity process in the Job module and submit it for analysis.

The Adaptivity Process Manager

The **Adaptivity Process Manager**, which is similar to other managers in Abaqus/CAE, allows you to do the following:

- Create an adaptivity process and associate the process with a selected model.
- Edit the selected adaptivity process.
- Copy, rename, or delete the selected adaptivity process.

In addition, the **Adaptivity Process Manager** allows you to do the following:

- Perform a *data check* on a model before submitting an adaptivity process.
- *Submit* an adaptivity process.
- *Continue* an adaptivity process after performing a data check.

You can display the **Adaptivity Process Manager** by selecting **Adaptivity->Manager** from the main menu bar. *Figure 1* shows the layout of the **Adaptivity Process Manager**.

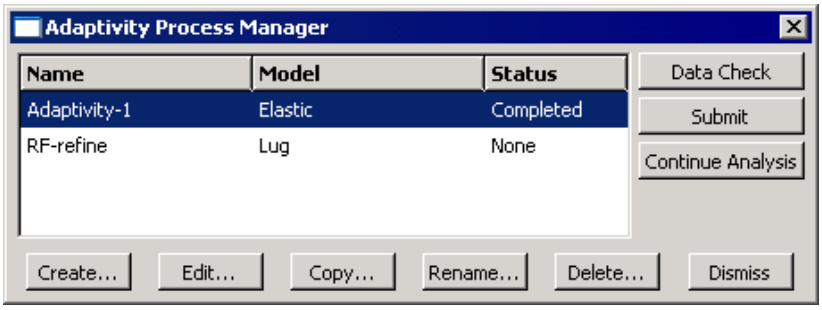


Figure 1: The Adaptivity Process Manager.

The three columns of the **Adaptivity Process Manager** display the following:

Name

The **Name** column displays the name of the adaptivity process. Click **Rename** to rename the selected process.

Model

The **Model** column displays the name of the model associated with the adaptivity process.

Status

The **Status** column displays the current status of the adaptivity process and is updated continually while the process is running. The status can be one of the following:

None

The process has not been submitted for analysis.

Check Submitted

The model is being submitted for a data check.

Check Running

The data check of the model is running.

Check Completed

The data check of the model completed successfully; you can now continue the adaptivity process.

Submitted

The process is being submitted for execution.

Running

The process has been submitted for analysis and is running.

Completed

The process has run to completion.

Aborted

The process has been aborted due to problems such as fatal errors in the model or lack of disk space.

Terminated

The process has been terminated by the user.

For detailed instructions on using the **Adaptivity Process Manager** to create, edit, and manipulate adaptivity processes, see the following sections :

- *Creating a new adaptivity process*
- *Performing a data check on an adaptivity process*
- *Submitting an adaptivity process*
- *Continuing an adaptivity process after a data check*
- *Terminating an adaptivity process*

Additional information

- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*
- *Using the adaptivity process editor*

Understanding co-executions

This section provides an overview of co-executions.

In this section:

- *What is a co-execution?*
- *The Co-execution Manager*
- *The co-execution editor*

What is a co-execution?

A co-execution is the co-simulation execution of two analysis jobs that are executed in Abaqus/CAE in synchronization with one another using the same functionality as described in *Understanding analysis jobs*. For more information on this analysis technique, see *Structural-to-Structural Co-Simulation*.

The Co-execution Manager

The **Co-execution Manager**, which is similar to other managers in Abaqus/CAE, allows you to do the following:

- Create a co-execution and associate it with selected analyses.
- Edit the selected co-execution.
- Copy, rename, or delete the selected co-execution.

In addition, the **Co-execution Manager** allows you to do the following:

- Perform a *data check* on the models before submitting a co-execution.
- *Submit* a co-execution.
- *View* the results of a co-execution.
- *Terminate* a co-execution that is currently running.

You can display the **Co-execution Manager** by selecting **Co-execution->Manager** from the main menu bar. *Figure 1* shows the layout of the **Co-execution Manager**.

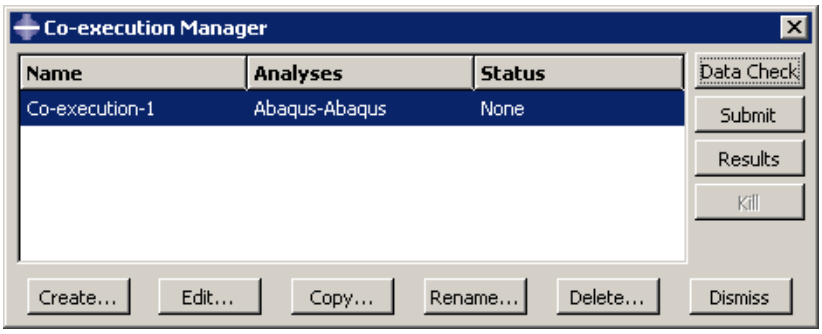


Figure 1: The Co-execution Manager.

The three columns of the **Co-execution Manager** display the following:

Name

The **Name** column displays the name of the co-execution. Click **Rename** to rename the selected co-execution.

Analyses

The **Analyses** column displays the type of analyses associated with the co-execution; for example, **Abaqus-Abaqus**.

Status

The **Status** column displays the status of the jobs in the co-execution and is updated continually while the co-execution is running. The status can be one of the following:

None

The co-execution has not been submitted for analysis.

Check Submitted

The analyses are being submitted for a data check.

Check Running

The data check of the analyses are running.

Check Completed

The data check of the analyses completed successfully; you can now continue the co-execution.

Submitted

The co-execution is being submitted.

Running

The co-execution has been submitted and is running.

Completed

The co-execution has run to completion.

Aborted

The co-execution has been aborted due to problems such as fatal errors in one of the models or lack of disk space.

Terminated

The co-execution has been terminated by the user.

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

The co-execution editor

You use the co-execution editor to select the models, define the initial job parameter settings in the individual job editors, and change the co-execution timeout value. After the co-execution is created, the job parameters in the co-execution editor are unavailable for editing; you must edit the individual jobs.

You can display the co-execution editor by selecting **Co-execution->Create** or **Co-execution->Edit->*co-execution name*** from the main menu bar. (You can also click **Create** or **Edit** in the co-execution manager.)

The co-execution editor contains the following tabbed pages for specifying the job parameters:

Submission

Use the **Submission** tabbed page to configure the submission attributes of your job, such as job type, run mode, and submit time. You can also use the submission tabbed page to specify that the job is to be submitted to remote queues that have been configured by your local Abaqus environment file or by your system administrator.

General

Use the **General** tabbed page to configure job settings such as the analysis input file processor printout and the name of the directory used for scratch files.

Memory

Use the **Memory** tabbed page to configure the amount of memory allocated to an Abaqus analysis.

Parallelization

Use the **Parallelization** tabbed page to configure the parallel execution of an Abaqus analysis job, such as the number of processors to use.

Precision

Use the **Precision** tabbed page to specify either single or double precision for Abaqus/Explicit analyses. You can also choose the precision of nodal output that is written to the output database during the analysis.

Additional information

- *Understanding co-executions*

Understanding optimization processes

This section provides an overview of optimization processes.

In this section:

- *What is an optimization process?*
- *Understanding the files generated by an optimization process*
- *Postprocessing an optimization*
- *The Optimization Process Manager*
- *The optimization process editor*

What is an optimization process?

Figure 1 illustrates how an optimization process iteratively updates the design variables, modifies the finite element model, and runs Abaqus analyses while it searches for an optimized solution.

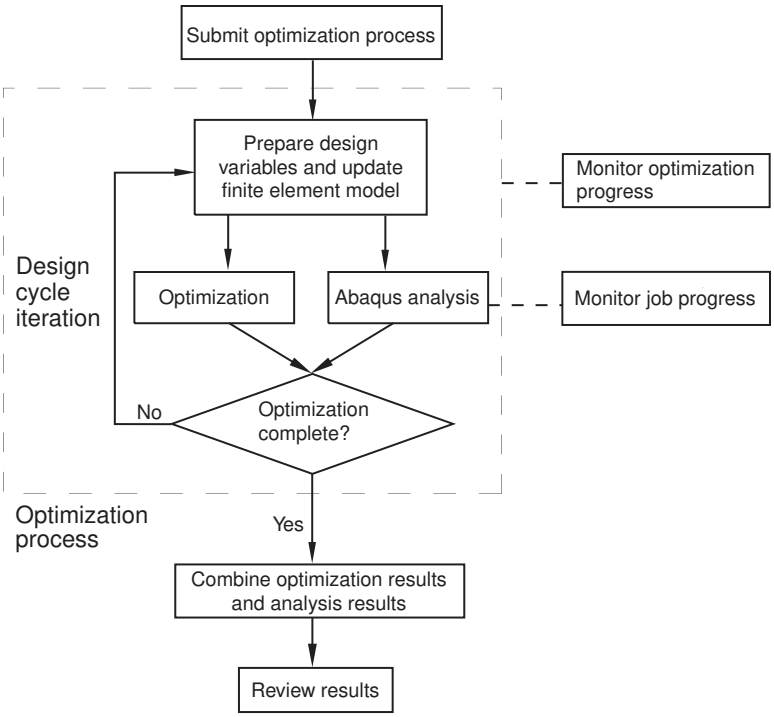


Figure 1: An optimization process iteratively searches for an optimized solution.

An optimization process reads an optimization task that you defined in the Optimization module and iteratively searches for an optimized solution based on the objective functions and constraints that you defined in the optimization task. Each iteration is called a design cycle. During each design cycle the optimization modifies the finite element model and an Abaqus analysis is performed on the modified model. The optimization process reads the results of the analysis and decides whether to end the optimization because either the solution is optimal or a specified stop condition has been reached or whether to continue the optimization and iterate on another design cycle.

An optimization process generates optimization results and analysis results. You must combine the optimization results and analysis results into a single output database file to view the results of the optimization in the Visualization module, as described in *Postprocessing an optimization*. The Optimization module does not support the use of parts and assemblies in the Abaqus input file. When you run an optimization task, the Optimization module generates a flattened input file that does not use parts and assemblies, regardless of your Abaqus model attributes.

An optimization process appears in the **Analysis** section of the Model Tree and contains the Abaqus jobs that were run during the optimization, as shown in *Figure 2*.

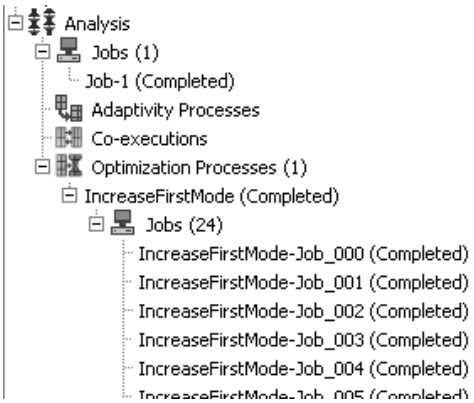


Figure 2: An optimization process in the Model Tree.

You can check the validity of your optimization process; however, the validation does not check your Abaqus model. You should run a complete analysis of your model and make sure that it runs to completion before you attempt to run an optimization process.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Understanding the files generated by an optimization process

When an optimization process is executing, it creates two types of data that are saved in separate files—optimization results and analysis results.

Optimization results

The optimization results consist of the optimization variables and the optimization values. The optimization variables depend on the type of optimization you are performing.

Topology optimization

The optimization variable is the normalized material distribution variable (MAT_PROP_NORMALIZED).

Shape optimization

The optimization variables are the displacement variables:

- DISP_OPT: a vector indicating the direction in which the nodes were moved during the shape optimization. Because of mesh smoothing and filtering, the vector may not be coincident with the node normal vector.
- DISP_OPT_VAL: the magnitude of the DISP_OPT vector with a sign indicating the direction of the displacement—positive for growth and negative for shrinkage.
- CTRL_INPUT: the value of the objective function at each node (stress, for example).

Sizing optimization

The optimization variables are the shell thickness and the change in shell thickness (THICKNESS and DELTA_THICKNESS).

Bead optimization

The optimization variables are the displacement variables:

- DISP_NORMAL_VAL: the magnitude of a vector indicating the direction in which the nodes were moved during the bead optimization along the node normal vector.
- DISP_OPT: a vector indicating the direction in which the nodes were moved during the bead optimization. Because of mesh smoothing and filtering, the vector may not be coincident with the node normal vector.
- DISP_OPT_VAL: the magnitude of the DISP_OPT vector with a sign indicating the direction of the displacement—positive for growth and negative for shrinkage.

The optimization variables are saved in separate optimization (`.onf`) files after every design cycle. In addition, the optimization values, such as the value of the objective and the constraints, are written to a comma-separated text file (`.csv`) after each design cycle. (You are viewing the optimization values when you monitor the progress of an optimization process, as described in *Monitoring your optimization process*.)

Analysis results

The analysis results are the field and history data generated by Abaqus during the analysis. During each design cycle a new output database file is created. In the initial design cycle, Abaqus writes both field and history data to the new output database file; however, during subsequent design cycles, Abaqus writes only field data. Abaqus

also generates data (`.dat`), message (`.msg`), and status (`.sta`) files during each design cycle. To conserve disk space and to speed up postprocessing, the output database file and the data, message, and status files are saved only at specified intervals; by default, after the initial, first, and final design cycles. (You specify the rate at which the files are saved when you create the optimization process, as described in *Creating and editing optimization processes*.) In addition, Abaqus saves the new input file that is generated during each design cycle.

Postprocessing an optimization

To view the results of an optimization process in the Visualization module, you must combine the optimization results and the analysis results into a single results output database file, as described in *Combining optimization results*.

Creating the base results output database file

The output database file that you choose to be the base results becomes the starting point for the combined results output database file.

Choosing between the initial and the final design cycle

You can specify that the base results be taken from the output database file generated by the initial design cycle or from the output database file generated by the final design cycle. In most cases you will select the initial design cycle and view the progression of the optimization from the initial design cycle to the final design cycle; for example, to follow the change in stiffness. Select the final design cycle if you performed a frequency optimization and want to view the frequency and the mode shape of the first few modes. If you performed a shape optimization, you should take the base results from the output database file generated by the first design cycle.

Choosing the original model

The Optimization module modifies the material definitions and the section assignments before running the initial design cycle of the optimization process. The original model is the model that existed before any modifications were made by the Optimization module.

You can specify that the base results be taken from the output database file generated by the original model. However, the optimization process does not run an Abaqus analysis of the original model. Therefore, you must run the analysis manually before you can choose the original model as the base results. (The Abaqus input file for the original model is stored, along with all the input files generated by the optimization process, in the *optimization process name*`\SAVE.inp` directory. The name of the input file for the original model is *optimization process name*`_org.inp`.)

The output database file that you use for the base results must be generated without parts and assemblies, as described in *Writing input files without parts and assemblies*. The input files generated by the Optimization module do not include parts and assemblies, regardless of any user settings in Abaqus/CAE. However, if you create the base results from an input file that was not generated by the Optimization module or by executing a job from Abaqus/CAE, you must ensure that the resulting output database file does not contain parts and assemblies.

Appending to the base results

After you have specified the output database file that will be used as the source of the base results, Abaqus/CAE appends the optimization results and the Abaqus analysis results to the combined output database file.

Appending the optimization results

Abaqus/CAE appends the optimization variables as field data to the combined output database file after every design cycle, and each design cycle appears as a frame in the combined output database file. Similarly, Abaqus/CAE appends the optimization values as history data to the combined output database file.

Appending the analysis results

You can do the following to specify which analysis results data are written to the combined results output database file:

- Specify from which design cycles the analysis results should be written.
- Specify from which models the analysis results should be written. Abaqus/CAE creates a combined results output database file for each model in your optimization process.
- Specify for a selected model from which steps within the model the analysis results should be written.
- Specify which analysis field variables should be written.

History data are not written to the combined database file during the combine process. The combined output database file contains only the history data from the base results output database file.

It is recommended that you save the analysis results after the initial and final design cycle when you create the optimization process. After the optimization is complete, you can select the output database file generated during the initial design cycle as the base results. You can then combine the base results output database file with the optimization results from every design cycle along with the analysis results from the final design cycle, as illustrated in *Table 1*.

Table 1: Saving optimization and analysis results in the combined output database file.

	Design Cycle				
	Initial	**First**	**Second**	**Third**	**Fourth (Final)**
Data saved	Analysis results (field and history data)	Optimization results	Optimization results	Optimization results	Optimization results and analysis results (field data only)
Action	Create base results	Append to base results	Append to base results	Append to base results	Append to base results

For example, if you executed a topology optimization, you can then do the following:

- View the initial state of your model; for example, the initial geometry and the loads and boundary conditions.
- View the change in the optimization variables—normalized material distribution variable (MAT_PROP_NORMALIZED)—during each design cycle to observe the progression of the optimization.
- View the final state of your model; for example, the optimized geometry and the displacements and the stresses and strains.
- Create a history plot that tracks the change in the objective and constraints.

You can apply similar conditions to view the beginning and ending state and the progression of the shape and sizing optimizations.

Combining optimization results is supported only for optimizations that you configured with the following analysis cases:

- Simple analysis (a single model, a single step, and a single load case)
- Frequency or modal analysis
- Linear perturbation analysis with multiple load cases

- Optimization with multiple models

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

The Optimization Process Manager

The **Optimization Process Manager** allows you to create and configure an optimization process and to edit, copy, rename or delete the selected optimization process.

In addition, the **Optimization Process Manager** allows you to do the following:

- Write copies of the TOSCA parameter (`.par`) file and the Abaqus input (`.inp`) file to your working directory.
- Validate the optimization before submitting an optimization process to ensure that the optimization task has been configured correctly and the Abaqus model exists.
- Submit an optimization process.
- Restart an optimization process that has failed because of a problem external to the optimization or the Abaqus analysis, such as a failure to obtain an Abaqus license.
- Monitor the progress of an optimization process.
- Extract a smooth isosurface meshed representation of the optimized model surface in the form of a file that can be transferred to a CAD system or back into Abaqus/CAE.
- Combine the optimization results and the analysis results created by an optimization process into a single results output database file that can be displayed by the Visualization module.
- View the results of an optimization process.

You can display the **Optimization Process Manager** by selecting **Optimization Process Manager->Manager** from the main menu bar. *Figure 1* shows the layout of the **Optimization Process Manager**.

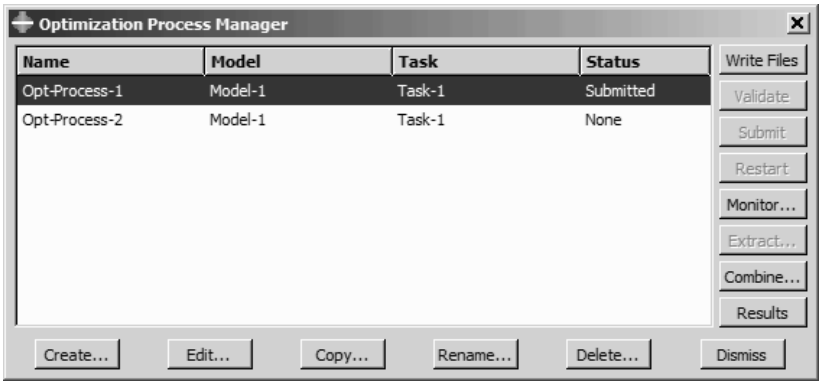


Figure 1: The Optimization Process Manager.

The four columns of the **Optimization Process Manager** display the following:

Name

The **Name** column displays the name of the optimization process. Click **Rename** to rename the selected optimization process.

Model

The **Model** column displays the Abaqus/CAE model associated with the optimization process.

Task

The **Task** column displays the optimization task associated with the optimization process.

Status

The **Status** column displays the status of the jobs in the optimization process and is updated continually while the optimization process is running. The status can be one of the following:

None

The optimization process has not been submitted for analysis.

Submitted

The optimization process is being submitted.

Running

The optimization process has been submitted and is running.

Completed

The optimization process has run to completion.

Aborted

The optimization process has been aborted due to problems such as fatal errors in one of the models or lack of disk space.

Terminated

The optimization process has been terminated by the user.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

The optimization process editor

You use the optimization process editor to select the Abaqus/CAE model and the optimization task that will be included in the optimization process. You can also configure the optimization settings, such as the maximum number of optimization iterations.

You can display the optimization process editor by selecting **Optimization Process->Create** or **Optimization Process->Edit->*optimization process name*** from the main menu bar. (You can also click **Create** or **Edit** in the **Optimization Process Manager**.)

The optimization process editor allows you to configure the following:

Optimization

Use the **Optimization** tabbed page to specify the maximum number of optimization iterations that should be run before the process terminates and the frequency at which data are saved. For more information, see *Creating and editing optimization processes*.

Submission

Use the **Submission** tabbed page to configure the submission attributes of your optimization process, such as the submit time and whether the process is to be submitted to remote queues that have been configured by your local Abaqus environment file or by your system administrator. For more information, see *Configuring job submission attributes*.

Memory

Use the **Memory** tabbed page to configure the amount of memory allocated to the Abaqus jobs during the optimization process. For more information, see *Controlling job memory settings*.

Parallelization

Use the **Parallelization** tabbed page to configure the parallel execution of the Abaqus analysis jobs during the optimization process, such as the number of processors to use. For more information, see *Controlling job parallel execution*.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Restarting an analysis

This section describes the restart capability in Abaqus/CAE.
If your model contains multiple steps, you do not have to analyze all of the steps in a single analysis job. Indeed, it is often desirable to run a complex analysis in stages. This allows you to examine the results and confirm that the analysis is performing as expected before continuing with the next stage. The restart files generated by an Abaqus analysis allow you to continue the analysis from a specified step. For more information, see *Restarting an Analysis*.

In this section:

- *Controlling a restart analysis*
- *Files required to restart an analysis*
- *Rules governing a restart analysis*
- *The relationship between the model and the restart analysis*
- *Restarting after adding more analysis steps to the model*
- *Restarting after modifying existing analysis steps*
- *Restarting from the middle of a step*
- *Visualizing results from restart analyses*
- *Recovering an Abaqus/Standard analysis*
- *Remote submission of restart jobs*

Controlling a restart analysis

By default, no restart information is written for an Abaqus/Standard analysis and restart information is written only at the beginning and end of each step for an Abaqus/Explicit analysis.
You can use the Step module to change the frequency at which restart information is written. For more information, see *Restart output requests*.

After an analysis has generated restart information, you can control the following aspects of the subsequent restart analysis:

Model attributes

To configure a restart analysis, you must specify whether the model should reuse data from a previous analysis of the same model. You can specify the step and the increment or time interval of the previous analysis from which the new analysis should start. You can also choose one of the following:

- Allow the selected step to continue to completion.
- Terminate the selected step at the specified increment and begin a new step.

For more information, see *Specifying model attributes*.

Job type

You use the job editor to specify the job type. If you edit the model attributes to specify that the model should reuse data from a previous analysis and you create a job that refers to this model, Abaqus/CAE sets the job type to **Restart**. For more information, see *Selecting a job type*. You cannot create a restart job that refers to a job that was associated with an input file.

Files required to restart an analysis

To restart an analysis, various files that were created by the previous analysis must be available in the directory from which you started the Abaqus/CAE session.

Abaqus/Standard

- Output database (`.odb`)
- Restart file (`.res`)
- Model file (`.mdl`)
- Part file (`.prt`)
- State file (`.stt`)

Abaqus/Explicit

- Output database (`.odb`)
- Restart file (`.res`)
- Model file (`.mdl`)
- Package file (`.pac`)
- Part file (`.prt`)
- State files (`.abq` and `.stt`)
- Selected results file (`.sel`)

The restart analysis generates an error if any of these files are not available in the current directory.

Rules governing a restart analysis

Defining the restart information for models and jobs is straightforward using Abaqus/CAE; however, you should understand the following before you use the analysis restart capability:

- The model used in the restart analysis must be the same as the model used in the original analysis up to the restart location. Specifically,
  - For the new model do not modify or add any geometry, mesh, materials, sections, beam section profiles, material orientations, beam section orientations, interaction properties, or constraints.
  - Similarly, do not modify any steps or prescribed conditions (loads, boundary conditions, fields, interactions, or output requests) at or before the restart location.
- Keyword editing is not supported for restart analyses. If you used the keyword editor to edit the original model, Abaqus/CAE ignores those changes when you restart the analysis.



Warning:

Abaqus/CAE does not perform any checks to ensure that the restart data stored for the original job are consistent with the model used in the restart analysis. In some cases the analysis will terminate with error messages. In other cases the analysis will run, but the results may not correspond with your intent. A specific example of a case that will result in an error message is one where you define a coupling constraint and its accompanying reference node in the original model and then suppress the constraint prior to the original analysis submission. In this example Abaqus/CAE maintains the reference node definition but Abaqus/Standard or Abaqus/Explicit may remove it from the model definition if it is not attached to other elements or constraints. Your subsequent Abaqus/CAE restart analysis in this example will be seen as inconsistent with the original analysis. In many cases like this example you can manually modify the restart analysis input file to correct the discrepancy.

The relationship between the model and the restart analysis

When you first submit a job based on a model for analysis, Abaqus/CAE generates an input file based on the definition of your model; and that input file, in turn, is submitted for analysis. The input file contains the element and node definitions generated by the Mesh module along with the materials, steps, output requests, loads, interactions, etc. that you specified using Abaqus/CAE.

To run a restart analysis of a job associated with a model, it is recommended that you copy the model to a new model. The new model contains all the element and node definitions from the original model along with all the materials, steps, output requests, loads, interactions, etc. To request a restart analysis, you edit the new model's attributes and specify that the analysis continue from a specified step of the original analysis. When you create a new restart job that refers to the new model, Abaqus/CAE sets the job type to **Restart**.

When you submit the new job for analysis, Abaqus/CAE generates an input file based on the restart information. No part, assembly, or property data are written to the restart input file. Only the following data are written:

- Steps that appear after the analysis is restarted.
- Data from step-dependent objects that are associated with steps that appear after the restart step; for example, loads, boundary conditions, fields, and output requests.
- New regions and amplitudes that are used by step-dependent objects that are associated with steps that appear after the restart step.

Abaqus reads the part, assembly, and property data, along with data from steps that appear before the restart step from the restart files that were generated by the original analysis. As a result, although the model contains all this information, the information is not written to the input file that is submitted to Abaqus/Standard or Abaqus/Explicit. This is an important consideration. Some changes that you make to the model in a restart analysis, say a material property, will not appear in the input file that is analyzed.

Restarting after adding more analysis steps to the model

The most common use of the restart capability is to analyze your model and then add one or more steps to the model and continue the analysis. This section describes an example of this usage.

Assume that you have done the following:

- Created a model called `Model-A` that has two analysis steps, `Step-1` and `Step-2`.
- Used the **Edit Restart Requests** dialog box in the Step module to output the restart information at the end of each step.
- Created a job called `Job-A` that uses `Model-A`.
- Analyzed the model.

After studying the results of the analysis, you decide to add another step, `Step-3`, to the model. The restart capability allows you to compute the results for `Step-3` without having to repeat the computations for `Step-1` and `Step-2`. The following steps describe a recommended procedure to follow:

1. Copy `Model-A` to a new model, say `Model-A-restart`, and make `Model-A-restart` the current model.
2. Add the new `Step-3` to `Model-A-restart`.
3. Add new prescribed conditions (loads, boundary conditions, interactions, fields, or output requests) in `Step-3`, or modify the prescribed conditions propagated from `Step-2`.
4. From the main menu bar, select **Model->Edit Attributes->Model-A-restart**. From the **Edit Model Attributes** dialog box that appears, do the following:
   - Enter `Job-A` as the job from which the restart data will be read.
   - Set the restart location:
     - Enter `Step-2` to indicate the step from which the restart data will be read.
     - Choose **Restart from the end of the step**. `Step-3` will continue the analysis after the end of `Step-2`.
5. Copy `Job-A` to a new job, say `Job-A-restart`, that uses `Model-A-restart`.
6. From the main menu bar, select **Job->Edit->Job-A-restart**. From the **Edit Job** dialog box that appears, choose a job type of **Restart**.
7. Submit `Job-A-restart` for analysis.

Restarting after modifying existing analysis steps

You can use the restart capability to analyze your model and then modify an existing step before continuing the analysis. This section describes an example of this usage.

Assume that you have done the following:

- Created a model called `Model-A` that has two analysis steps, `Step-1` and `Step-2`.
- Used the **Edit Restart Requests** dialog box in the Step module to output the restart information at the end of each step.
- Created a job called `Job-A` that uses `Model-A`.
- Analyzed the model.

After examining the results of `Job-A`, you realize that you need to make changes to `Step-2`, to an output request in `Step-2`, or to a prescribed condition in `Step-2`, such as a load or boundary condition. You may also want to add some new steps with additional output requests and loads and with changes to the boundary conditions. You know that you can still use the results from `Step-1`; however, you realize that the results from `Step-2` are no longer valid. The restart capability allows you to recompute the results for `Step-2` without having to repeat the computations for `Step-1`. The following steps describe a recommended procedure to follow:

1. Copy `Model-A` to a new model, say `Model-A-restart`, and make `Model-A-restart` the current model.
2. Do the following:
   - Make the desired changes to `Step-2`.
   - Add new steps after `Step-2`.
   - Create new prescribed conditions in `Step-2` and subsequent steps.
3. From the main menu bar, select **Model->Edit Attributes->Model-A-restart**. From the **Edit Model Attributes** dialog box that appears, do the following:
   - Enter `Job-A` as the job from which the restart data will be read.
   - Set the restart location:
     - Enter `Step-1` to indicate the step from which the restart data will be read.
     - Choose **Restart from the end of the step**. `Step-2` will continue the analysis after the end of `Step-1`.
4. Copy `Job-A` to a new job, say `Job-A-restart`, that uses `Model-A-restart`. Abaqus/CAE sets the job type to `Restart`.
5. Submit `Job-A-restart` for analysis.

Restarting from the middle of a step

You can use the restart capability to continue the analysis from the middle of a completed step or from the middle of a partially completed step. The restarted analysis uses a new step that continues the analysis from a specified increment of the previous step. This section describes an example of this usage.

Assume that you have done the following:

- Created a model called `Model-A` that has two Abaqus/Standard analysis steps, `Step-1` and `Step-2`.
- Used the **Edit Restart Requests** dialog box in the Step module to output the restart information every 10 increments.
- Created a job called `Job-A` that uses `Model-A`.
- Analyzed the model.

You suspected that the loading is too severe for the structure and might lead to instabilities or collapse of the structure and, hence, to numerical convergence difficulties. As a result, you used the Step module to save the restart information every 10 increments of each step. When you ran the analysis, you found that the analysis terminated before the end of `Step-2`, at increment `25`. You used the Visualization module to look at the results of the analysis and realized that the negative eigenvalue messages confirmed your suspicion that the structure might have become unstable. Now that you have some idea of the collapse load level, you want to restart the analysis at increment `20` of `Step-2` with lower levels of loading and more frequent output of results data. The following steps describe a recommended procedure to follow:

1. Copy `Model-A` to a new model, say `Model-A-restart`, and make `Model-A-restart` the current model.
2. Add the new `Step-3` to `Model-A-restart`.
3. Modify the load levels and output requests in `Step-3`. For example, if the load at `Step-2`, increment `25` was `260` and the load at `Step-2`, increment `20` was `250`, you might change the load level for the end of `Step-3` to `262`. In addition, you can increase the frequency at which data are written to the output database so that you can follow the progression of the collapse of the structure. You can also reduce the size of the maximum increment to control the resolution of the data that are generated by the analysis. For more information, see *Restarting an Analysis*.
4. From the main menu bar, select **Model->Edit Attributes->Model-A-restart**. From the **Edit Model Attributes** dialog box that appears, do the following:
   - Toggle on **Read data from job** and enter `Job-A` to indicate the job from which the restart data will be read.
   - Set the restart location:
     - Enter `Step-2` to indicate the step from which the restart data will be read.
     - Choose **Restart from increment, interval, iteration, or cycle**, and enter `20` to indicate the increment from which the restart data will be read.
     - Choose **and terminate the step at this point** to indicate that `Step-2` should terminate at increment `20`. `Step-3` will continue the analysis from this location.
5. Copy `Job-A` to a new job, say `Job-A-restart`, that uses `Model-A-restart`. Abaqus/CAE sets the job type to `Restart`.
6. Submit `Job-A-restart` for analysis.

Visualizing results from restart analyses

Each restart analysis creates a new output database. Consider the example described in *Restarting after adding more analysis steps to the model*, where you did the following:

- Ran an analysis of `Job-A` that referred to `Model-A` that contains `Step-1` and `Step-2`. As a result, the output database generated by the analysis, `Job-A.odb`, contains results from `Step-1` and `Step-2`.
- Ran a restart analysis of `Job-A-restart` that referred to `Model-A-restart`. Although `Model-A-restart` contains `Step-1`, `Step-2`, and `Step-3`, the output database generated by the restart analysis, `Job-A-restart.odb`, contains results from only `Step-3`.

You can use the Visualization module to create deformed, contour, and symbol plots of field data; however, you can plot the results from only one output database at any time. You must change the output database if you are looking at the results from `Step-1` or `Step-2` and want to look at the results from `Step-3`.

If you want to create an animation of results from all three steps, you must combine the two output databases. Abaqus provides an execution procedure for this purpose. For more information, see *Joining Output Database (.odb) Files from Restarted Analyses*.

Alternatively, you can create a history plot of a variable over all three steps using the following technique:

- Create one *X–Y* data object for the variable from the original output database—`Job-A.odb`.
- Create a second *X–Y* data object from the output database generated by the restart analysis—`Job-A-restart.odb`.
- Create an *X–Y* plot of the two *X–Y* data objects. Alternatively, you can create a new *X–Y* data object using the **Operate on XY data** option, append the first two *X–Y* data objects, and create an *X–Y* plot of the result.

For more information, see *X–Y plotting*.

You should take care that your *X–Y* data objects are not referring to the same step. For example, consider the restart analysis described in *Restarting after modifying existing analysis steps*, where you modified an existing step and added a new step before continuing the analysis. The first output database contains data from `Step-1` and `Step-2`. The second output database contains data from a modified `Step-2` and the new `Step-3`. The first *X–Y* data object that you create should use only the data from `Step-1` and should exclude the data from the original `Step-2`. The second *X–Y* data object should use the data from the modified `Step-2` and the new `Step-3`.

Recovering an Abaqus/Standard analysis

Abaqus/Explicit has a recovery mechanism for analysis jobs that terminate prematurely; for example, because of disk space or power failures. For more information, see *Selecting a job type*. However, unlike Abaqus/Explicit, Abaqus/Standard does not have a recovery mechanism. The following example describes how you can use the restart capabilities of Abaqus/Standard to continue an analysis that terminated prematurely. Assume the following:

- You analyzed `Model-A` containing `Step-1`, `Step-2`, and `Step-3`.
- Because of a power outage, the analysis ended prematurely at `Increment 17` of `Step-2`.
- You requested that restart information be saved every 10 increments. As a result, the last restart information was saved at `Increment 10` of `Step-2`.

The following procedure describes how you can recover the analysis using the last restart information that was saved:

1. Copy `Model-A` to a new model, say `Model-A-recover`, and make `Model-A-recover` the current model.
2. From the main menu bar, select **Model->Edit Attributes->Model-A-recover**. From the **Edit Model Attributes** dialog box that appears, do the following:
   - Toggle on **Read data from job**, and enter `Job-A` to indicate the job from which the restart data will be read.
   - Set the restart location:
     - Enter `Step-2` to indicate the step from which the restart data will be read.
     - Choose **Restart from increment, interval, iteration, or cycle**, and enter `10` to indicate the increment from which the restart data will be read.
     - Choose **and complete the step**. `Step-2` will continue the analysis after increment `10` and run to completion.
3. Copy `Job-A` to a new job, say `Job-A-recover`, that uses `Model-A-recover`. Abaqus/CAE sets the job type to `Restart`.
4. Submit `Job-A-recover` for analysis.

Remote submission of restart jobs

If you submit a job to a remote machine for analysis, by default Abaqus copies the files required for a restart analysis back to your local directory when the job is complete. For more information, see *Submitting a job remotely*. If the remote job terminates prematurely, you may have to copy the files back to your local directory manually.

Most of the files required for a restart analysis are platform-dependent binary files. As a result, you will not be able to continue a restart analysis on a Windows machine after starting the analysis on a Linux machine, or vice versa. The part (`.prt`) file contains ASCII text and can be copied across platforms.

The lack of portability across platforms with different binary formats does not preclude you from restarting an analysis on a machine that is different from the machine on which the original analysis was run. However, the two machines must be binary compatible, and it is recommended that the original analysis and the restart analysis both be executed on the same platform. Abaqus does not test cross-platform compatibility of restart files.

Creating, editing, and manipulating jobs

This section describes how you create, edit, and manipulate jobs using either the main menu or the **Job Manager**.

In this section:

- *Creating a new analysis job*
- *Writing the input file only*
- *Performing a data check on a model*
- *Submitting an analysis job*
- *Continuing an analysis job after a data check*
- *Terminating an analysis job*
- *Viewing the results of your job*

Creating a new analysis job

To create a new analysis job, select **Job->Create** from the main menu bar. Analysis jobs are stored in the model database and are maintained between sessions.

1. From the main menu bar, select **Job->Create**.

   The **Create Job** dialog box appears.

   Tip: You can also create a new analysis job by clicking the icon in the module toolbox or by clicking **Create** in the **Job Manager**.

2. Type the name of the new job in the **Name** text field. The name that you specify must adhere to the file name rules of your operating system.
3. Click the arrow to the right of the **Source** field to choose the source for the job.
   - Select **Model** to create a job based on a model created in Abaqus/CAE.

     The **Model** list displays all the models defined in the model database. From this list, select the model to associate with the new job.
   - Select **Input file** to create a job based on an input file that may or may not have been created in Abaqus/CAE.

     Click ; Abaqus/CAE lists all the files in the selected directory with the file extension `.inp`. Select the input file to associate with the new job, and click **OK**.

     Note:

     You cannot create a job based on an input file that contains references to other results files, such as a restart analysis, an import analysis, or a submodel analysis.

4. Click **Continue**.

   The job editor appears.
5. If desired, enter the job description. When you submit the job for analysis, Abaqus/CAE writes the job description immediately following the input file header. The job description is not written to the output database and is not retained upon import into Abaqus/CAE. For more information, see *Importing descriptions*.
6. In the editor, enter all data necessary to define the job and click **OK**. (For more information, see *Using the job editor*.)

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Writing the input file only

By default, when you submit a job associated with a model for analysis, Abaqus/CAE generates an input file representing your model and then Abaqus analyzes that input file. However, sometimes you may prefer to generate the input file and then view or edit it before performing the analysis.

To write the input file without immediately performing the analysis, select **Job->Write Input->*job of your choice*** from the main menu bar. The input file *job name* **.** `inp` is written to the directory from which you started Abaqus/CAE. You can also write an input file by selecting the job of your choice and then clicking **Write Input** in the **Job Manager**.

The input file is written in ASCII format and can be viewed and edited using a text editor. If you are familiar with Abaqus keywords, you can check the input file for errors and check that the keywords, parameters, and data were generated as expected. Keep in mind that Abaqus/CAE processes your input to the graphical user interface, performing additional calculations when needed, to determine the keywords that accurately reflect your modeling intent. You can also modify the contents of the input file. For example, you can change a material property or the magnitude of a load.



Warning:

If you edit the input file for a model using a text editor outside Abaqus/CAE and then submit the job for that model in the Job module, your changes to the input file will be lost. Instead, you must submit the modified input file directly for analysis by creating a new job and selecting **Input file** as the job **Source**. However, if you use the **Keywords Editor** to modify the generated keywords for a model, those modifications are retained in the model and apply to any jobs associated with that model. (You can display the **Keywords Editor** by selecting **Model->Edit Keywords** from the main menu bar.)

Additional information

- *Adding unsupported keywords to your Abaqus/CAE model*
- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Performing a data check on a model

A data check runs the input file through the analysis input file processor to ensure that the model is consistent and that all the required model options have been set.
This option generates (or regenerates) the input file for the job if the job source is an Abaqus/CAE model.

To perform a data check on a model, select **Job->Data Check->*job of your choice*** from the main menu bar. You can also perform a data check by selecting the job of your choice and then clicking **Data Check** in the **Job Manager**.

To see the results of a data check analysis, review the data (`.dat`) file in your working directory. You can also review the results of a data check by selecting **Job->Monitor->*job of your choice*** to display the job monitor dialog box. For more information on the job monitor, see *Monitoring the progress of an analysis job*.

A data check also creates an output database. You can view the output database in the Visualization module to verify model information such as face labels, surface normals, and material orientations. For more information on viewing an output database, see *Viewing results*.

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Submitting an analysis job

To submit a job for analysis, select **Job**->**Submit**->***job of your choice*** from the main menu bar. You can also submit a job for analysis by selecting the job name in the **Job Manager** and clicking **Submit**. Abaqus/CAE submits the job for analysis using the job settings defined in the job editor.

For information on monitoring a job that you have submitted, see *Monitoring the progress of an analysis job*.

When you start an Abaqus/CAE session and submit an analysis job, Abaqus sends messages to Abaqus/CAE indicating the current status of the job, and Abaqus/CAE updates the **Status** column in the **Job Manager** accordingly.

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Continuing an analysis job after a data check

When you perform a data check analysis (see *Performing a data check on a model*), Abaqus creates and saves all the files necessary to continue the full analysis at a later time. To continue an analysis, select **Job->Continue->*job of your choice*** from the main menu. You can also continue an analysis by selecting the job name in the **Job Manager** and clicking **Continue**. Abaqus/CAE continues the analysis using the job settings defined in the job editor.

For information on monitoring a job that you have submitted, see *Monitoring the progress of an analysis job*.

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Terminating an analysis job

You can use one of the following methods to terminate an analysis job:

- From the main menu bar, select **Job->Kill->*job of your choice***.
- In the **Job Manager**, select the job name and then click **Kill**.
- In the job monitor dialog box for that particular job, click **Kill**. (For more information, see *Monitoring the progress of an analysis job*.)

Abaqus/CAE asks you for confirmation, terminates the job, and updates the status of the job in the **Job Manager** to `Terminated`.

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Viewing the results of your job

Once your analysis job is completed, Abaqus/CAE stores the results of the analysis in an output database. You can use the Visualization module to view these results graphically. Using the **Results** command from the **Job** menu starts the Visualization module and gives you a basic plot of your model.

1. From the main menu bar, select **Job->Results->*job of your choice***.

   Tip: You can also select the job name in the **Job Manager** and click **Results**.

   The Visualization module starts and presents you with a plot of the model.

   Note:

   Using the **Results** command exits the Job module. To reenter the Job module, select **Job** from the **Module** list located in the context bar.

2. Use the Visualization module to create and customize different plots of your results. For more information on using the Visualization module, see *Viewing results*."

Additional information

- *Understanding analysis jobs*
- *Creating, editing, and manipulating jobs*

Using the job editor

This section describes how you use the job editor to configure the settings of a job.

In this section:

- *Navigating the job customization options*
- *Configuring job submission attributes*
- *Choosing the job type*
- *Setting the license type for the SimUnit license model*
- *Choosing the job run mode*
- *Setting the job submit time*
- *Specifying general job settings*
- *Controlling job memory settings*
- *Controlling job parallel execution*
- *Controlling job precision*

Navigating the job customization options

Use the job editor to customize a job's settings before submitting it for analysis. To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. The job editor contains the following tabbed pages:

Submission

Use the **Submission** tabbed page to configure the basic attributes of your job, such as the job type, the run mode, and the run time. You can also use the submission page to submit jobs to a *remote queue*.

General

Use the **General** tabbed page to configure preprocessor printout and to specify a scratch directory. Preprocessor printout options are not available for a job associated with an input file; you must specify these options in the input file itself. You also use the **General** tabbed page to specify any user subroutines that are referred to by the model and the output format of the results.

Memory

Use the **Memory** tabbed page to configure the amount of memory allocated to an Abaqus analysis.

Parallelization

Use the **Parallelization** tabbed page to configure the parallel execution of an Abaqus analysis job, such as the number of processors to use.

Precision

Use the **Precision** tabbed page to specify either single or double precision for Abaqus/Explicit analyses. You can also choose the precision of nodal output that is written to the output database during the analysis.

You can also use the Abaqus environment file (`abaqus_v6.env`) to control the default value of many of the settings in the job editor tabbed pages. For more information, see *Environment File Settings*.

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Abaqus/Standard and Abaqus/Explicit Execution*

Configuring job submission attributes

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Submission** tabbed page in the job editor to configure attributes related to job submission. You can do the following:

- Choose the type of analysis. (For more information, see *Choosing the job type*.)
- Choose the type of license. (For more information, see *Setting the license type for the SimUnit license model*.)
- Choose the run mode. (For more information, see *Choosing the job run mode*.)
- Choose the submit time. (For more information, see *Setting the job submit time*.)

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*
- *Abaqus/Standard and Abaqus/Explicit Execution*

Choosing the job type

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar.
Use the **Job Type** options in the **Submission** tabbed page to choose the type of job that Abaqus will run. You can choose one of the following options:

Full analysis

Select this option to submit a complete model or input file for analysis.

Recover (Explicit)

Select this option to resubmit an Abaqus/Explicit analysis that has previously terminated prematurely. This option is not valid for Abaqus/Standard analyses.

Restart

Select this option to submit an analysis of a model that uses saved data from a previous analysis of the same model. The model attributes must specify the location of the saved data.

The available job types are covered in more detail in *Selecting a job type*.

1. In the job editor, click the **Submission** tab to display the **Submission** tabbed page.
2. From the **Job Type** options at the top of the tabbed page, select the job type of your choice.

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*

Setting the license type for the SimUnit license model

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **SIM Unit Licensing** options in the **Submission** tabbed page to choose license options for your analysis job execution.

The **SIM Unit Licensing** options are enabled when you use the SimUnit license model or when Abaqus/CAE is launched by the **3D**EXPERIENCE Platform Connector for Abaqus/CAE. For all other cases, these options are disabled. If you are using the Abaqus Learning Edition, this option is not available.

1. In the job editor, click the **Submission** tab to display the **Submission** tabbed page.
2. When using the SimUnit license model, from the **SIM Unit Licensing** options in the middle of the page, choose one of the following:

Default

The job executes with the default license type in the Abaqus environment file. The **license_type** parameter value is used, if applicable. If you do not specify the parameter in an Abaqus environment file, tokens are used.

Tokens

The job executes using tokens. This selection overrides the **license_type** parameter in the Abaqus environment file.

Credits

The job executes using credits. This selection overrides the **license_type** parameter in the Abaqus environment file.

3. If Abaqus/CAE is launched by the **3D**EXPERIENCE Platform Connector for Abaqus/CAE, you see an additional option to choose the License Server:

3DExperience On-Cloud

The job executes using the ***3D****EXPERIENCEPlatform* license server.

Local

The job executes using the locally installed license server.

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*

Choosing the job run mode

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Run Mode** options in the **Submission** tabbed page to choose the mode in which Abaqus will run your job. You can choose one of the following options:

- **Background** to run your job locally in the background.
- **Queue** to submit your job to a named local or remote batch queue; select the batch queue from the pull-down list. (You must define the queue in the Abaqus environment file. For more information, see *Defining analysis batch queues*. You must also make the queue available to Abaqus/CAE. For more information, see *Submitting a job remotely*.)

The **Run Mode** setting is analogous to parameters of the Abaqus execution procedure.

1. Locate the job editor.

   From the main menu bar, select **Job->Edit->*jobname***.

   Abaqus/CAE displays the **Edit Job** dialog box.
2. Click the **Submission** tab to display the **Submission** tabbed page.
3. From the **Run Mode** options in the middle of the page, select either **Background** or **Queue**.

Additional information

- *Understanding analysis jobs*
- *Submitting a job remotely*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*
- *Abaqus/Standard and Abaqus/Explicit Execution*
- *Environment File Settings*

Setting the job submit time

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Submit Time** options in the **Submission** tabbed page to choose when your analysis job will execute.

1. In the job editor, click the **Submission** tab to display the **Submission** tabbed page.
2. From the **Submit Time** options at the bottom of the page, choose one of the following options:

Immediately

The job executes in the background of your local machine immediately or is submitted to the batch queue immediately.

Wait

The job executes in the background of your local machine after a waiting period. Type the waiting period in the hours and minutes fields. (This option is available only on Linux platforms. In addition, it is not available if the job is being submitted to a batch queue.)

At

The job executes at a time that you specify in either the background of your local machine or in a batch queue. Click  to the right of the **At** field for information on what syntax to use when specifying a time. If you want the job to run in a batch queue at the specified time, you must configure the queue to accept a time variable using the **after_prefix** queue parameter in the Abaqus environment file. For more information, see *Defining analysis batch queues*. (The **At** option is available for jobs that run in either the background or in a batch queue on Linux platforms. However, the **At** option is available only for jobs that execute in a batch queue on Windows platforms.)

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*

Specifying general job settings

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **General** tabbed page to configure miscellaneous job settings. You can configure the following:

Preprocessor Printout

The **Preprocessor Printout** options allow you to control whether Abaqus prints an echo of the input data, contact constraints, model definition data, and history data to the data (`.dat`) file. By default, each of these options is toggled on.

Preprocessor printout options are not available for a job associated with an input file; you must specify these options in the input file itself.

Scratch directory

The **Scratch directory** option allows you to specify the name of the directory used for scratch files. On Linux systems the default scratch directory is the value of the `$TMPDIR` environment variable or `/tmp` if the variable is not defined. On Windows systems the default scratch directory is the value of the `TEMP` environment variable or `\TEMP` if the variable is not defined. To specify a scratch directory, you can do one of the following:

- Click in the **Scratch directory** text field, and type the directory path.
- Click  to display the **Select Scratch Directory** dialog box, and select the directory of your choice.

User subroutine file

Provide the name of the file containing all user subroutines that are referred to by the model. To specify a user subroutine file, you can do one of the following:

- Click in the **User subroutine file** text field, and type the file path.
- Click  to display the **Select User Subroutine File** dialog box, and select the file of your choice.

If your model refers to a user subroutine but you do not specify the name of the subroutine file in the **General** tabbed page, Abaqus generates an error that is reported by the job monitor dialog box. (You can display the job monitor dialog box by selecting **Job->Monitor->*job of your choice*** from the main menu bar.) For more information on subroutines, see *About User Subroutines and Utilities*.

Results Format

The **Results Format** options allow you to write the results from an Abaqus analysis in ODB format or SIM format. For an Abaqus/Standard or Abaqus/Explicit analysis, you can also write the results to both formats. For more information, see *The Output Database*.



Note: If you select the ODB format, a `.sim` file may still be created; however, it will not contain the analysis results.

You can use the Abaqus environment file (`abaqus_v6.env`) to control the default value of most of the settings in the **General** tabbed page; for more information, see *Environment File Settings*.

Additional information

• *Understanding analysis jobs*

- *Using the job editor*
- *Creating, editing, and manipulating co-executions*

Controlling job memory settings

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Memory** tabbed page to control the amount of memory allocated to an Abaqus/Standard or Abaqus/Explicit analysis. Abaqus/CAE initially populates the settings in the **Memory** tabbed page with the values specified in the Abaqus environment file (`abaqus_v6.env`), and you can edit the environment file if you want to change the default memory settings. For more information about customizing the memory settings, see *Environment File Settings*.

If no memory values are specified in the Abaqus environment file, Abaqus automatically detects the physical memory on the machine and allocates a percentage of this available memory. The default percentages are platform specific, but they typically represent a large portion of the available physical memory. You should check the memory allocation values that Abaqus selects upon your next submission of the job to confirm that the allocation is appropriate for your system and for the job. For details on the default memory allocation settings, refer to the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Smaller jobs may use less memory than the maximum available amount. You might want to reduce the memory allocation for your jobs or adjust the default allocation if you usually run multiple jobs at the same time. You can control the following:

- The **Memory allocation units** for the memory size specification. You can allocate memory as a **Percent of physical memory** or in **Megabytes** or **Gigabytes**.
- The **Maximum preprocessor and analysis memory** value. This option specifies either the percentage of physical memory or the number of megabytes or gigabytes of physical memory that will be allocated as a maximum for the analysis.
- The **Increase memory allocation based on analysis estimates** option. This setting enables you to use the memory allocation setting that Abaqus determines to be suitable for this job, with an additional amount of memory as a buffer, as the memory allocation settings for all future submissions of this job. The job memory value will be updated only if the estimate is higher than the value you specified.

For more information about memory settings, see *Managing Memory and Disk Resources*.

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*

Controlling job parallel execution

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Parallelization** tabbed page to control the parallel execution of an Abaqus analysis job, as follows:

- Select the number of processors to use for the analysis if parallel processing is available.
- For an Abaqus/Standard analysis, you can choose to use GPGPU acceleration of the direct sparse solver and specify the number of GPGPUs. For more information, see *Parallel Execution in Abaqus/Standard*.
- For an Abaqus/Explicit analysis, you can choose the following:
  - The number of domains. If you choose domain-level parallelization, the number of domains must be equal to or a multiple of the number of processors.

    When the number of domains is a multiple of the number of processors, you can toggle on **Activate dynamic load balancing** if desired. This option may improve the performance of your analysis job. If you have 4 processors and choose 8, 12, 16, etc. domains, for example, the dynamic load balancing feature becomes available. Dynamic load balancing is turned off by default. This option is equivalent to the `dynamic_load_balancing` option in the Abaqus/Explicit execution procedure. For more information, see *Abaqus/Standard and Abaqus/Explicit Execution*.

    Dynamic load balancing is most likely to improve the computational speed and efficiency in applications with a strongly time-dependent and/or spatially varying computational load. Most imbalanced problems will see an optimal performance improvement when the number of domains is two to four times the number of processors. For more details, see *Domain-Level Parallelization*.
  - Whether the parallelization method should be **Domain** (default) or **Loop**. The domain-level method splits the model into a number of topological domains that are distributed evenly among the available processors. The loop-level method parallelizes low-level loops in the code that are responsible for most of the computational cost.

    If you select **Loop** for the parallelization method, you must select **Default** for the multiprocessing mode. For more information, see *About Parallel Execution*.

  For more information, see *Parallel Execution in Abaqus/Explicit*.
- For both an Abaqus/Standard and an Abaqus/Explicit analysis, you can choose whether the multiprocessing mode should be **Default**, **Threads**, **MPI** (Message Passing Interface), or **Hybrid**. The default multiprocessing mode is dependent on the platform on which the analysis product executes.
  - For the **Hybrid** multiprocessing mode, you can choose the number of threads per MPI process.
  - You cannot choose the number of threads per MPI process for all other multiprocessing modes. For **MPI** and **Default**, the number of threads per MPI process has value of 1. For **Threads**, the number of threads per MPI process is equal to the number of processors.



Note:

You use the Step module to choose between the Abaqus/Standard iterative and direct sparse solvers. For more information, see *Using the step editor*.

Additional information

- *Understanding analysis jobs*
- *Using the job editor*
- *Creating, editing, and manipulating co-executions*
- *Parallel Execution*

Controlling job precision

To locate the job editor, select **Job->Edit->*jobname*** from the main menu bar. Use the **Precision** tabbed page to control the precision of the Abaqus/Explicit analysis or the precision of nodal output for an Abaqus/Standard or Abaqus/Explicit analysis. You can do the following:

- Select the single, forced single (override environment variable **double_precision** setting if necessary), or double (analysis only, constraints only, or analysis and packager) setting for Abaqus/Explicit precision. This option is equivalent to the `double` option in the Abaqus/Explicit execution procedure.
- Select either single or full nodal output precision. This option is equivalent to the `output_precision` option in the Abaqus execution procedure.

For more information, see *Abaqus/Standard and Abaqus/Explicit Execution*, and *Environment File Settings*.

Additional information

• *Understanding analysis jobs*
• *Using the job editor*
• *Creating, editing, and manipulating co-executions*
• *Abaqus Execution Guide*

Creating, editing, and manipulating adaptivity processes

This section describes how you create, edit, and manipulate adaptivity processes using either the main menu or the **Adaptivity Process Manager**.

In this section:

- *Creating a new adaptivity process*
- *Performing a data check on an adaptivity process*
- *Submitting an adaptivity process*
- *Continuing an adaptivity process after a data check*
- *Terminating an adaptivity process*

Creating a new adaptivity process

An adaptivity process is a succession of analysis jobs. Abaqus/CAE modifies the mesh in each analysis job in response to error estimates that were computed from data written to the output database during the final increment of the previous analysis. For more information, see *What is an adaptivity process?*. To create a new adaptivity process, select **Adaptivity->Create** from the main menu bar.

1. From the main menu bar, select **Adaptivity->Create**.

   The **Edit Adaptivity Process** dialog box appears.

   Tip: You can also create a new adaptivity process by clicking the icon in the module toolbox or by clicking **Create** in the **Adaptivity Process Manager**.

2. Type the name of the new adaptivity process in the **Name** text field. The name that you specify must adhere to the file naming rules for your operating system.
3. Select the model to associate with the adaptivity process from the **Model** field.
4. If desired, type the description of the adaptivity process in the **Description** text field. The description is stored in the model database and the output database, and you can use this description to help you manage your adaptivity processes.
5. Enter the job prefix in the **Job Prefix** text field. The job name for each iteration of the adaptivity process will start with this prefix. The prefix that you specify must adhere to the file naming rules for your operating system. If you do not provide a job prefix, Abaqus/CAE uses the name of the adaptivity process.
6. In the editor, enter all data necessary to define the process and click **OK**. (For more information, see *Using the adaptivity process editor*.)

Additional information

- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*

Performing a data check on an adaptivity process

A data check runs the input file for a model through the analysis input file processor to ensure that the model is consistent and that all the required model options have been set. Performing a data check on an adaptivity process will create a job and verify model information for the first iteration of the overall adaptivity process.

To perform a data check on an adaptivity process, select **Adaptivity**->**Data Check**->***adaptivity process name*** from the main menu bar. You can also perform a data check by selecting the process name in the **Adaptivity Process Manager** and clicking **Data Check**.

The job created by the data check appears in the **Jobs** container using the naming conventions discussed in *Creating a new adaptivity process*. To see the results of a data check, review the data (`.dat`) file for this job in your working directory. You can also review the results of a data check by selecting **Job**->**Monitor**->***created job*** from the main menu bar to display the job monitor dialog box.

Additional information

- *Monitoring the progress of an analysis job*
- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*

Submitting an adaptivity process

To submit an adaptivity process for analysis, select **Adaptivity**->**Submit**->***adaptivity process name*** from the main menu bar. (You can also submit an adaptivity process for analysis by selecting the process name in the **Adaptivity Process Manager** and clicking **Submit**.) Abaqus/CAE submits the process for analysis and remeshing using the settings defined in the adaptivity process.

When you start an Abaqus/CAE session and submit an adaptivity process, Abaqus/Standard sends messages to Abaqus/CAE indicating the current status of the process. Abaqus/CAE then updates the **Status** column in the **Adaptivity Process Manager** accordingly. You must keep the current Abaqus/CAE session active for automatic remeshing to occur between analysis jobs. If you exit Abaqus/CAE and start a new session, you must use manual remeshing to continue the adaptivity process. For more information, see *When do I need to use manual adaptive remeshing?*.

In addition, Abaqus/CAE displays status information in the message area after each iteration of the adaptivity process. The status information lists the error indicator variables in each remeshing rule and whether the remeshing rule has been satisfied for that variable.

Additional information

- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*

Continuing an adaptivity process after a data check

When you perform a data check on an adaptivity process, a job is created for the first iteration of the adaptivity process (as described in *Performing a data check on an adaptivity process*). In addition, Abaqus creates and saves all the files necessary to complete this job at a later time. You can also use these files to continue the full adaptivity process from the current state of the first iteration.

To complete only the job associated with the first iteration of the adaptivity process, use the procedures listed in *Continuing an analysis job after a data check*. Remeshing and further iterations will not occur.

To continue the full adaptivity process, select **Adaptivity->Continue Analysis->*adaptivity process name***. You can also continue the adaptivity process by selecting the process name in the **Adaptivity Process Manager** and clicking **Continue Analysis**. The adaptivity process proceeds as described in *Submitting an adaptivity process*.

The **Continue Analysis** option can be used only in conjunction with a data check; it is not used to continue remeshing after an adaptivity process is complete. Once an adaptivity process runs to completion, you can continue to remesh a model by resubmitting the full process. You can also consider using the combination remeshing techniques described in *Using a combination of automatic and manual mesh adaptivity*.

Additional information

- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*

Terminating an adaptivity process

To terminate an adaptivity process, you should use the Job Manager to display the analysis job that is running and to end the job. Ending the job will terminate the adaptivity process. For more information, see *Terminating an analysis job*.

Additional information

- *Understanding adaptivity processes*
- *Creating, editing, and manipulating adaptivity processes*

Using the adaptivity process editor

This section describes how you use the adaptivity process editor to configure the settings of an adaptive remeshing process.

In this section:

- *Navigating the adaptivity process customization options*
- *Configuring adaptivity process attributes*
- *Specifying general adaptivity process settings*
- *Controlling adaptivity process memory settings*
- *Controlling adaptivity process parallel execution*
- *Controlling adaptivity process precision*

Navigating the adaptivity process customization options

Use the adaptivity process editor to customize your adaptivity processes before submitting them. The settings are analogous to parameters of the Abaqus execution procedure. For more information, see *Abaqus/Standard and Abaqus/Explicit Execution*.

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. The adaptivity process editor contains the following tabbed pages:

Adaptivity

Use the **Adaptivity** tabbed page to specify the maximum number of analysis iterations and to select the run mode. For more information, see *Configuring adaptivity process attributes*.

General

Use the **General** tabbed page to configure preprocessor printout and to specify a scratch directory. You also use the **General** tabbed page to specify any user subroutines that are referred to by the model and the output format of the results. For more information, see *Specifying general adaptivity process settings*.

Memory

Use the **Memory** tabbed page to configure the amount of memory allocated to each of the adaptivity analyses. For more information, see *Controlling adaptivity process memory settings*.

Parallelization

Use the **Parallelization** tabbed page to configure the parallel execution of an Abaqus analysis job, such as the number of processors to use and the parallelization method. For more information, see *Controlling adaptivity process parallel execution*.

Precision

Use the **Precision** tabbed page to specify the precision of nodal output that is written to the output database during each analysis. For more information, see *Controlling adaptivity process precision*.

The settings that you specify in the adaptivity process editor apply to each of the analyses that runs after each iteration of the adaptive remesh.

You can also use the Abaqus environment file (`abaqus_v6.env`) to control the default value of many of the settings in the adaptivity process editor tabbed pages. For more information, see *Environment File Settings*.

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Configuring adaptivity process attributes

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. Use the **Adaptivity** tabbed page in the adaptivity process editor to specify the maximum number of iterations and to select the run mode.

Choose either of the following to specify the run mode:

- **Background** to run your job locally in the background.
- **Queue** to submit your job to a named local or remote batch queue; select the batch queue from the pull-down list. (You must define the queue in the Abaqus environment file. For more information, see *Defining analysis batch queues*. You must also make the queue available to Abaqus/CAE. For more information, see *Submitting a job remotely*.)

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Specifying general adaptivity process settings

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. Use the **General** tabbed page to configure miscellaneous adaptivity process settings. You can configure the following:

Preprocessor Printout

The **Preprocessor Printout** options allow you to control whether Abaqus prints an echo of the input data, contact constraints, model definition data, and history data to the data (`.dat`) file. By default, each of these options is toggled off.

Scratch directory

The **Scratch directory** option allows you to specify the name of the directory used for scratch files. On Linux systems the default scratch directory is the value of the `$TMPDIR` environment variable or `/tmp` if the variable is not defined. On Windows systems the default scratch directory is the value of the `TEMP` environment variable or `\TEMP` if the variable is not defined. To specify a scratch directory, you can do one of the following:

- Click in the **Scratch directory** text field, and type the directory path.
- Click  to display the **Select Scratch Directory** dialog box, and select the directory of your choice.

User subroutine file

Provide the name of the file containing all user subroutines that are referred to by the model. To specify a user subroutine file, you can do one of the following:

- Click in the **User subroutine file** text field, and type the file path.
- Click  to display the **Select User Subroutine File** dialog box, and select the file of your choice.

If your model refers to a user subroutine but you do not specify the name of the subroutine file in the **General** tabbed page, Abaqus generates an error that is reported by the job monitor dialog box. (You can display the job monitor dialog box by selecting **Job->Monitor->*job of your choice*** from the main menu bar.) For more information on user subroutines, see *About User Subroutines and Utilities*.

Results Format

The **Results Format** options allow you to write the results from an Abaqus analysis in ODB format, SIM format, or both formats. For more information, see *The Output Database*.



Note: If you select the ODB format, a `.sim` file may still be created; however, it will not contain the analysis results.

You can use the Abaqus environment file (`abaqus_v6.env`) to control the default value of most of the settings in the **General** tabbed page; for more information, see *Environment File Settings*.

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Controlling adaptivity process memory settings

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. Use the **Memory** tabbed page to control the amount of memory allocated to an Abaqus analysis. Abaqus/CAE initially populates the settings in the **Memory** tabbed page with the values specified in the Abaqus environment file (`abaqus_v6.env`), and you can edit the environment file if you want to change the default memory settings. For more information about customizing the memory settings, see *Environment File Settings*.

If no memory values are specified in the Abaqus environment file, Abaqus automatically detects the physical memory on the machine and allocates a percentage of this available memory. The default percentages are platform specific, but they typically represent a large portion of the available physical memory. For details on the default memory allocation settings, refer to the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Smaller jobs may use less memory than the maximum available amount. You might want to reduce the memory allocation for your jobs or adjust the default allocation if you usually run multiple jobs at the same time. You can control the following:

- The **Memory allocation units** for the memory size specification. You can allocate memory as a **Percent of physical memory** or in **Megabytes** or **Gigabytes**.
- The **Maximum preprocessor and analysis memory** value that specifies either the percentage of physical memory or the number of megabytes or gigabytes of physical memory that will be allocated as a maximum for the analysis.

For more information about memory settings, see *Managing Memory and Disk Resources*.

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Controlling adaptivity process parallel execution

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. Use the **Parallelization** tabbed page to control the parallel execution of an Abaqus process job as follows:

- Select the number of processors to use for the analysis if parallel processing is available.
- Select the **Threads** multiprocessing mode. Only the linear equation solver in Abaqus/Standard will execute in parallel.



Note:

You use the Step module to choose between the Abaqus/Standard iterative and direct sparse solvers. For more information, see *Using the step editor*.

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Controlling adaptivity process precision

To locate the adaptivity process editor, select **Adaptivity->Edit->*adaptivity process name*** from the main menu bar. Use the **Precision** tabbed page to control the precision of nodal output. You can select either single or full nodal output precision. This option is equivalent to the `output_precision` option in the Abaqus execution procedure. For more information, see *Abaqus/Standard and Abaqus/Explicit Execution*.

Additional information

- *Understanding adaptivity processes*
- *Using the adaptivity process editor*
- *About Adaptive Remeshing*

Creating, editing, and manipulating co-executions

This section describes how you create, edit, and manipulate co-executions using either the main menu or the **Co-execution Manager**.

In this section:

- *Creating and editing co-executions*
- *Performing a data check on a co-execution*
- *Submitting a co-execution*
- *Viewing the results of your co-execution*
- *Terminating a co-execution*

Creating and editing co-executions

A co-execution is the co-simulation execution of two analysis jobs. You specify job parameters in a co-execution to create both analysis jobs. After creating the co-execution, you can edit the job parameters only in the individual analysis jobs.

The jobs are executed in synchronization with one another using the same functionality as described in *Understanding analysis jobs*. For more information on this analysis technique, see *Structural-to-Structural Co-Simulation*.

1. From the main menu bar, select **Co-execution->Create** or **Co-execution->Edit->*co-execution name***.

   The **Edit Co-execution** dialog box appears.

   Tip: You can also create a co-execution by clicking the icon in the module toolbox or clicking **Create** in the **Co-execution Manager**.

2. Type the name of the new co-execution in the **Name** text field. The name that you specify must adhere to the file naming rules for your operating system.
3. If desired, type the description of the co-execution in the **Description** text field. The description is stored in the model database and the output database, and you can use this description to help you manage your co-executions.
4. In the **Models** table, select the models in the current model database to associate with the co-execution from the list in the **Model Name** column.

   The analysis product used in each of the selected models and the default job name for each model are displayed in the table.

5. If desired, click in the **Job Name** column and edit the job name.
6. Use the **Submission**, **General**, **Memory**, **Parallelization**, and **Precision** tabbed pages to define the initial job parameter settings in the individual job editors. After the co-execution is created, the job parameters in the co-execution editor are unavailable for editing; you must edit the individual jobs.

   Define the initial job parameter settings as described in the following sections:

   - *Configuring job submission attributes*
   - *Specifying general job settings*
   - *Controlling job memory settings*
   - *Controlling job parallel execution*
   - *Controlling job precision*

7. If desired, enter a different value in the **Communication time out** field to specify the time after which Abaqus/CAE terminates if it does not receive any communication from the coupled analyses. The default value is 10 minutes.
8. Click **OK**.

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

Performing a data check on a co-execution

A data check runs the input files for the models through the analysis input file processor to ensure that the models are consistent and that all the required model options have been set. Performing a data check on a co-execution will verify model information in each of the jobs created for the co-execution.

To perform a data check on a co-execution, select **Co-execution**->**Data Check**->***co-execution name*** from the main menu bar. You can also perform a data check by selecting the co-execution name in the **Co-execution Manager** and clicking **Data Check**.

The jobs involved in the co-execution appear in the Model Tree in the **Jobs** container under the co-execution in the **Co-executions** container. You can review the results of a data check by clicking mouse button 3 on the job in the Model Tree and selecting **Monitor** to display the job monitor dialog box. For more information, see *Monitoring the progress of an analysis job*.

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

Submitting a co-execution

To submit a co-execution, select **Co-execution->Submit->*co-execution name*** from the main menu bar. (You can also submit a co-execution by selecting the co-execution name in the **Co-execution Manager** and clicking **Submit**.) Abaqus/CAE submits the jobs for analysis using the job settings defined in the co-execution editor or the individual job editors.

The jobs involved in the co-execution appear in the Model Tree in the **Jobs** container inside the **Co-executions** container. You can monitor the jobs that you have submitted by clicking mouse button 3 on the job in the Model Tree and selecting **Monitor** to display the job monitor dialog box. For more information, see *Monitoring the progress of an analysis job*.

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

Viewing the results of your co-execution

Once your co-execution is completed, Abaqus/CAE stores the results of each analysis job in an output database. You can use the Visualization module to view the combined results graphically.

1. From the **Co-execution Manager**, select the co-execution name and click **Results**.

   The Visualization module starts and presents you with an overlay plot of both models.

2. Use the Visualization module to create and customize different plots of your results. For more information on using overlay plots, see *Overlaying multiple plots*.

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

Terminating a co-execution

To terminate a co-execution, you should use the **Co-execution Manager** to display the co-execution that is running and to end the analysis jobs. Ending the jobs terminates the co-execution. Abaqus/CAE asks you for confirmation, terminates the jobs, and updates the status of the co-execution in the **Co-execution Manager** to `Terminated`. Note that co-execution termination may result in various error messages written to the message area, reflecting the unpredictable manner in which co-execution inter-job messaging is terminated. These error messages occurring immediately following co-execution termination can generally be disregarded

Additional information

- *Understanding co-executions*
- *Creating, editing, and manipulating co-executions*

Creating, editing, and manipulating optimization processes

This section describes how you create, edit, and manipulate optimization processes using either the main menu or the **Optimization Process Manager**.

In this section:

- *Creating and editing optimization processes*
- *Creating the optimization files*
- *Validating an optimization process*
- *Submitting an optimization process*
- *Continuing an optimization process that terminated*
- *Monitoring your optimization process*
- *Extracting a smoothed mesh*
- *Combining optimization results*
- *Viewing the results of your optimization process*

Creating and editing optimization processes

An optimization process reads an optimization task that you defined in the Optimization module and iteratively searches for an optimized solution based on the objective functions and constraints that you defined in the optimization task.

1. From the main menu bar, select **Optimization->Create** or **Optimization->Edit->*optimization process name***.

   The **Edit Optimization Process** dialog box appears.

   Tip: You can also create an optimization process by clicking the  icon in the module toolbox or clicking **Create** in the **Optimization Process Manager**.

2. Type the name of the new optimization process in the **Name** text field. The name that you specify must adhere to the file naming rules for your operating system. A directory is created with this name, and the output from the optimization process is stored in this directory.
3. Select the model to associate with the optimization process from the **Model** field.
4. Select the optimization task to associate with the optimization process from the **Optimization Task** field.

   If desired, type the description of the optimization process in the **Description** text field. The description is stored in the model database and the output database, and you can use this description to help you manage your optimization processes.
5. Enter the maximum number of optimization design cycles that are allowed before Abaqus ends the optimization process.
6. Choose how often to write the analysis results to the output database file that is created by the Abaqus job that runs for each design cycle of the optimization process. This option also specifies how often to create the related Abaqus analysis files, such as the status file and the message file. You can choose the following:

   - **Initial**: save data after the initial cycle (n=0) . The Optimization module uses the initial cycle to prepare the model for the first design cycle; for example, by creating groups and output requests. A finite element analysis is not performed during the initial cycle.
   - **First**: save data from the first design cycle (n=1).
   - **Last**: save data from the last design cycle.
   - **Every *n* cycles**: save data every n design cycles relative to the initial (n=0) design cycle. For example, if n=3, the optimization process will save data corresponding to design cycle 2, 5, 8, etc.
   - **Every**: Save data after every design cycle, including the initial (n=0) design cycle.

     Note: Selecting this option could lead to large disk space requirements for Abaqus models that generate large output database files.

   For more information, see *Understanding the files generated by creating and analyzing a model*. To postprocess the results of an optimization, you must combine the output databases that were generated by each design cycle into a single output database. For more information, see *Combining optimization results*.
7. Use the **Submission**, **Memory**, and **Parallelization** tabbed pages to define the initial settings for each Abaqus job that runs during the optimization process, as described in the following sections:

   - *Configuring job submission attributes*

- *Controlling job memory settings*
- *Controlling job parallel execution*

8. Click **OK**.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Creating the optimization files

You can create copies of the optimization parameter (`.par`) file and the Abaqus input (`.inp`) file in your working directory. The parameter file contains parameters that are used to execute the optimization and contains information about the input file that is associated with the optimization; the input file, in turn, defines the Abaqus model you are optimizing.

You can run the optimization from the command line with the following command:

```
abaqus optimization -task parameter file -job results folder
```

The parameter file and the input file must be saved in the same directory. During the optimization, Abaqus creates a folder called *results folder* in which the results of the optimization are stored.

To create the optimization files, select **Optimization->Write Files->*optimization process name*** from the main menu bar. You can also create the optimization files by selecting the process name in the **Optimization Process Manager** and clicking **Write Files**.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Validating an optimization process

A validation checks the optimization process to ensure that the optimization task has been configured correctly and the Abaqus model exists. A validation does not check your Abaqus model. You should run a complete analysis of your model and make sure that it runs to completion before you attempt to run an optimization process.

To validate an optimization process, select **Optimization->Validate->*optimization process name*** from the main menu bar. You can also validate an optimization process by selecting the process name in the **Optimization Process Manager** and clicking **Validate**.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Submitting an optimization process

To submit an optimization process, select **Optimization**->**Submit**->***optimization process name*** from the main menu bar. (You can also submit an optimization process by selecting the optimization process name in the **Optimization Process Manager** and clicking **Submit**.) Abaqus/CAE submits the optimization process using the settings defined in the optimization process editor. Click **Monitor** in the **Optimization Process Manager** to monitor the optimization process. For more information, see *Monitoring your optimization process*.

The optimization process appears in the Model Tree. The **Optimization Process** container includes a **Jobs** container that includes the Abaqus jobs executed during the optimization process. You can monitor an Abaqus job that is running by clicking the **Monitor** button from the top of the optimization process monitor. If your optimization process includes multiple models, the Optimization module creates an analysis job for each model and you must select the desired job to monitor. You can also view the monitor for the Abaqus jobs that completed during previous design cycles by clicking mouse button 3 on the job in the Model Tree and selecting **Monitor**. To terminate an optimization process, you must terminate the current Abaqus job. For more information, see *Monitoring the progress of an analysis job*.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Continuing an optimization process that terminated

You can restart an optimization process that has failed for a reason external to the optimization. To continue the optimization process, select **Optimization->Restart->*optimization process name***. You can also continue the optimization process by selecting the optimization process name in the **Optimization Process Manager** and clicking **Restart**.

You can restart an optimization process only if it has failed because of a problem external to the optimization or the Abaqus analysis, such as:

- A license was not available.
- The Abaqus analysis failed due to insufficient disk space or memory.
- The user terminated the Abaqus analysis job by clicking the **Kill** button from the job monitor.

If the optimization fails due to errors in your Abaqus model, performing a restart will not allow it to continue. For example, if Abaqus stops because of excessively distorted elements, the same problem will occur after you attempt to restart the optimization. Before you restart the process, you can use the optimization process editor to change the optimization task and the model referred to by the optimization process. In addition, you can change the maximum number of optimization design cycles that are allowed before Abaqus ends the optimization process.

The optimization always restarts with an Abaqus analysis using the model that was most recently modified.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Monitoring your optimization process

You can monitor the progress of your optimization process while it is running by selecting **Optimization->Monitor->*optimization process name***. You can also monitor the optimization process by selecting the optimization process name in the **Optimization Process Manager** and clicking **Monitor**. In addition, you can monitor the Abaqus jobs that completed during previous design cycles by clicking mouse button 3 on the job in the Model Tree and selecting **Monitor**.

The optimization process monitor displays a table showing the optimization values after each design cycle, as described in *Understanding the files generated by an optimization process*. By default, the table includes only the objective function and any constraints in the optimization process. Abaqus/CAE adds a row to the table when a design cycle completes. You can click mouse button 3 on a cell in the table and do the following:

- Select **Hide** to remove the selected column from the table.
- Select **Edit visible columns** and choose the variables to display. *Table 1* describes the variables that can be displayed.

Table 1: Variable categories

Category	Description
Objective Function	The value of the objective function—a single design response or a weighted combination of design responses and reference values.
Objective Function-Design Response	The value of a design response that is used by the objective function.
Objective Function Term	The weighted value of a design response and its reference value.
Constraint	The value of a constraint.
Variable	The value of each design response.

See *Objectives and Constraints*, for more information about the formulation of the objective function from a combination of weighted design responses and reference values.

To view an *X–Y* plot showing the change in the value of the design responses after each design cycle, click the **Plot** button in the upper left corner of the optimization process monitor. The *X–Y* plot appears in a new viewport and includes every design response listed in the table. You can use the *X–Y* plot options to customize the appearance of the plot, as described in *X–Y plotting*.

The data that are displayed in the table or in the *X–Y* plot are read from the files `optimization_status_all.csv` and `optimization_report.csv`. The Optimization module must be able to write to these files while the optimization process is executing. For example, you cannot open the files in an application such as Microsoft Excel while the optimization process is executing.

The optimization process monitor also displays the following information:

- The **Log** tabbed page displays the start and end times for the optimization process along with a log of the progress of the process.
- The **Errors** and **Warnings** tabbed pages display the errors or the warnings associated with the optimization process. Abaqus/CAE indicates the presence of errors or warnings by prepending an exclamation point before the **Errors** and **Warnings** tab.

  The number of error and warning messages that appear in the job monitor is limited by the environment parameters **cae_error_limit** and **cae_warning_limit**, respectively (see *Job customization parameters*, for details). If the number of errors or warnings exceeds the optimization process limit, consult the data, message, or status file for a complete list of messages.

- The **Output File** tabbed page displays a detailed record of each output data entry as it is written to the output file, including status, warning, and error messages. The output file is called `atom.out`, and it is stored in the optimization process directory (the directory with the same name as the optimization process).

To monitor the Abaqus analysis job associated with the optimization process, click the **Monitor** button in the upper right corner of the optimization process monitor. (If your optimization process includes multiple models, the Optimization module creates an analysis job for each model and you must first select the desired job to monitor.) Abaqus/CAE displays the job monitor and continues to update the job monitor for each iteration of the optimization. If desired, you can terminate the Abaqus analysis job and, hence, the optimization process, from the job monitor. For more information, see *Continuing an optimization process that terminated*, and *Monitoring the progress of an analysis job*.

An optimization process outputs a large amount of information to the optimization process monitor, especially to the **Output File** tabbed page. To help you locate a particular text string, you can use the **Search Text** fields in the lower half of the optimization process monitor. Select the desired file tab, enter a search string in the **Text to find** field, and click **Next** or **Previous** to step through the output from one hit to the next. Toggle on **Match case** to perform a case-sensitive search.

The information presented in the optimization process monitor is updated continually as the optimization progresses. If the optimization fails, the **Errors** tabbed page appears in front of the other tabbed pages automatically to help you determine the cause of the failure. In addition, an exclamation point appears on the tab if any error or warning messages are output.

If you start monitoring an optimization process and then either exit Abaqus/CAE, close the current model database, or open a new model database, the optimization process will continue; but, you will no longer be able to monitor the optimization process from Abaqus/CAE.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Extracting a smoothed mesh

When the optimization process has completed, you can choose to extract a smooth isosurface meshed representation of the optimized model surface in the form of a file that can be transferred to a CAD system or back into Abaqus/CAE.

You can extract a smoothed surface mesh from either a topology, shape, or bead optimization. You can extract a single file from the entire assembly, or you can extract individual files from selected part instances.

The optimized model is usually a combination of tetrahedral and hexahedral elements and can include a significant number of interior elements that are not included in the isosurface representation that will be generated. In addition, the surface of the model might be rough and irregular, depending on the element density. The smoothing operation is an iterative process that displaces the nodes at the surface and generates a smooth triangular mesh on the surface of the model.

The smoothing operation can be followed by a data reduction operation that combines adjacent triangles with nearly coincident planes and reduces the number of surface triangles. Data reduction continues iteratively until the decrease in the number of triangles reaches a specified percentage of the original number of triangles. You can specify the angle that determines if adjacent planes are coincident.

Finally, you can choose to apply an optional filtering operation to remove local irregularities. The filtering is applied before the smoothing operation.

1. From the main menu bar, select **Optimization->Extract->*optimization process name***. You can also smooth the results by selecting the optimization process name in the **Optimization Process Manager** and clicking **Extract**.

   The **Extract Surface Mesh** dialog box appears.

2. Enter the name of the output file that will be extracted.
3. Select the content of the output files that will be extracted. You can choose either of the following:

   - Create a single file from the entire assembly.
   - Create individual files from selected part instances. Abaqus/CAE names the files by appending the name of the part instance to the name of the output file.

4. Select the format of the file that will be generated. You can choose from the following:

   - **Abaqus input file** (`.inp`): file that can be imported into Abaqus/CAE.
   - **STL** (`.stl`): file that can be read by 3D CAD systems, such as CATIA V5 and SOLIDWORKS.
   - **IGES1** (`.igs`): IGES file using block 114.
   - **IGES2** (`*_2.igs`): IGES file with explicit point information.
   - **IGES3** (`*_3.igs`): IGES file using block 128.

5. In the **Design Cycle** field, enter the design cycle from which the isosurface should be extracted. By default, the **Extract Surface Mesh** dialog box displays the last design cycle. The default value will not be displayed for sensitivity-based topology optimization since the solver is called once at the beginning of the optimization procedure.
6. Enter a value between zero and one specifying the **Iso value**.

   The smoothing process determines which elements are on the surface of the model and uses the isovalue to calculate where on the interior edges of the elements new nodes are created. Increasing the isovalue leads to shifting the isosurface toward the inside of the model, which results in a decrease in the model volume. For structures with thin components, you should not enter an isovalue greater than `0.7` to prevent the structure from becoming disconnected.

7. Enter the **Reduction percentage**, which defines the percent of faces that should be removed during the data reduction operation. A value of 0 indicates that no faces should be removed. A value of 100

indicates the data reduction stops when no more faces can be removed, given the value of the reduction angle.

8. Enter a value between 0 and 90 specifying the **Reduction angle**. The reduction angle defines the maximum angle between adjacent faces at a node such that the node might be removed during the data reduction.
9. Select the number of iterative smoothing cycles. Larger values lead to smoother models but can result in an increase of the computation time. In practice, 5 to 10 smoothing cycles are usually sufficient. Further smoothing can result in the narrowing or contracting of thin components. You can turn off smoothing by selecting a value of `0`.
10. Enter a value between zero and one specifying the **Target volume**.

    The target, or relative, volume of a solid model is the ratio of the volume defined by the isosurface and the original volume (calculated from the volume of the solid elements generated by the optimization). Likewise, the target area of a shell model is the ratio of the volume defined by the isosurface and the original area (calculated from the area of the shell elements generated by the optimization). If the model contains both solid and shell elements, only the solid elements are considered. The calculated volume does not account for voids inside a solid model.

    If you specify the target volume, the data reduction process calculates the isovalue that will result in the specified volume (the value of the isovalue that you enter is ignored). The data reduction process generates an error if you enter an excessively large value for the target volume that is larger than the volume calculated with an isovalue of zero or if you enter an excessively small value for the target volume that is smaller than the volume calculated with an isovalue of one. The data reduction process also generates an error if it fails to converge before 20 iterations.
11. In some cases, the optimized model will contain small irregularities that you can choose to remove with an optional filtering operation. Filtering is applied before the smoothing operation and results in a more homogeneous material distribution. Select **Moderate** to apply a single filtering cycle, select **Full** to apply five filtering cycles.
12. Click **OK** to extract the smoothed mesh.

    The specified output file is created in the TOSCA_POST directory. For subsequent runs, a unique number is appended to the file name; for example, *name*`_002.inp`.



Note:

If you extracted the smoothed mesh in the form of an Abaqus input file, you can select **File->Import->Model** from the main menu bar to import the mesh into Abaqus/CAE. For more information, see *Importing a model*.

Additional information

- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Combining optimization results

To view the results of an optimization in the Visualization module, you must combine the optimization results and analysis results into a single results output database file, as described in *Postprocessing an optimization*. The optimization results and the analysis results are appended to the output database file you specified as the base results.

1. From the main menu bar, select **Optimization->Combine->*optimization process name***. You can also combine optimization results by selecting the optimization process name in the **Optimization Process Manager** and clicking **Combine**.

   The **Combine Optimization Results** dialog box appears.

2. The **Optimization result directory** displays the path to the directory in which the output files from the selected optimization process reside. Click  to change the path.

3. Choose the output database file that will be used to create the base results. Abaqus/CAE uses the output database file defined in the base results as the source of the combined output database file. During the combination process, Abaqus/CAE appends additional data to the base results output database file.

   - Choose **Initial** to create the base results from the output database file generated during the initial optimization design cycle.
   - Choose **Last** to create the base results from the output database file generated during the last optimization design cycle. You should not choose **Last** if you performed a shape optimization.
   - Choose **Original Model** to create the base results from the output database file generated by running a separate analysis job of the model prior to the optimization. The optimization process does not run an Abaqus analysis of the original model. Therefore, you must run the analysis manually before you can select the output database file generated by the original model as the base results. (The Abaqus input file for the original model is stored in *optimization process name*`\SAVE.inp\`*optimization process name*`_org.inp`.)

4. Choose the output database files that will be combined with the base results output database file and the optimization results. The resulting output database file may be very large, so you should select only the data that you need to view in the Visualization module.

   - Choose **All** to combine the base results with the analysis results from all the output database files that were generated by the optimization process.
   - Choose **Last** to combine the base results with the analysis results from the output database file from the last design cycle.
   - Choose **Initial and last** to combine the base results with the analysis results from the output database files from the initial and last design cycles.
   - Choose **Every n cycles** and enter a value for *n* to combine the base results with the analysis results from the output database files created at periodic intervals starting with the initial design cycle.
   - Choose **Specify** to combine the base results with the analysis results from specified output database files; for example, `1,5,7,9`.

5. Select the **Models** and the **Steps** from within the models from which data will be combined into the results output database file.
6. From the list that appears, select the **Analysis Field Variables** that will be included in the results output database file.

   Abaqus/CAE always includes the optimization variables as field variables in the results output database file. The optimization data depend on the type of optimization that was performed:

7. Click **Submit** to create the results output database file.

Abaqus/CAE displays the **Combine Optimization Results Monitor** with the status of the merge operation and a log of output database files that are being merged.

8. Click **Close** to close the **Combine Optimization Results Monitor**.
9. From the **Optimization Process Manager**, click **Results** to view the combined results output database file in the Visualization module.

Additional information

- *Postprocessing an optimization*
- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

Viewing the results of your optimization process

The results from each design cycle are stored in a separate output database file. To view the results of the optimization, you must combine the separate output database files into a single output database file, as described in *Combining optimization results*.

To view the results of your optimization process, select **Optimization->Results->*optimization process name***. You can also view the results by selecting the optimization process name in the **Optimization Process Manager** and clicking **Results**. Abaqus/CAE enters the Visualization module and, if you executed a topology optimization, automatically displays a view cut superimposed on the undeformed view. The isosurface variable of the view cut is the normalized material property that the Optimization module uses to "add" or "remove" elements from the analysis. If you executed a shape optimization, Abaqus/CAE displays an undeformed view showing the new position of the surface nodes. If you executed a sizing optimization, Abaqus/CAE displays an undeformed view showing the shell thickness. You can view the results of an optimization while it is running. For more information, see *Monitoring your optimization process*.

Additional information

- *Postprocessing an optimization*
- *Understanding optimization processes*
- *Creating, editing, and manipulating optimization processes*

The Sketch module

Sketches are two-dimensional profiles that are used to help form the geometry defining an Abaqus/CAE native part.

You use the Sketch module to create a sketch that defines a planar part, a beam, or a partition or to create a sketch that might be extruded, swept, or revolved to form a three-dimensional part. This chapter explains how you use the tools within the Sketch module to create, modify, and manage sketches.

In this section:

- *Understanding the role of the Sketch module*
- *Entering and exiting the Sketch module*
- *Using the Sketch module*
- *Basic Sketcher concepts*
- *Sketcher geometry*
- *Specifying precise geometry*
- *Controlling sketch geometry*
- *Modifying, copying, and offsetting objects*
- *Customizing the Sketcher*
- *Sketching simple objects*
- *Creating construction geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Editing dimensions*
- *Adding reference geometry*
- *Projecting edges onto a sketch*
- *Moving or copying sketch geometry*
- *Modifying objects*
- *Repairing short edges, gaps, and overlaps*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*
- *Resetting the view*
- *Managing stand-alone sketches*

Understanding the role of the Sketch module

You use the Sketch module to create and manage two-dimensional profiles that are not associated with a feature; these profiles are known as stand-alone sketches. Stand-alone sketches can be incorporated in the current sketch, and they will overlay any existing geometry. For more information, see *Managing stand-alone sketches*.

Entering and exiting the Sketch module

You can enter the Sketch module at any time during an Abaqus/CAE session by clicking **Sketch** in the **Module** list located in the context bar. The Sketch module tools appear in the module toolbox, and the **Sketch** menu appears on the main menu bar.

To exit the Sketch module, first exit the current drawing tool by clicking mouse button 2. Then click the **Done** button that appears in the prompt area, and select any other module from the **Module** list. You need not take any specific action to save your sketches before exiting the module; they are saved automatically when you save the entire model by selecting **File->Save** or **File->Save As** from the main menu bar.

You should recognize the difference between the Sketch module and the Sketcher. When you create or edit a stand-alone sketch, the Sketch module starts the Sketcher and displays the Sketcher tools in the module toolbox. In addition, Abaqus/CAE starts the Sketcher whenever you do one of the following:

- Create or edit a feature while defining a part in the Part module.
- Sketch a partition on a face or cell while working in the Part module.
- Sketch a partition on a face or cell in the assembly while working in the Assembly module or the Mesh module.

To exit the Sketcher and incorporate your sketch into the part or assembly, first exit the current drawing tool by clicking mouse button 2. Then, click the **Done** button that appears in the prompt area. Where applicable, follow any additional prompts to extrude, revolve, or sweep the feature. Abaqus/CAE exits the Sketcher and returns to the module that invoked it. In addition, Abaqus/CAE restores the original view of the part or assembly.

Using the Sketch module

This section provides an overview of the Sketch module.

In this section:

- *Stand-alone sketches*
- *Imported sketches*

Stand-alone sketches

You use the Sketch module to create and manage sketches that are not associated with a feature; these sketches are called stand-alone sketches.

A stand-alone sketch is stored in the model; like other objects in the model, such as parts and loads, a stand-alone sketch can be edited, copied, renamed, and deleted. To create a stand-alone sketch, click the sketch create tool  in the Sketch module toolbox.

When you are using the Sketcher (for example, when you are sketching the profile of a solid extrusion in the Part module), you can save your work as a stand-alone sketch by clicking the sketch save tool  in the Sketcher toolbox.

Similarly, you can incorporate one stand-alone sketch in another sketch by clicking the sketch retrieve tool  in the Sketcher toolbox. When you retrieve a stand-alone sketch, Abaqus/CAE does the following:

- Positions the retrieved sketch so that its origin is coincident with the origin of the current sketch.
- Resizes the viewport so that both the current sketch and the retrieved sketch are displayed.
- Asks if you want to translate, rotate, mirror, or scale the retrieved sketch from the default size and position.

Additional information

• *Managing stand-alone sketches*
• *Using the Sketch module*

Imported sketches

You can import stand-alone sketches into the model by selecting **File->Import->Sketch** from the main menu bar. You can import sketches from files stored in the following industry-standard formats:

- AutoCAD (file extension `.dxf`)
- IGES (file extension `.igs`)
- ACIS (file extension `.sat`)
- STEP (file extension `.stp`)

When you import a file as a stand-alone sketch, Abaqus/CAE translates the contents of the file into a set of Sketcher entities, such as lines, arcs, and circle. Abaqus/CAE can import most planar entities from an IGES-format file. For a complete list of the IGES entities that can be imported into a sketch, see *IGES entities recognized by Abaqus/CAE when importing a part or a sketch*. When you import an AutoCAD- or ACIS-format file, Abaqus/CAE recognizes only the entities listed in *Table 1* and *Table 2*. If Abaqus/CAE finds geometry that it cannot translate, it ignores that geometry and continues importing the sketch.

Table 1: AutoCAD entities supported by the Sketcher.

AutoCAD entity	Sketcher entity
Line	Line
Circle	Circle
Arc	Arc
Vertex	Point

Table 2: ACIS entities supported by the Sketcher.

ACIS entity	Sketcher entity
Line	Line
Circle	Circle
Arc	Arc
Vertex	Point
Ellipse	Ellipse
Spline and polyline	Spline

For Abaqus/CAE to convert the file to a sketch, the file must contain a two-dimensional planar profile that can be mapped to the sketch plane. If the file contains three-dimensional geometry, Abaqus/CAE cannot import the sketch.

In some cases, when you import a sketch to create the base feature of a part, Abaqus/CAE positions the sketch relatively far from the Sketcher origin. As a result, the part will be far from its origin, and the analysis of your model can be affected by a loss of precision. To improve precision, you may have to move the part closer to its origin by moving the vertices of the sketch closer to the origin of the Sketcher grid. For more information, see *The Sketcher sheet and grid*.

Basic Sketcher concepts

You use the Sketcher to sketch the lines and curves that form the two-dimensional profile of a feature, to add constraints to the sketch, and to modify the sketch. This section describes some of the basic concepts used by the Sketcher and how these concepts influence the behavior of the Sketcher tools and appearance of the sketch.

In this section:

- *The Sketcher tools*
- *The Sketcher sheet and grid*
- *How Abaqus/CAE orients your sketch*
- *Realigning the sketch grid relative to the sketch*
- *The Sketcher cursors and preselection*
- *Using the chain method to select edges in the Sketcher*
- *How Sketcher customization options are initialized and saved*
- *Using the Query toolset in the Sketcher*

The Sketcher tools

You can access all the Sketcher tools through either the main menu bar or the toolbox. *Figure 1* shows the hidden icons for all the tools in the Sketcher toolbox.

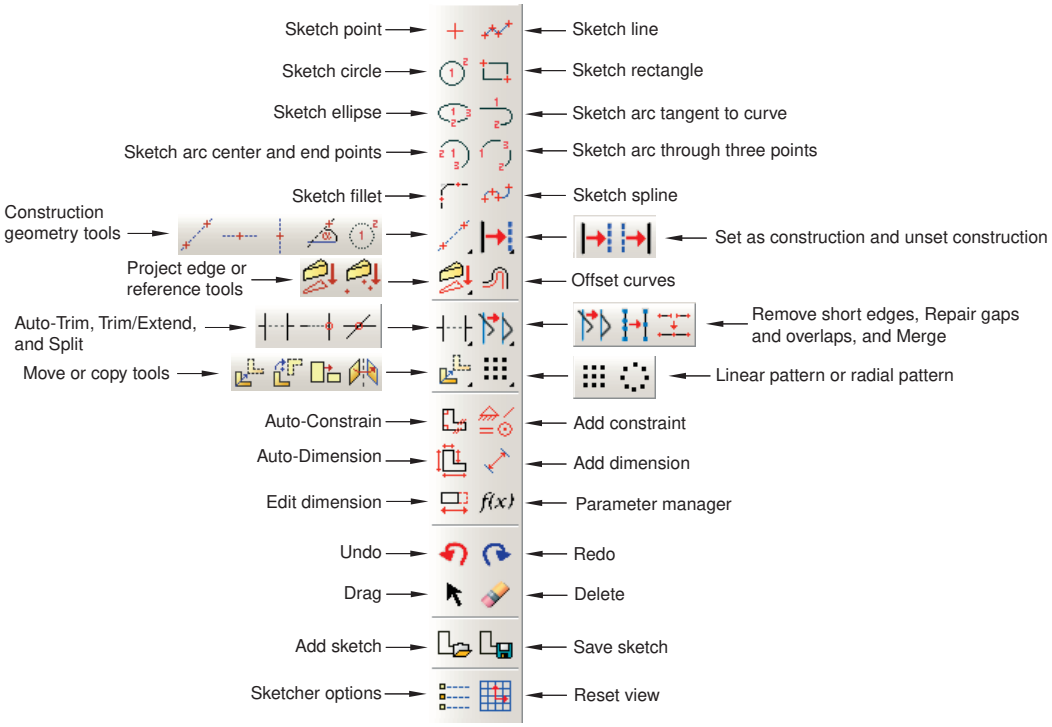


Figure 1: The Sketcher toolbox.

To see a tooltip containing a brief definition of a Sketcher tool, hold the mouse over the tool for a moment. For information on using toolboxes and selecting hidden icons, see *Using toolboxes and toolbars that contain hidden icons*.

The Sketcher tools allow you to do the following:

- Create basic sketch entities, such as lines, circles, arcs, ellipses, fillets, and splines.
- Add construction geometry to help you position and align sketch entities.
- Add constraints, dimensions, and parameters to control your sketch geometry and add precision.
- Translate, rotate, scale, or mirror sketch geometry.
- Drag, trim, extend, split, or merge sketch entities.
- Create similar objects by offsetting, creating linear patterns, or creating radial patterns.

Additional information

• *Basic Sketcher concepts*
• *Using the Sketch module*

The Sketcher sheet and grid

When you enter the Sketcher and either create a new sketch or edit an existing sketch, Abaqus/CAE displays a sheet in the current viewport on which you sketch. In addition, the triad in the lower-left corner of the viewport indicates the orientation of the part or the assembly relative to the Sketcher sheet. The sheet is always square, and its height and width are determined by the sheet size. Abaqus/CAE determines the sheet size and the location of the origin, depending on what you are sketching:

- If you are sketching the base feature of a new part, the sheet size is the same as the approximate size of the part that you provided when you created the part. The origin of the sheet is located at the origin of the part's coordinate system. Similarly, if you are creating a stand-alone sketch, the sheet size is the same as the approximate size of the sketch that you provided when you created the sketch.
- If you are adding a feature to a part or to the assembly, the default sheet size depends on the size of the face on which you are sketching. The origin of the sheet is located at the centroid of the selected face. If you selected a datum plane as the sketching plane, the origin of the sheet is located at the center of the part or assembly; the sheet size depends on the total size of the part or assembly.

You can use the Sketcher customization options to increase or decrease the sheet size if it does not correspond with the size of the geometry you are trying to sketch. To access the Sketcher customization options, select the customization tool  from the Sketcher toolbox. You may need to use the magnify tool  to view the entire Sketcher sheet within the viewport.

Abaqus/CAE overlays the sheet with a grid of invisible grid points to help you position the cursor as you draw, move, resize, or reshape objects. By default, when you move the cursor near a grid point, the cursor automatically moves, or snaps, to the point. This behavior allows you to easily position the cursor precisely on a grid point while also providing you with full control when the cursor is not close to any grid points. If it is more convenient, you can *disable* the snapping behavior so that you have full control over the cursor. To help you visualize the grid points underlying the Sketcher grid, Abaqus/CAE displays visible gridlines that pass through the grid points at a selected interval; for example, every other grid point.

For example, *Figure 1* shows a magnified view of a sheet whose spacing is set to 2 units.

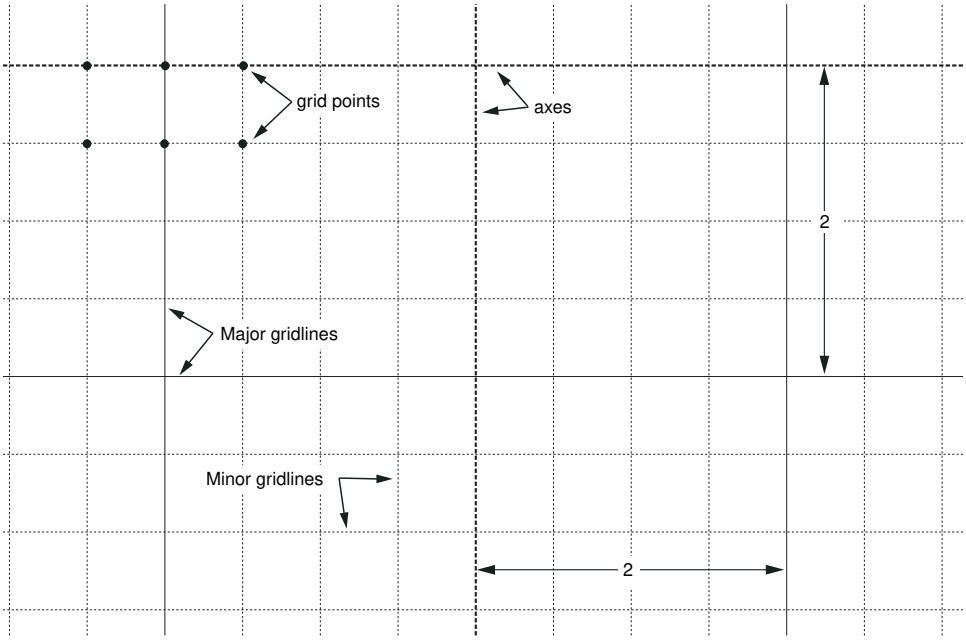


Figure 1: Magnified view of the Sketcher grid.

In this example the Sketcher displays both major gridlines (in a solid line style) and minor gridlines (in a thin, dashed line style). Minor gridlines appear in the Sketcher only when the magnification level is high. In addition, the thick dashed lines in the sheet indicate the *X*- and *Y*-axes of the sketch; the lines intersect at the origin of the sketch.

You can *customize* the appearance and behavior of the grid by choosing the spacing of grid points, the spacing of the visible gridlines that overlay the grid points, the number of minor intervals, and the sheet size. You can also realign the grid relative to the sketch by moving the origin of the grid and by rotating the grid. Your sketch can extend beyond the Sketcher grid; however, if you find that you need to sketch outside the grid, it is recommended that you increase its size to include the entire sketch. If you change the grid origin or rotation, the Sketcher displays cursor coordinates in two forms as you continue to modify the sketch. Grid coordinates indicate the cursor position using the new grid origin or rotation; sketch coordinates indicate the position based on the original grid location.

The analysis of your model can be affected by a loss of precision if the coordinates of a part are far from the origin of the part. For example, after you import a sketch to create the base feature of a part, the sketch may be relatively far from the Sketcher origin; as a result, the part will be far from its origin. To improve precision, you may have to move the part closer to its origin by moving the sketch geometry closer to the origin of the Sketcher grid using the translate tool, . You should not move the origin of the sketch closer to the geometry. Moving the origin of the Sketcher grid closer to the sketch is a graphical convenience only and has no effect on the underlying precision of the coordinates.

Additional information

- *Turning snapping on or off*
- *Customizing the sheet size and grid*
- *Basic Sketcher concepts*

How Abaqus/CAE orients your sketch

When the Sketcher starts, Abaqus/CAE orients the view of the part or assembly so that the face or datum plane on which you are sketching is parallel to the screen; this plane is known as the sketch plane. The orientation of the sketch plane depends on the modeling space of the part or assembly and the type of feature you are creating:

Two-dimensional or axisymmetric modeling space

When you add a planar feature to a two-dimensional or axisymmetric part or assembly, Abaqus/CAE starts the Sketcher and aligns the *X*- and *Y*-axes of the part with the axes of the Sketcher grid, regardless of the type of feature you are creating. (Abaqus/CAE derives the *X*- and *Y*-axes of the part from the axes of the sketch that defined the base feature.)

Three-dimensional modeling space

When you add a feature to a three-dimensional part or assembly, you must use the following technique to control the orientation of the view relative to the Sketcher grid:

1. Select the plane on which to sketch by selecting appropriate geometry; for example, a face of a part or a datum plane.
2. Select an edge. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. Alternatively, you can choose a different orientation for the edge in the Sketcher grid before you select the edge; for example, horizontal and on the top of the Sketcher grid. You can select any edge from the part or assembly that is not perpendicular to the selected plane. You can select a datum axis, or you can select one of the axes of a datum coordinate system; but you cannot select the edge of a datum plane. You can select a curved edge, but the resulting orientation of the part or assembly is system dependent.

Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected edge aligns with the grid in the desired orientation. The orientation of the view on the Sketcher grid also depends on the type of feature you are creating:

- When you are sketching a cut feature in the Part module, Abaqus/CAE orients the view so that the resulting feature will cut away from you and into the screen.
- When you are sketching a feature other than a cut (for example, an extruded solid), Abaqus/CAE orients the view so that the resulting feature will protrude out of the screen toward you.
- When you are adding a swept feature, you must sketch a sweep path and sweep profile. For details of the resulting view orientation, see *Adding a swept solid feature*; *Adding a swept shell feature*; and *Creating a swept cut*.

When you are partitioning faces using the sketch method in the Partition toolset, Abaqus/CAE orients the view based upon the modeling space of the part or assembly and the type of faces you selected to partition. For more information, see *Using the sketch method to partition faces*.

If you are unsure of the part's or assembly's orientation relative to the sketch plane, use the view manipulation tools to examine the sketch plane and the object on which you are sketching. Use the reset view tool  to return to the original view.

Figure 1 illustrates how Abaqus/CAE determines the sketch plane orientation relative to a three-dimensional part after you select a face, an edge, and the orientation of the edge on the Sketcher grid.

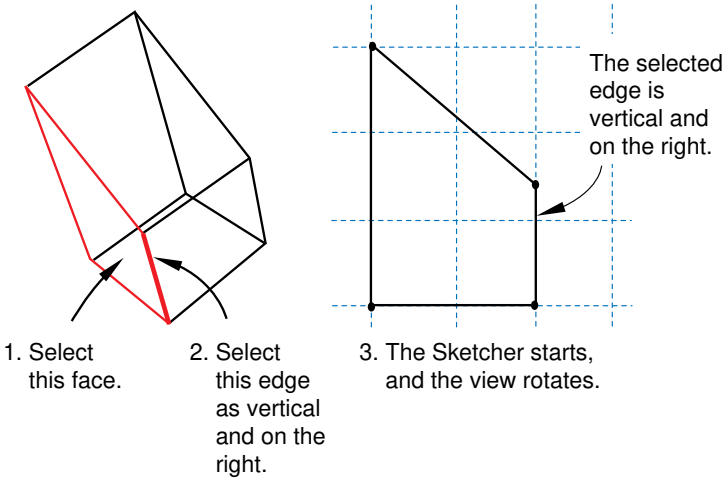


Figure 1: Determining the sketch plane orientation.

Additional information

- *Adding a swept solid feature*
- *Adding a swept shell feature*
- *Creating a swept cut*
- *Basic Sketcher concepts*
- *Using the Sketch module*
- *Using the sketch method to partition faces*

Realigning the sketch grid relative to the sketch

When you are sketching a feature, the sketch grid does not always align with the vertices and lines of the sketch or the underlying reference geometry. You can use the following techniques to realign the grid:

- Shift the grid relative to the sketch by selecting a vertex representing the origin of the grid.
- Rotate the grid relative to the sketch by selecting a line that will be parallel to the *X*-axis of the sketch.
- Remove any realignment and reset the grid to its original orientation.

Realignment applies only to the particular sketch, and the alignment of the grid is stored along with the sketch in the model database. New and existing sketches retain their default alignment.

Changing the origin and rotating the sketch grid allows you to quickly create features that might be difficult to create using the default grid location. Rotating the grid has the following effect:

- The grid coordinates of the current cursor position are displayed in the upper-left corner of the viewport, and you can use these coordinates as a guide. The grid coordinates displayed are relative to the current alignment of the sketch grid and change if you realign the grid.
- The sketch coordinates of the current cursor position are displayed in the upper-right corner of the viewport. The sketch coordinates displayed are relative to the original alignment of the sketch grid and do not change if you realign the grid.
- When you enter the coordinates of a vertex in the prompt area, those coordinates are relative to the current alignment of the sketch grid.
- Old and new horizontal and vertical dimensions remain horizontal and vertical.
- Old horizontal and vertical construction lines remain horizontal and vertical.
- New horizontal and vertical construction lines and preselection points are aligned with the rotated grid.
- The lines generated by the rectangle tool are aligned with the rotated grid.

Additional information

• *Realigning the sketch grid*
• *Basic Sketcher concepts*
• *Using the Sketch module*

The Sketcher cursors and preselection

There are two cursors active within the Sketcher: the primary cursor and the secondary cursor. The two cursors are shown in *Figure 1*.

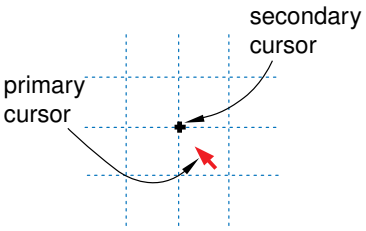


Figure 1: The Sketcher cursors.

The primary cursor is the one you use with most applications on your computer, including Abaqus/CAE. The primary cursor usually appears as an arrow pointer or a pointing hand; you position this cursor by moving the mouse. The secondary cursor is active only in the Sketcher; it appears near the primary cursor when the Sketcher prompts you to select a point. The position of the secondary cursor allows you to see exactly which point is selected before committing the selection. If you move the primary cursor near a point that is eligible for selection, the secondary cursor jumps directly to the point while the primary cursor remains fixed. This behavior is called preselection. If you click the mouse button, Abaqus/CAE selects the point under the secondary cursor.

The appearance of the secondary cursor changes as you move around the sketch. *Table 1* shows the shape assumed by the secondary cursor when the primary cursor is close to the Sketcher entity listed.

Table 1: Preselection cursors.

Preselection cursor	Sketcher entity
□	A vertex
	The midpoint of a line or curve
×	The intersection of lines and curves
×——	The projection of a line onto other lines and curves
	Points along existing lines or curves
+	The intersection of gridlines with each other or with other lines and curves

The secondary cursor works in conjunction with other preselection symbols in the sketch to indicate special points that you can select. *Table 2* shows the other preselection symbols used in the Sketcher and the meaning of each symbol.

Table 2: Preselection symbols.

Preselection symbol	Location and Meaning
□	At the starting point of a new line segment when the line starts on existing sketch, reference, or construction geometry. The new line segment is perpendicular to the existing geometry.
‖	Near the starting point when creating a new line starting on an existing edge; the symbol is aligned with the existing edge. The new line segment is tangent to the existing edge.
H	Near the secondary cursor when creating a new line. The new line segment is horizontal.
V	Near the secondary cursor when creating a new line. The new line segment is vertical.

The preselection cursors and symbols also indicate constraints that Abaqus/CAE will apply to the sketch if you choose the preselected point. Constraints such as coincidence, perpendicularity, and tangency help control sketch geometry and preserve your design intent.

Preselection applies to any entity that is a valid selection in the sketch. For example, as you move the cursor around the sketch, preselection highlights the following to indicate a valid selection:

- Vertices, intersections, projections, and points of tangency highlight when you are sketching a line, as shown in *Table 1*.
- Sketch, construction, and reference geometry highlight when you are adding a dimension.
- Dimensions highlight when you are modifying a dimension.

You can customize the cursor's behavior as follows:

- By default, the secondary cursor snaps to grid points that are close to the primary cursor. If you turn off this snapping, the secondary cursor follows the primary cursor and can be positioned anywhere on the Sketcher sheet.
- You can turn off preselection.

For more information on customizing the secondary cursor, see *Turning snapping on or off*, and *Turning preselection on or off*. The secondary cursor is available only in the Sketcher, but you can use another form of preselection to help you select viewport objects in other Abaqus/CAE modules. For more information on selecting viewport objects, see *Selecting and unselecting individual objects*.

Additional information

- *Controlling sketch geometry*
- *Customizing the Sketcher*
- *Using the Sketch module*
- *Basic Sketcher concepts*

Using the chain method to select edges in the Sketcher

When offsetting or copying edges in a sketch, you can select a "chain" of connected edges. A chain is a connected group of edges in which each edge may be connected to, at most, one other edge at each endpoint—like the links in a chain. The chain method is available only in the Sketcher, and it allows you to select a chain of edges as if they were a single entity.

When you are performing a task that allows you to pick more than one edge from a sketch, Abaqus/CAE displays a field in the prompt area. The field allows you to choose between the two selection methods—**individually** and **by chain**, as shown in *Figure 1*.

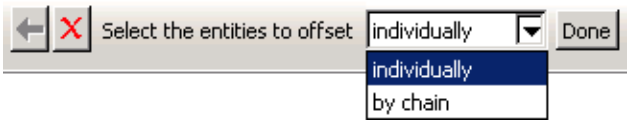


Figure 1: Choose the selection method from the field in the prompt area.

After you use the chain method, you can click the **individually** method in the prompt area and **[Shift] + Click** on individual edges to append them to your selection. You can also **[Ctrl] + Click** on edges to unselect them. In addition, you can continue to use the chain method and use **[Shift] + Click** to append more chains to your selection.

Additional information

- *Basic Sketcher concepts*
- *Using the Sketch module*
- *Selecting objects within the current viewport*

How Sketcher customization options are initialized and saved

Sketcher customization options are grouped into the following settings:

Sketcher options

- Snapping
- Preselection behavior
- Whether to display construction geometry, dimensions, parameter names, and constraints
- The types of constraints and dimensions that Abaqus/CAE can add automatically

Sketch options

- Sheet size, grid spacing, and grid display
- The origin of the grid and the alignment of the axes
- The text height and decimal places of dimensions and text height of parameter names

Sketcher options control the interactive behavior of the Sketcher, while Sketch options control the appearance of individual sketches. Abaqus/CAE stores and uses your Sketcher customization settings only for the duration of the session. In contrast, Abaqus/CAE stores your Sketch customization settings along with each sketch in the model database. As a result, if you exit the Abaqus/CAE session and return to the sketch at a later date, the Sketch customization options are retained.

When you create a new part or stand-alone sketch, Abaqus/CAE always uses the default Sketch options. The sheet size and grid spacing are recalculated based on the approximate size of the part or the selected sketch plane, the origin of the sketch is at `0,0` , the grid is displayed and aligns with the 1- and 2-axes, and the text height and decimal places are at their default settings.

Additional information

• *Using the Sketch module*
• *Customizing the Sketcher*

Using the Query toolset in the Sketcher

You can use the Query toolset to request either general information or module-specific information.

For a discussion of the information displayed by general queries, see *Obtaining general information about the model*.

Select **Tools->Query** from the main menu bar, or click the  tool in the Query toolset to start the Query toolset.

The following queries are specific to the Sketcher:

Constraint

Abaqus/CAE displays the constraint type and the names of the constrained entities in the message area and highlights the entities in the sketch.

Detail

Abaqus/CAE displays the number of geometries, vertices, constraints, dimensions, and unconstrained degrees of freedom in the message area.

Additional information

- *Basic Sketcher concepts*
- *The Query toolset*

Sketcher geometry

This section describes the different types of geometry available in the Sketcher.

In this section:

- *Reference geometry*
- *Construction geometry*

Reference geometry

When you sketch the profile of a feature, Abaqus/CAE projects onto the sketch sheet all the edges and vertices of the part or assembly that are in the same plane as the sketch plane. Abaqus/CAE also projects all datum points and datum axes onto the sketch sheet, regardless of their location in the part or assembly. These projected points and lines are called reference geometry, and you can use them for reference while sketching.

You typically use reference geometry to position your sketch accurately by specifying a dimension between a sketched entity, such as a line or a vertex, and the reference geometry. You can also use reference geometry to realign the sketch grid. The sketch retains any reference geometry that is used to position or dimension the sketch—the reference geometry reappears when you use the Sketcher to edit the feature again. Abaqus/CAE discards any reference geometry that is not used in the sketch.

If your part is complex, projecting only the edges, vertices, and datum points in the sketch plane onto the sketch sheet reduces the amount of clutter on the screen and makes the selection procedure more straightforward. However, your selection will be limited to only edges, vertices, and datum points that are in the sketch plane. If you want to use other edges, vertices, or datum points–including orphan element edges and nodes—as references, you must project them onto the sketch plane as reference geometry by using the **Project References** tool . Alternatively, if you want to create new sketch edges from existing model edges or orphan element edges, you can use the **Project Edges** tool  (for more information, see *Projecting edges onto a sketch*).

When you sketch a new feature, you can constrain the position of the sketch to the underlying reference geometry using one of the following methods:

- Select entities from the reference geometry while sketching the new feature; for example, select a projected vertex to define the center of a circle or one end of a line.
- Create a constraint, dimension, or parameter between the sketch and any projected reference geometry. For more information, see *Controlling sketch geometry*.

Constraining the sketch defines how it is positioned relative to the reference geometry and how it will be repositioned if you modify and regenerate the part or assembly.

When you exit the Sketcher, Abaqus/CAE determines whether you have constrained the sketch to the reference geometry. If a constraint is found, Abaqus/CAE creates a parent-child relationship between the new feature (the child) and the selected reference geometry (the parent). If no constraints are found, Abaqus/CAE displays the following message in the message area: `No sketch placement constraints specified on feature feature name`. You do not have to constrain the sketch; but if you do not, its position relative to the underlying geometry of the part or assembly may change if you modify and regenerate the part or assembly.

If you save a stand-alone sketch, Abaqus/CAE saves only the sketch and does not save any underlying reference geometry. As a result, if the sketch includes a dimension between a sketch line and a reference line, the dimension is not saved with the stand-alone version of the sketch.

The order in which you create features influences the available reference geometry. Feature-based modeling allows a child feature to be dependent on a parent feature but does not allow a parent to be dependent on a child. As a result, if a feature was created after the feature you are editing, the edges and vertices of the newer feature cannot be projected onto the sketch sheet. You can use the Query toolset to determine the order in which features were created; for more information, see *Using the Model Tree to obtain feature information*.

Additional information

- *The Feature Manipulation toolset*
- *Using the Sketch module*

Construction geometry

You create construction geometry in the Sketcher to help you position and align the geometry in your sketch; objects that you might need to position include holes, arcs, slots, or gear teeth.

The Sketcher allows you to add construction lines and circles to your sketch; in addition, points that you create using the isolated point tool  are considered construction geometry. Construction lines, circles, and points do not appear in the feature you are creating or modifying.

While you are adding construction geometry and moving the cursor around the sketch, Abaqus/CAE displays preselection symbols at the following locations:

- The intersection of the new construction geometry and sketched lines or curves.
- The intersection of the new construction geometry and existing construction geometry.
- The intersection of the new construction geometry and reference geometry. Abaqus/CAE displays preselection symbols only when the reference geometry was created before the feature being edited. (See *Reference geometry*, for more information about the behavior of reference geometry.)

The preselection symbols generated by construction geometry allow you to align objects precisely; for example, along an oblique line or around a circle. For example, you could create an oblique construction line and several vertical construction lines to help align a group of circles, as illustrated in *Figure 1*.

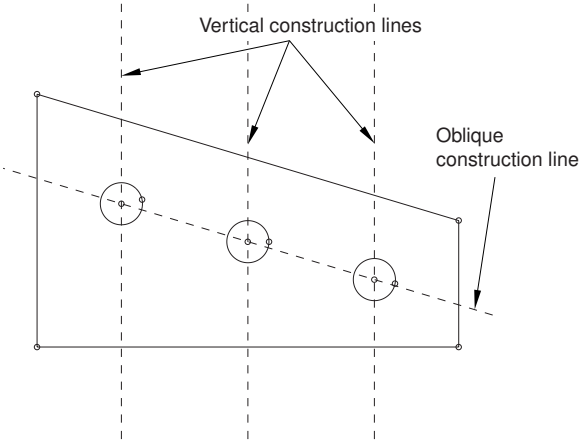


Figure 1: Using construction lines to align sketch geometry.

Construction lines also define the axis of rotation when sketching revolved solids and surfaces. For more information about the relationship between construction lines and the axis of revolution, see *Defining the axis of revolution for axisymmetric parts and for revolved features*.

Construction geometry is shown using dashed lines to distinguish it from sketched geometry. Construction geometry is visible only while you are working on a sketch; as soon as you exit the Sketcher, the construction geometry disappears. Abaqus/CAE saves construction geometry with the original sketch; if the Sketcher is invoked to modify a sketch that included construction geometry, the construction geometry reappears along with the sketch.

To create construction geometry, select one of the construction geometry tools from the Sketcher toolbox or select **Add->Construction** from the main menu bar.

You can also convert components of your sketch into construction geometry, and you can reverse this process for items that had been converted to construction geometry. To convert a component to construction geometry, select the

component and select **Edit->Set as Construction** from the main menu bar; to reverse this process, select the construction geometry component and select **Edit->Unset Construction**.

Additional information

- *Creating construction geometry*
- *Reference geometry*
- *Controlling sketch geometry*
- *Using the Sketch module*

Specifying precise geometry

When you are using the Sketcher to sketch features, you can use the underlying grid and preselection to position the cursor and create or modify sketch geometry. In addition, you can use the following techniques to position the geometry precisely:

Entering coordinates

When you are using the Sketcher to create a feature and Abaqus/CAE asks you to position a point (for example, to define the endpoint of a line or the center of a circle), you can do one of the following:

- Select a point on the sketch using the Sketcher grid, existing vertices, or preselection.
- Enter the desired *X*- and *Y*-coordinates of the point in a text box in the prompt area.

  To help you determine the coordinates to provide, the *X*- and *Y*-axes of the sketch are indicated by dashed lines, which intersect at the origin of the sketch. The coordinates that you specify are relative to this origin. In addition, when you select a Sketcher tool, the coordinates of the current cursor position are displayed in the upper-left corner of the viewport, and you can use these coordinates as a guide.

For detailed instructions, see *Sketching simple objects*.

Creating and modifying dimensions

After you have created the geometry, the Sketcher allows you to add dimensions between lines and vertices and other pieces of geometry in the sketch. You can then refine the sketch by replacing these dimensions with precise dimensions, and the sketch changes to reflect the new dimensions. In addition, dimensions allow you to annotate sketches for future reference.

You add dimensions by selecting the dimension tool  in the Sketcher toolbox or by selecting **Add->Dimension** from the main menu bar—editing the dimension value is the final step in creating a new dimension. To modify an existing dimension, select the modify dimension tool  from the Sketcher toolbox or select **Edit->Dimension** from the main menu bar. For detailed instructions, see *Editing dimensions*.

Additional information

- *Automatically constraining a sketch*
- *Adding individual dimensions*
- *Modifying objects by changing dimensions or adding parameters*
- *Using the Sketch module*

Controlling sketch geometry

Controlling sketch geometry helps you prevent unintended changes by creating relationships between sketch entities. If all the geometry in a sketch is controlled, the sketch is fully constrained—the entities in the sketch cannot be modified without violating the controls, and you cannot add another control without creating a conflict with one that already exists.

The following section describes how you can use constraints, dimensions, and parameters to control sketch geometry. All three types of controls can be considered constraints since they all work to restrict the degrees of freedom in a sketch, but each control creates a different type of relationship between the selected entities.

In this section:

- *Using constraints to control sketch geometry*
- *Using dimensions to control sketch geometry*
- *Using parameters to control sketch geometry*
- *Fully constrained geometry*

Using constraints to control sketch geometry

Constraints create logical relationships that control the position or size of geometry. Constraints are defined without using numerical values. The other types of geometry controls, dimensions and parameters, are specialized constraints that use values or evaluated expressions to define numerical relationships in a sketch. Constraints are not required in Abaqus/CAE sketches; however, they can help you preserve the design intent. When you modify a constrained sketch, you must either work within the existing controls, modify them, or delete them to complete your changes.

Table 1 lists each constraint type, its symbol, and a brief description of its use.

Table 1: Sketcher constraints.

Constraints	Symbol	Description
Coincident	o	Abaqus/CAE moves the selected entities so that they share a common point. The point may be a point on both entities or a projected point that would exist only if one of the entities was extended. If two straight lines are selected, they become colinear.
Concentric	⊙	Abaqus/CAE moves the selected arcs or circles to share a common center.
Equal length	/	Abaqus/CAE makes the selected lines all the same length.
Equal radius	—	Abaqus/CAE makes the selected arcs or circles the same radius.
Fixed	△	Abaqus/CAE fixes the radius of an arc or circle, the angle of a line, or the position of a vertex.
Horizontal	H	Abaqus/CAE aligns the selected line with the *X*-axis of the sketch. The horizontal constraint is not affected by grid orientation.
Equal distance	|o|	Abaqus/CAE moves the selected vertex so that it is the same distance from the two vertices or lines used to define the constraint.
Parallel	//	Abaqus/CAE makes the selected lines parallel.
Perpendicular	□	Abaqus/CAE makes the selected lines perpendicular.
Symmetry	o|o	Abaqus/CAE makes the selected entities symmetrical about your chosen line of symmetry.
Tangent	‖	Abaqus/CAE makes the selected entities tangent at their nearest point. At least one of the entities must be curved.
Vertical	V	Abaqus/CAE aligns the selected line with the *Y*-axis of the sketch. The horizontal constraint is not affected by grid orientation.



Note:

The behaviors described in *Table 1* assume that no other constraints are applied to the selected entities. The presence of other constraints and the constraint solution method selected in the **Sketcher Options** may affect how Abaqus/CAE moves entities to comply with the constraints.

You can create multiple instances of the same type of constraint in a sketch. You can use the sketch module queries to locate the geometries controlled by a constraint or to list the constraint details for the entire sketch (for more information, see *The Query toolset*).

You can use the following techniques to add constraints to a sketch:

Preselecting geometry

If preselection is active, Abaqus/CAE can add some constraints while you create geometry. Coincident, horizontal, perpendicular, tangent, and vertical constraints can be added using the preselection cursor. For more information, see *The Sketcher cursors and preselection*.

Selecting midpoints of lines or curves

If you select the midpoint of a line or curve as the vertex for additional sketch geometry, Abaqus/CAE also creates an equal distance constraint between the midpoint and the endpoints of the line or curve.

Adding constraints automatically

After you create a sketch, you can use the auto-constrain tool  to add constraints to the entire sketch or to a selected group of entities. Use the sketch options to control the constraint types that Abaqus/CAE can add automatically and the allowable tolerances for adding a constraint. For more information, see *Customizing the use of constraints in the Sketcher*.

Adding individual constraints

After you create sketch geometry, you can select the constraint type and pick the entities to apply the constraint. This method gives you the most control, but it is also the most time consuming. For more information, see *Adding individual constraints*.

When you add constraints, you must not overconstrain the sketch geometry. Geometry is overconstrained when multiple controls are applied to a single degree of freedom (for more information, see *Fully constrained geometry*). Abaqus/CAE colors conflicting constraints magenta to indicate an overconstraint condition. You must resolve any overconstraints before you can use a sketch to create a feature or save it with the model database.

Additional information

- *The Sketcher cursors and preselection*
- *Customizing the Sketcher*
- *Constraining, dimensioning, and parameterizing a sketch*

Using dimensions to control sketch geometry

Dimensions are a type of constraints that use numerical values to define sizes, angles, or distances in a sketch. Dimensions control sketch geometry by preventing changes to the dimensioned quantities. To change a dimensioned quantity, you must modify the associated dimension.

The following dimension types are available in the Sketcher:

- horizontal
- vertical
- oblique
- angular
- radial

The first three dimension types correspond to linear distances between two points (including the endpoints of a single line), two lines, or a point and a line. Angular dimensions always indicate the angle, in degrees, between two lines. Radial dimensions indicate the radius of a circle, arc, or fillet or the major and minor radii of an ellipse.

You can use the following techniques to add dimensions to a sketch:

Adding dimensions automatically

After you create a sketch, you can use the auto-dimension tool  to add dimensions to the entire sketch or to a selected group of entities. Use the sketch options to control the dimension types that Abaqus/CAE can add automatically. For more information, see *Customizing the format and use of dimensions in the Sketcher*.

Adding individual dimensions

After you create sketch geometry, you can add individual dimensions. This method gives you the most control, but it is also the most time consuming. For more information, see *Adding individual dimensions*.

You can modify any dimension in a sketch. When you edit an existing dimension, you can modify it in several ways. You can

- edit the numerical value to modify the sketch,
- convert the dimension to a reference dimension, or
- link the dimension to a parameter.

Reference dimensions annotate quantities that are already controlled elsewhere in the sketch or quantities such as projected reference geometry that cannot be controlled within the current sketch. Reference dimension values are enclosed in parentheses, and Abaqus/CAE updates them automatically to match changes in the associated geometry. Dimensions that have been linked to parameters are assigned a variable name for use in creating mathematical expressions (parametric equations) to define other parameters in the sketch. For detailed instructions on editing dimensions, see *Editing dimensions*.

When you add or modify dimensions, you must not overconstrain the sketch geometry. Geometry is overconstrained when multiple controls are applied to a single degree of freedom (for more information, see *Fully constrained geometry*). Abaqus/CAE colors conflicting constraints magenta to indicate an overconstraint condition. You must resolve any overconstraints before you can use a sketch to create a feature or save it with the model database.

Additional information

• *Modifying objects by changing dimensions or adding parameters*

- *Customizing the Sketcher*
- *Constraining, dimensioning, and parameterizing a sketch*

Using parameters to control sketch geometry

Parameters are dimensions or numerical constants that have been given a variable name and are used to create parametric equations between pieces of geometry. While dimensions always have a numerical value, parameters may have a numerical value or a value resulting from a mathematical expression. For example, you can parameterize the radius dimension of a circle so it will always be half the size of another dimension in the sketch, but you must first associate both dimensions with parameters to create the expression. You can use the sketch options to control whether Abaqus/CAE displays the names or values of parameters in a sketch (for more information, see *Customizing the format and use of dimensions in the Sketcher*).

Parameters must be defined within a single sketch—you cannot define parameters in one sketch and use them to create expressions in another sketch. Defining a parameter adds it to the **Parameter Manager** for the sketch. The order of parameters in the manager indicates the order in which they can be used in expressions. *Figure 1* shows two parameters; the *length* parameter is used to define the *width* parameter. If you instead wanted to define the length in terms of the width, the width parameter would need to appear first in the manager. You can use the buttons on the right side of the dialog box to create parameters by selecting dimensions from the viewport or to insert, move, or delete parameters from the manager. For detailed instructions, see *Adding and editing parameters*.

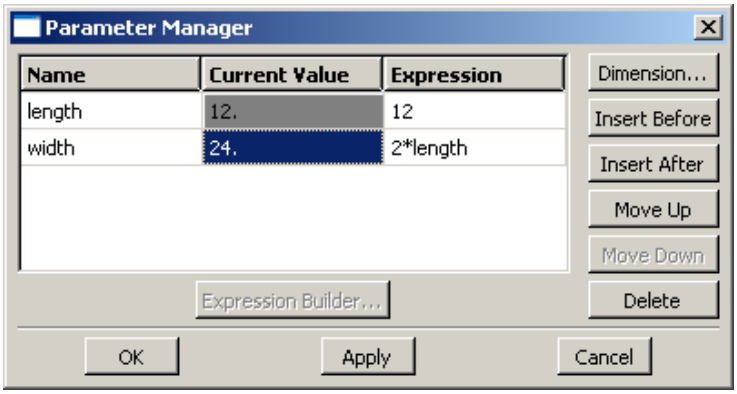


Figure 1: The Parameter Manager.

Additional information

- *Modifying objects by changing dimensions or adding parameters*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Creating parametric equations*
- *Modifying objects*

Fully constrained geometry

Constraints, dimensions, and parameters are all used to create relationships that control, or constrain, Sketcher geometry. Geometry is constrained by the removal of degrees of freedom from the sketch. For example, if you apply a perpendicular constraint to two lines, you remove the angular degree of freedom between those lines. If you drag one of the lines, the other line must also move to maintain the perpendicular constraint.

If geometry is controlled such that you cannot change it without redefining the relationships—changing constraints, dimensions, or parameters—the geometry is fully constrained. Abaqus/CAE colors fully constrained geometry green to indicate that it cannot be changed. If you add too many constraints to a sketch, the geometry becomes overconstrained. Abaqus/CAE colors conflicting constraints magenta to indicate the overconstraint condition. To resolve an overconstraint, you can

- undo the last modification or constraint addition,
- delete constraints or dimensions, or
- convert unnecessary dimensions to reference dimensions.

If you intend to fully constrain an entire sketch, you can use the **Detail** query to list sketch information—including the number of unconstrained degrees of freedom—in the message area. For more information, see *Using the Query toolset in the Sketcher*.

Additional information

- *Constraining, dimensioning, and parameterizing a sketch*
- *Editing dimensions*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Modifying, copying, and offsetting objects

The following section describes the techniques you can use to modify your sketch and to copy or offset Sketcher objects.

In this section:

- *Modifying a sketch by dragging objects*
- *Modifying objects by changing dimensions or adding parameters*
- *Modifying or copying objects by selecting edges*
- *Modifying edges by trimming, extending, splitting, or merging*
- *Copying sketch objects to create patterns*
- *Offsetting objects*

Modifying a sketch by dragging objects

You can drag vertices, edges, construction geometry, and dimension text to new locations in a sketch. You can drag only one object at a time, and the object you drag will snap to gridlines when the **Snap to grid** option in the **Sketcher Options** dialog box is toggled on. This option is toggled on by default.

When you drag a vertex or an edge, Abaqus/CAE repositions the selection and uses the **During drag** constraint solution method to modify any connected edges (for more information, see *Customizing the use of constraints in the Sketcher*). Sketch constraints may cause multiple objects to move along with the one you select. Constraints may also restrict or prevent dragging some objects. For example, consider the sketch shown in *Figure 1*.

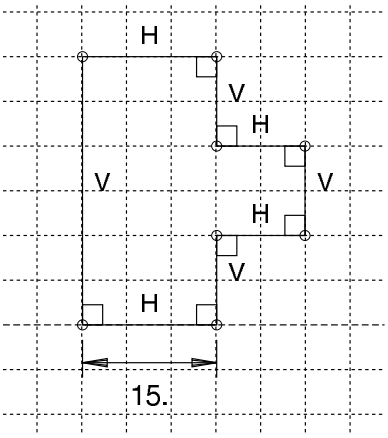


Figure 1: The original sketch.

The sketch includes the default horizontal, vertical, and perpendicular constraints created by Abaqus/CAE during the sketching process and one added length dimension. With these constraints in place, dragging the left edge of the sketch has the same effect as dragging the vertex at either end of the left edge, as shown in *Figure 2*.

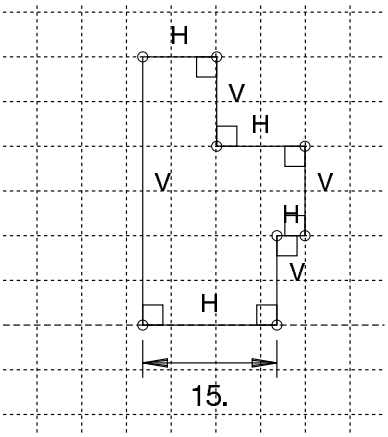


Figure 2: Dragging the left edge or vertices with constraints.

With the constraints and dimension removed, dragging the edge produces similar results to those shown in *Figure 2*, but dragging a vertex modifies the sketch as shown in *Figure 3*. The result in *Figure 3* is achieved using the default **Minimum move** constraint solution method. For this sketch, dragging the vertex using any of the other constraint solution methods repositions the entire sketch.

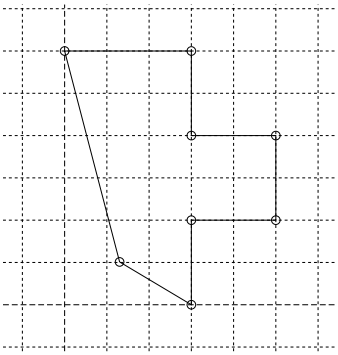


Figure 3: Dragging the lower left vertex without constraints.

Dragging edges and vertices is a quick, but imprecise, method of moving them since the motion is based on the cursor position as opposed to an exact numerical change. To move objects more precisely, use the methods described in *Modifying objects by changing dimensions or adding parameters*, and *Modifying or copying objects by selecting edges*.

Dragging dimension text is one way to clean up the sketch after using the **Auto-Dimension** tool. The lack of precision does not matter since the exact location of dimension text is not important. When you drag dimension text, you can make the following changes:

- Move a linear dimension closer to, farther from, or to the opposite side of the dimensioned quantity.
- Move an angular dimension closer to or farther from the angle's vertex.
- Move an angular dimension across one of the edges that define the angle to dimension a supplement of the original angle.

Possible locations for dragging an angular dimension are shown in *Figure 4*.

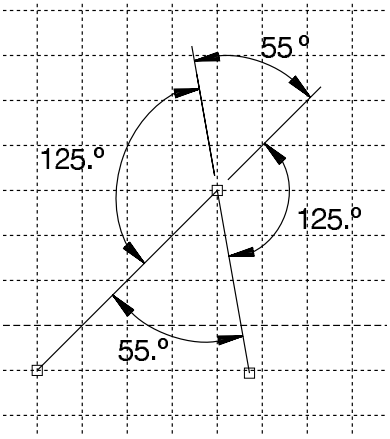


Figure 4: Four possible positions for an angular dimension.

Dimension text is not controlled by any constraints other than the attachment to the dimensioned object. The constraint solution methods also do not affect dragging dimension text in a sketch.

Additional information

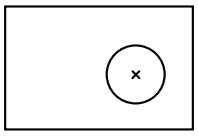
- *Specifying precise geometry*
- *Modifying objects*

Modifying objects by changing dimensions or adding parameters

Use dimensions between lines and vertices to define and maintain dimensional relationships between features. Dimensions can be applied between lines and vertices from any of the following:

- Sketch geometry
- Reference geometry
- Construction geometry

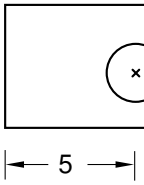
Parameters are dimensions or constants that have been associated with a name so you can use them to create functions and to define relationships between features. For example, consider a three-dimensional shell with an extruded circular post, as shown in the following figure:



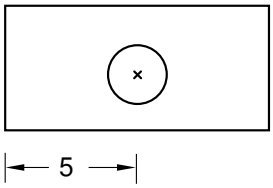
This part was created in two steps:

1. Create a part consisting of only the shell.
2. Extrude a solid feature from the face of the shell (the circular post).

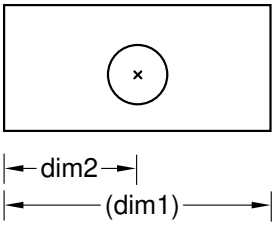
If you want to keep the distance between the left edge of the shell and the center of the post constant, you can edit the post and add a dimension between the center of the sketched circle and the left edge of the reference geometry that represents the shell, as shown in the following figure:



If you move either the left or the right edge of the shell, the distance between the center of the circle and the left edge remains constant, as shown in the following figure:



If you want to keep the post centered between the left and right edges of the shell, you must add two dimensions and associate them with parameters. You can then define a parametric equation, *dim2*=(*dim1*)/2, to set the distance from the left edge to the post at half the width of the shell, as shown in the following figure:



Note:

The distance dimension associated with the parameter *dim1* is defined in the sketch of the shell feature and cannot be edited in the sketch of the post. When you dimension reference geometry, Abaqus/CAE colors the dimension magenta to indicate an overconstraint. To clear the overconstraint, you must make *dim1* a reference dimension. Abaqus/CAE places parentheses around the value of reference dimensions (or around the parameter name if the dimension is associated with a parameter) and automatically updates their values if the dimensioned quantity changes. For more information, see *Editing dimensions*.

You can modify any dimension or parameter in a sketch. For detailed instructions, see *Editing dimensions*, and *Adding and editing parameters*, respectively.

Modifying or copying objects by selecting edges

When you move edges, Abaqus/CAE repositions the selected edges and uses the **During edit** constraint solution method to modify any connected edges (for more information, see *Customizing the use of constraints in the Sketcher*). The sketch constraints and constraint solution method may restrict the types of moves that you can make or cause Abaqus/CAE to reposition edges that you did not select. When you copy edges, the copies are positioned according to the copy method—the copied edges are not affected by any existing sketch constraints. You cannot copy objects between sketches. Abaqus/CAE provides the following methods for moving or copying selected edges:

Translate

You can translate selected edges by specifying the start and end coordinates of a translation vector. The translation vector can start from any point; however, you may find it more meaningful to start the vector from the vertex at one end of a selected edge and end it at the vertex's new location. *Figure 1* shows the selected edges, the translation vector, and the result of the translation. To achieve the results shown, **Fixed** constraints were added to prevent the two vertices of the upper rectangle from moving.

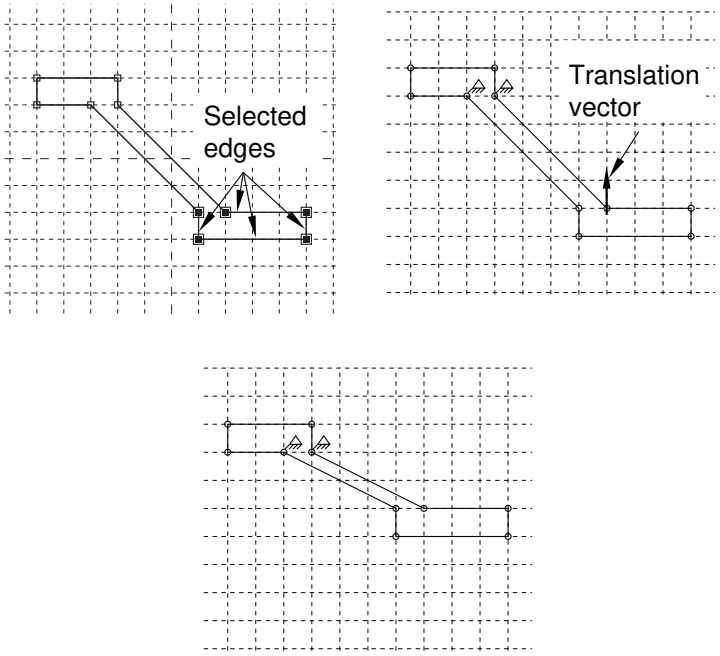


Figure 1: Translating selected edges.

Rotate

You can rotate selected edges by specifying the coordinates of the center of rotation and entering the angle of rotation; a positive angle indicates a counterclockwise rotation. *Figure 2* shows the selected edges, the selected center of rotation, and the result of a 90° rotation. The selected edges are not connected or otherwise constrained to other edges in the sketch, so the constraint solution method has no effect on the rotation.

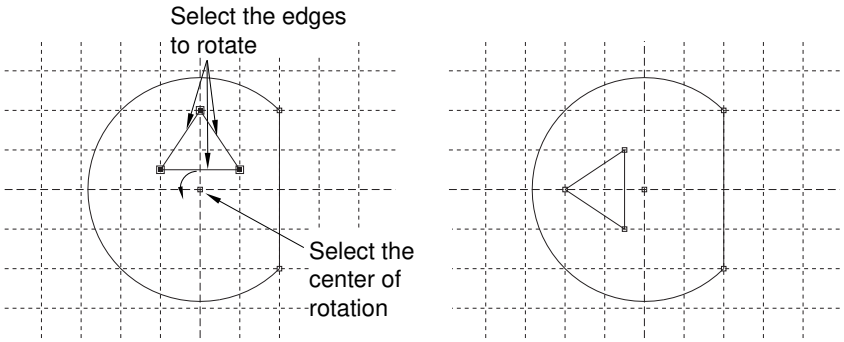


Figure 2: Rotating selected edges.

Mirror

You can mirror selected edges by selecting a straight line in the sketch as the mirror line. *Figure 3* shows the selected edges, the mirror line, and the resulting copy including a mirror constraint for each of the four selected edges. The mirror line may be any straight object line or construction line in the sketch.

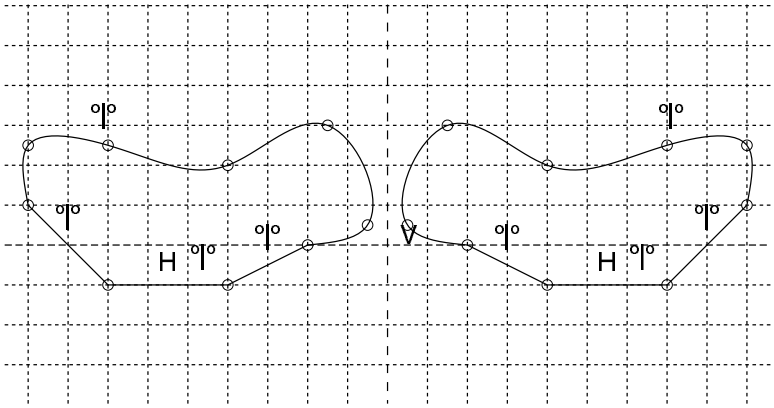


Figure 3: Mirroring selected edges.

Scale

You can scale selected edges by specifying a center point and a scale factor. A scale factor less than 1.0 decreases the spacing between the edges, and a scale factor greater than 1.0 increases the spacing between the edges. *Figure 4* shows the selected edges, the selected center point, and the result for a scale factor of 1.5.

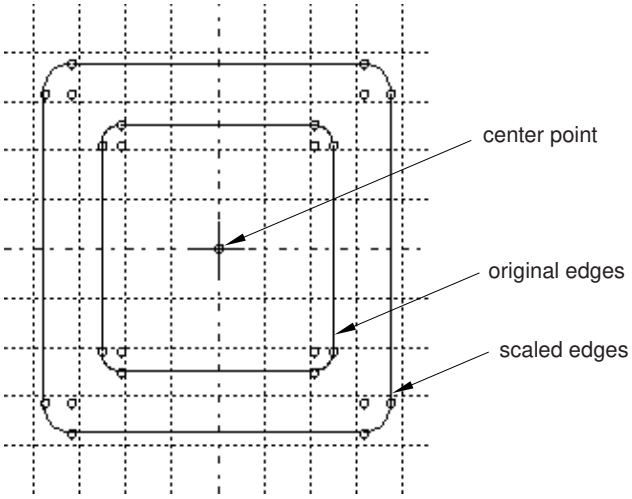


Figure 4: Scaling selected edges.

Additional information

• *Modifying objects*

Modifying edges by trimming, extending, splitting, or merging

When you are modifying a sketch, you can implement your changes by modifying a dimension, by moving selected vertices, or by trimming, extending, splitting, or merging edges. Abaqus/CAE provides the following methods for trimming, extending, splitting, and merging edges:

Trim/Extend

You can trim or extend one end of a line or arc; you can also trim a spline curve, but spline curves cannot be extended. To trim or extend an edge, first select the edge near the end that you want to modify and then select a second edge to define an intersection point. The second edge may be any object in the sketch, including construction geometry. The intersection point may lie beyond the current endpoints of either selected edge. Abaqus/CAE trims or extends the first edge at the intersection point. If you want to trim or extend the second edge to the same point, repeat the process, reversing the selection order (see *Figure 1*).

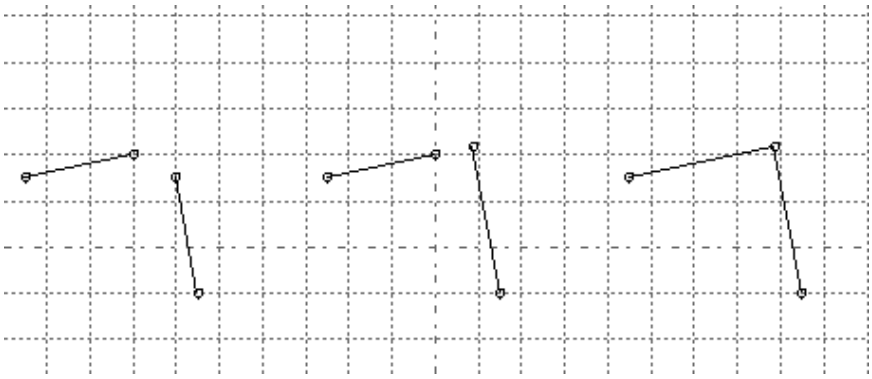


Figure 1: Extending two edges to create a corner.

For detailed instructions on trimming and extending edges, see *Modifying Sketcher objects by trimming or extending edges*.

Auto-trim

You can trim an intermediate or end segment from a line, arc, circle, or spline curve. **Auto-trim** is the fastest method you can use to remove unwanted edge segments from a sketch. Abaqus/CAE uses preselection to highlight edges that you can remove. Preselection is based on cursor proximity and the two nearest "trim points." Trim points include intersection points, endpoints, vertices, and the picked points that define the edges of circles. Unlike when you use **Trim/Extend**, Abaqus/CAE does not split the intersecting edges. *Figure 2* shows several possible trim combinations based on the circle and line shown in the upper left corner.

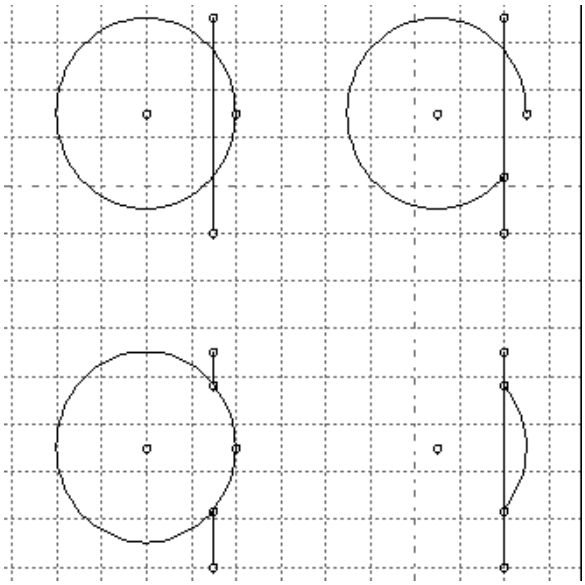


Figure 2: Auto-trim selections.

For detailed instructions on automatically trimming edges, see *Modifying Sketcher objects by automatically trimming edges*.

Split

You can split edges, creating separate pieces with a common vertex, where they intersect in the sketch. Select the first edge to be broken, then select a second edge that intersects the first at the desired split point. As you move the cursor near the second edge, a circle appears around the split point that you would create. If there is more than one intersection point between the two edges, Abaqus/CAE highlights the intersection closest to the cursor. Abaqus/CAE splits the first edge at the indicated point. *Figure 3* shows an intersecting line and spline on the left, indicates the split point in the center image (the cursor is not shown), and shows the three resulting edges on the right. In this case the spline was selected first, so the straight line remains unbroken.

Note:

Spline curves retain their exact shape when you split or trim them, but the control points are removed so you can no longer edit the shape of the curves.

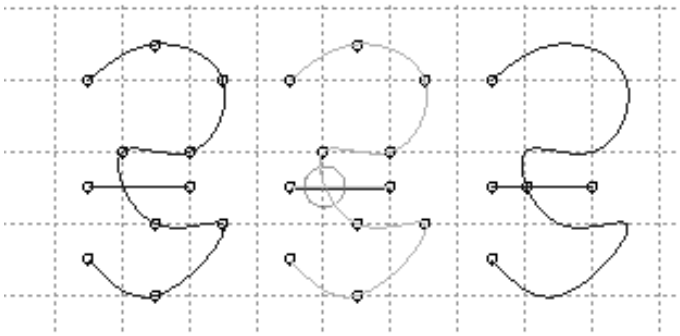


Figure 3: Splitting a line and a spline.

For detailed instructions on splitting edges, see *Modifying Sketcher objects by splitting edges*.

Merge

The Merge tool enables you to close small gaps in a sketch. These gaps commonly occur in the following situations:

- When you use the **Project Edges** tool on edges that are not in a plane parallel to the sketch plane, the endpoints may be created in a slightly different location from other selected edges.
- When you import sketch geometry into an existing sketch, the edges may not line up exactly.

The Merge tool is intended only for closing small gaps. Merging edges to close larger gaps may change the sketch geometry significantly. If you want to move elements in a sketch over a large distance, use the drag tool from the Sketcher toolbox to move vertices in your sketch to a new location. For more information, see *Dragging Sketcher objects*.

For detailed instructions on merging edges, see *Modifying Sketcher objects by merging edges*.

Additional information

• *Modifying objects*

Copying sketch objects to create patterns

You can copy objects in your sketch and create a pattern of copied objects; for example, *Figure 1* illustrates radial patterns of gear teeth and holes on a solid part.

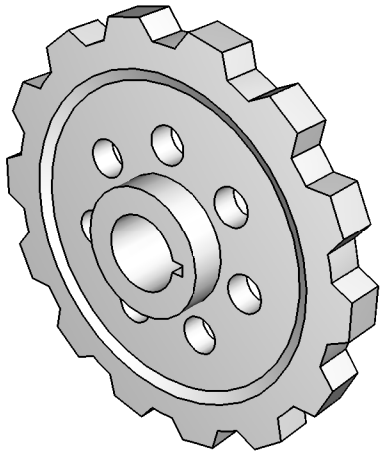


Figure 1: Radial patterns of gear teeth and holes.

You can choose either of the following methods to define the pattern:

Linear pattern

A linear pattern positions copies of selected objects along a direction; for example, the *X*-direction. You can specify the number of copies and the spacing between the copies; and you can choose to position the copies in a positive direction or a negative direction. In addition, you can change the orientation of the linear pattern. *Figure 2* shows how you can create different linear patterns in a single direction.

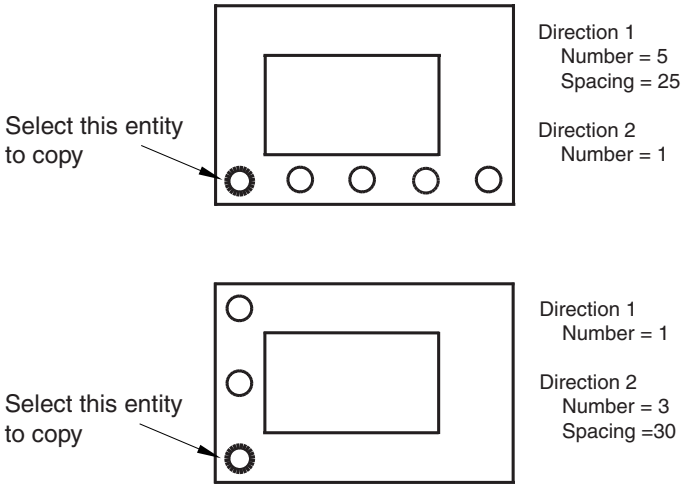


Figure 2: Linear patterns in a single direction.

You can create a matrix of copied objects by creating copies in a second direction; for example, the *Y*-direction. The options are the same as for the first direction; you can control the number of copies, the direction, and the angle. *Figure 3* shows how you can create different linear patterns in two directions.

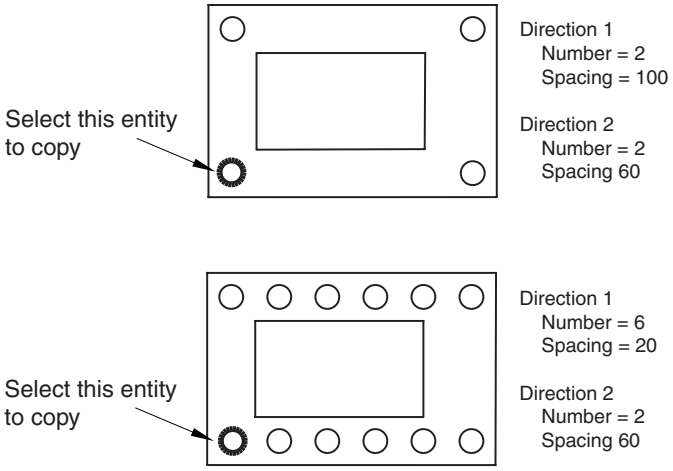


Figure 3: Linear patterns in two directions.

By default, the first direction is the *X*-axis and the second direction is the *Y*-axis. However, you can change the orientation of the first direction or the second direction by selecting a line from the sketch that represents the new orientation. *Figure 4* shows a linear pattern oriented along a selected line.

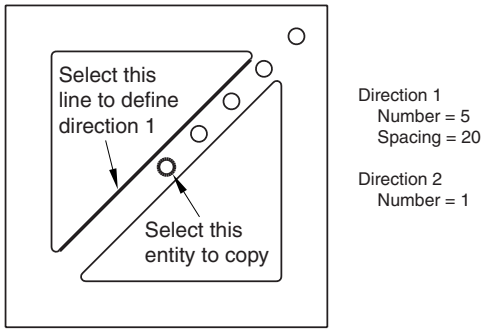


Figure 4: A linear pattern oriented along a line.

By default, Abaqus/CAE creates the pattern in a positive direction; however, you can reverse the direction. For example, *Figure 5* shows the same linear pattern as in *Figure 4* but with the direction reversed.

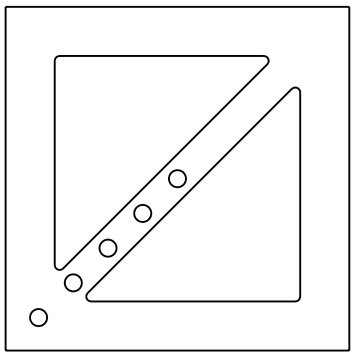


Figure 5: The effect of reversing the direction of a linear pattern.

For detailed instructions on creating linear patterns, see *Creating linear patterns of objects*.

Radial pattern

A radial pattern positions copies of selected objects in a circular pattern. You can specify the number of copies, and you can specify the angle between the first and last copy, where a positive angle corresponds to a counterclockwise direction. In addition, you can select a point from the sketch that defines the center point of the circular pattern. *Figure 6* shows how you can create different radial patterns.

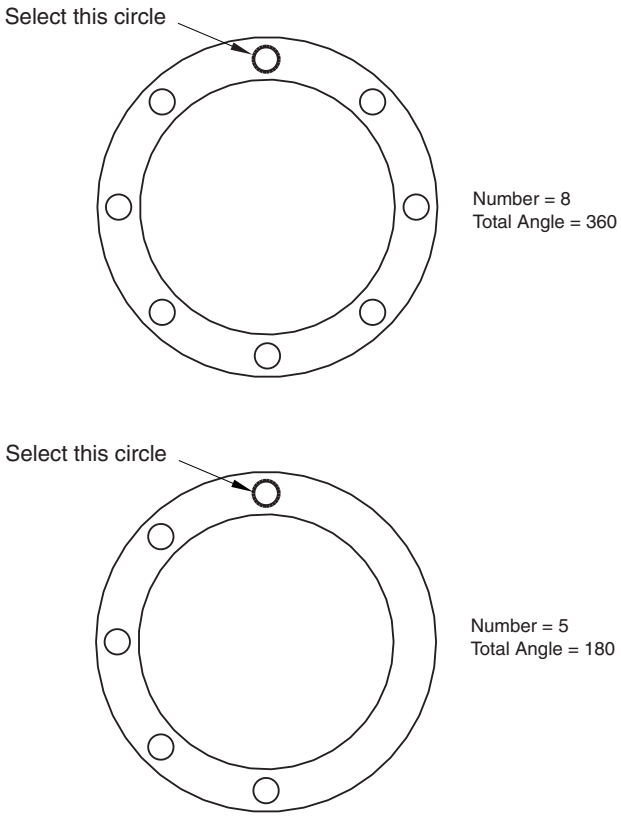


Figure 6: Radial patterns.

For detailed instructions on creating radial patterns, see *Creating radial patterns of objects*.

By default, Abaqus/CAE displays a preview of the linear and radial pattern as you enter the settings. In most cases the preview will help you decide on the settings that will create the desired pattern. However, if you are creating a large number of copies, the preview may slow down the generation of the sketch. In this case you should toggle off the **Preview** button.

Several methods are also available for creating individual copies of sketch objects. These methods are discussed in *Modifying or copying objects by selecting edges*.

Offsetting objects

You can offset objects in your sketch to create similar objects of a larger or smaller size. You select the edges to offset and the offset distance; Abaqus/CAE displays a preview so that you can choose the offset direction. You can offset any continuous open or closed loops that do not contain branches. *Figure 1* shows the two possible offsets that can be created for several edge selections. (For clarity, vertices and datum points are not shown in the figure.)

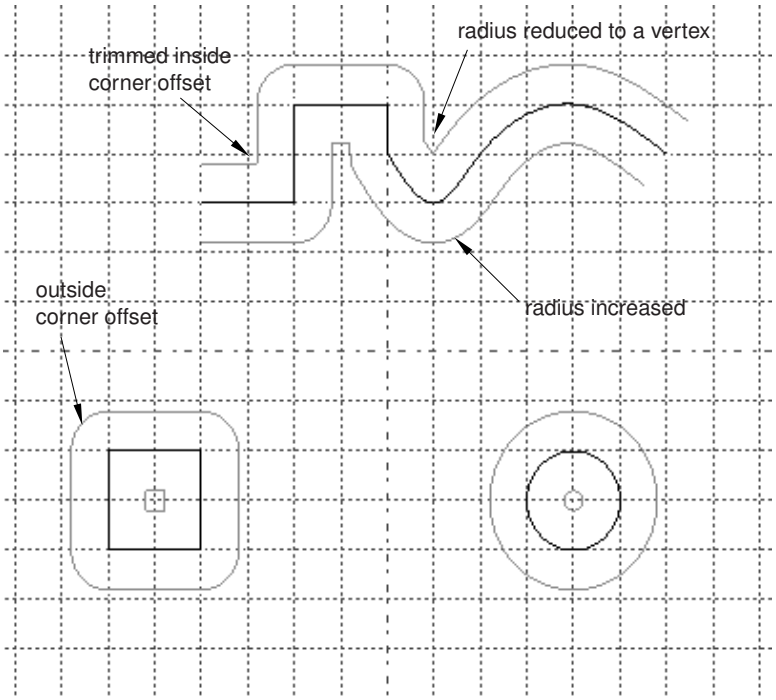


Figure 1: Offsets of a spline curve, a square, and a circle.

Abaqus/CAE offsets each object in the same way that you would offset edges on paper with a drafting compass. Using a compass, you would draw the compass point along the original edge tracing a new edge with the pencil at the required perpendicular (offset) distance. At any sharp corners, you would rotate the compass to make it perpendicular to the second edge at the corner. At the end of this process, you would trim any offset edges that intersected with other offset edges.

As shown in *Figure 1*, Abaqus/CAE completes a similar process:

- Straight edges are copied.
- Offset edges that overlap, such as corners offset toward the “inside,” are trimmed to make new corners.
- Where corners are offset toward the “outside,” you can choose to create sharp corners or fillets with radii equal to the offset distance.
- Radii of existing curves are increased or decreased by the offset distance.

If the radius of a curve would be decreased to zero or less by the offset, Abaqus/CAE creates a new vertex joining the surrounding edges that were previously connected by the curve. For detailed instructions, see *Offsetting the edges of Sketcher objects*.

Customizing the Sketcher

This section describes how you customize the behavior and the appearance of the Sketcher.

In this section:

- *Using Sketcher customization options*
- *Turning snapping on or off*
- *Turning preselection on or off*
- *Customizing the sheet size and grid*
- *Realigning the sketch grid*
- *Displaying and hiding construction geometry*
- *Limiting the projection of coplanar entities*
- *Setting the maximum number of recorded sketching operations*
- *Customizing the format and use of dimensions in the Sketcher*
- *Customizing the use of constraints in the Sketcher*
- *Managing images in the Sketcher background*

Using Sketcher customization options

Use the Sketcher customization options to control the appearance and behavior of the Sketcher.

The following options control the behavior of the Sketcher throughout the Abaqus/CAE session:

- Whether the cursor *snaps* to the grid.
- Whether *preselection* is active.
- Whether dimensions are displayed.
- Whether names or values of parameters are displayed.
- Limiting the number of coplanar entities that are projected.
- The types of dimensions and constraints available for use with the **Auto-Dimension** and **Auto-Constrain** tools.
- The *constraint solution method* for use while editing or dragging objects.
- Whether *sketch constraints* are displayed.
- Whether implied constraints are added during entity creation.
- Whether *construction geometry* is displayed.
- Whether a *background image* is displayed.

The following options are stored with each sketch; they control the behavior and appearance of the current sketch and new sketches that you create:

- The sheet size, the *grid* spacing, and the number of gridlines to display.
- Whether the sheet size and grid spacing are controlled automatically.
- Whether the grid is *visible*.
- The text font size and the number of decimal places displayed in *sketch dimensions*.

The origin of the sketch grid and the alignment of the *X*-axis are also stored with each sketch. However, the grid origin and alignment settings control only the behavior and appearance of the current sketch; they are reset to their default values for any new sketch that you create.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .
   The **Sketcher Options** dialog box appears.
2. Set the desired customization options on each of the tabbed pages. For detailed help, request context-sensitive help on individual items in the dialog box.

    **Tip:** You can also click **Defaults** to revert all Sketcher options to their default settings.
3. Click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Turning snapping on or off

Snapping to the grid helps you position the cursor when you want to select a Sketcher grid point or drag an object to a specific point on the grid. When snapping is enabled and you move the cursor near a grid point, the secondary cursor jumps, or snaps, to the grid point. Toggle **Snap to grid** in the **Sketcher Options** dialog box to enable or disable snapping. (By default, snapping is enabled when you start the Sketcher.) The behavior you choose for snapping applies to all sketches in the Abaqus/CAE session. Sketcher preselection overrides snapping if a preselection point and a grid point are both close to the cursor position.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .

   The **Sketcher Options** dialog box appears.
2. From the **General** tabbed page, toggle **Snap to grid**.

   When **Snap to grid** is on, the secondary cursor snaps to nearby grid points as you move around the sketch. Depending on the grid size, magnification of the viewport, and whether preselection is active, you may also be able to select other points in addition to the grid points and the existing sketch geometry.

   When **Snap to grid** is off, the secondary cursor aligns with the primary cursor as you move around the sketch. You can select anywhere on the sheet.
3. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *The Sketcher sheet and grid*
- *The Sketcher cursors and preselection*
- *Customizing the Sketcher*
- *Turning preselection on or off*

Turning preselection on or off

Preselection helps you select reference geometry or Sketcher objects, such as lines, vertices, and dimensions. When preselection is enabled and you move the cursor near one of these objects, Abaqus/CAE highlights the object so that you can more easily select it. Preselection also helps constrain your sketch by indicating selections that will create horizontal, vertical, colinear, perpendicular, and coincident constraints while you sketch. For a list of the preselection symbols displayed by different Sketcher objects as you move the cursor around the sketch, see *The Sketcher cursors and preselection*.

Toggle **Preselect geometry** in the **Sketcher Options** dialog box to enable or disable preselection. (By default, preselection is enabled when you start the Sketcher.) Preselect customization is stored for the duration of the Abaqus/CAE session and applies to all sketches. Preselection overrides snapping to the grid if a preselection point and a grid point are both close to the cursor position.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .

   The **Sketcher Options** dialog box appears.

2. From the **General** tabbed page, toggle **Preselect geometry**.

   When **Preselect geometry** is on, Abaqus/CAE highlights valid selections, such as vertices, midpoints, and dimensions, as you move around the sketch.

   When **Preselect geometry** is off, Abaqus/CAE does not highlight valid selections.

3. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *The Sketcher cursors and preselection*
- *Customizing the Sketcher*
- *Turning snapping on or off*
- *How Sketcher customization options are initialized and saved*

Customizing the sheet size and grid

Use the **Sketcher Options** dialog box to control the sheet size and the grid.

You can customize the following:

Sheet size

The boundary of the sheet is always square, and its height and width are equal to the sheet size. If you find that the sheet size is too large or too small, you can use the **Sketcher Options** dialog box to change the size.

When you create a part or a stand-alone sketch, you define the approximate size of the new part or sketch; Abaqus/CAE bases the initial sheet size on the approximate size that you provide. The approximate size must be between 10^{-3} and 10^{4} units for a part and between 10^{-3} and 10^{5} units for a sketch. Abaqus/CAE does not use specific units, but the units must be consistent throughout the model. The **Auto** toggle next to the **Sheet size** field unlocks the field and controls whether or not Abaqus/CAE can automatically change the sheet size in the current sketch and in new sketches that you create.

Grid spacing

You can use this option to change the grid spacing for major gridlines using the same units that define the **Sheet size**. If ***Snap to grid*** is enabled, the cursor will snap to each grid point. The **Auto** toggle next to the **Grid spacing** field unlocks the field and controls whether or not Abaqus/CAE can automatically change the grid size in the current sketch and in new sketches that you create.

Minor intervals

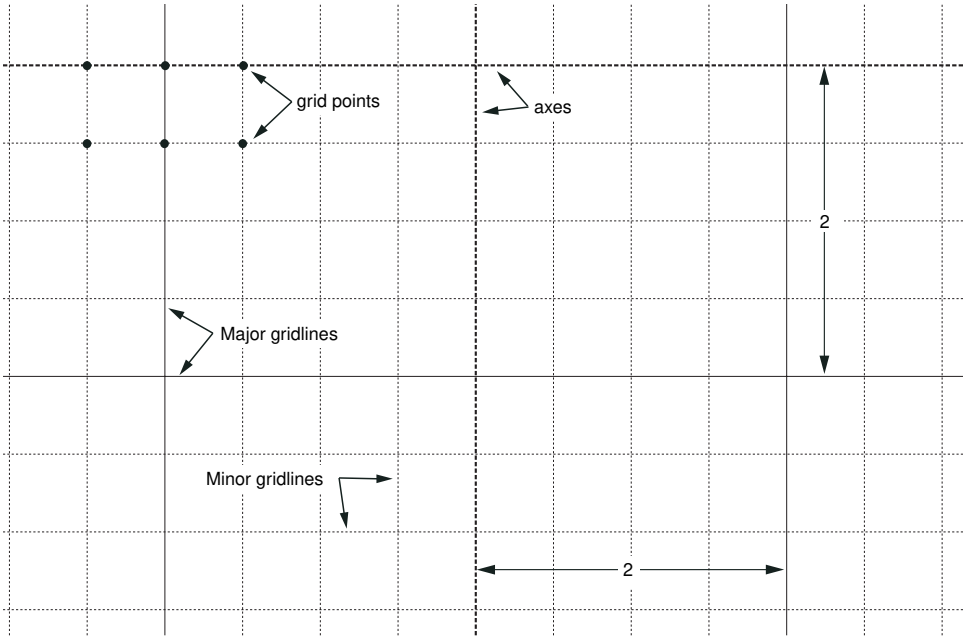
Minor gridlines subdivide the space between major gridlines into intervals and enable you to position items in your sketch with greater precision when you have the Sketcher set to a high magnification level. You can use this option to change the number of minor intervals between each major gridline.

The display of minor gridlines is dynamic: Abaqus/CAE hides these lines at the default magnification level and reveals them as you zoom in.

Show gridlines

You can toggle the display of major gridlines and minor gridlines in the Sketcher. Minor gridlines can be displayed only when major gridlines are displayed as well.

The following figure shows the relationship between the grid spacing, major gridlines, and minor gridlines:



Sheet size and grid customization options apply to the current sketch and are stored along with the sketch. When you create a new sketch, Abaqus/CAE uses the most recent settings to determine whether or not to recalculate the sheet size and grid spacing and whether to display the grid.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .

   The **Sketcher Options** dialog box appears.
2. From the **General** tabbed page, toggle off **Auto** next to the **Sheet size** text box.
3. In the text box, type the dimension of the square sheet that will contain the feature being created or edited.

   Specify the sheet size in units consistent with those used to describe the rest of the model.
4. Toggle off **Auto** next to the **Grid spacing** text box.
5. In the text box, type the desired spacing between grid points.

   Specify the grid spacing in the same units that you used to define the sheet size.
6. Click the arrows to the right of the **Minor intervals** field to increase or decrease the number of minor grid intervals displayed between each major gridline.
7. Toggle **Major** to show or hide the major gridlines.

   If major gridlines are displayed, you can also toggle **Minor** to show or hide the minor gridlines.
8. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *The Sketcher sheet and grid*
- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Realigning the sketch grid

In some circumstances the sketch grid does not align with the vertices and lines of the sketch or the underlying reference geometry. You may find it easier to create the desired sketch if you use Sketcher customization options to realign the sketch grid. If you add an existing sketch to a new blank sketch (such as for a new part), Abaqus/CAE automatically realigns the sketch grid. The customization options allow you to do the following:

- Shift the grid relative to the sketch by selecting a vertex representing the origin of the grid. To help you locate the origin of the sketch, the *X*- and *Y*-axes of the sketch are indicated by dashed lines that intersect at the origin.
- Rotate the grid relative to the sketch by selecting a line from the sketch that will be parallel to the *X*-axis of the grid.
- Reset the grid to the original sketch coordinates.

Realigning the grid does not change existing features in the sketch. However, the change is applied to any new sketch entities. For example, if you rotate the grid, an existing horizontal construction line will not change; however, a new horizontal construction line will be parallel with the new *X*-axis. Dimensions are an exception; horizontal and vertical dimensions—new and existing—align with the default grid rotation.

When you change the grid origin or rotation and create new sketch geometry, Abaqus/CAE displays two sets of cursor coordinates. The grid coordinates are relative to the current alignment of the sketch grid; they are displayed in the upper-left corner of the viewport. The sketch coordinates are relative to the original alignment of the sketch grid; they are displayed in the upper-right corner of the viewport and do not change if you realign the grid. When both sets of coordinates are the same, Abaqus/CAE displays a single coordinate set in the upper-left corner of the viewport.

1. From the bottom of the Sketcher toolbox, select the customization tool .

   The **Sketcher Options** dialog box appears.
2. From the buttons across the center of the **General** tabbed page, do the following:
   - Click **Origin** to move the origin of the sketch grid relative to the part. Select the new origin from the sketch or type the coordinates of the origin in the prompt area.
   - Click **Angle** to rotate the sketch grid relative to the part. Select a line that will be parallel to the *X*-axis of the sketch.
   - Click **Reset** to restore the original sketch origin and to restore the original alignment of the *X*-axis. Resetting the grid aligns the grid coordinates with the sketch coordinates.

   Your grid alignment changes are applied to the sketch immediately.
3. Click **OK** to apply any other customizations that you made and to close the **Sketcher Options** dialog box.

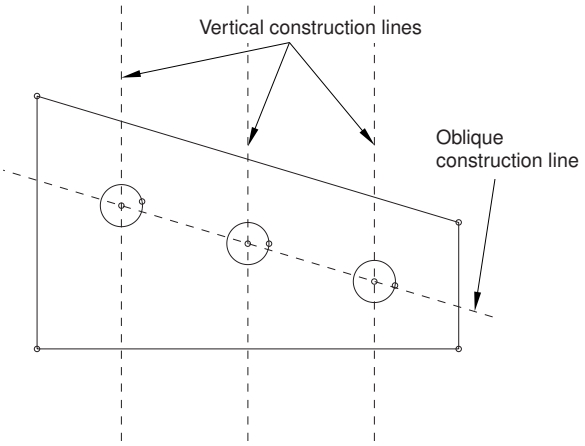
Additional information

- *Realigning the sketch grid relative to the sketch*
- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Displaying and hiding construction geometry

You can create to help you position and align objects in your sketch.

For example, the following figure shows a series of holes aligned along an oblique construction line. The holes are centered where the vertical construction lines intersect the oblique construction line:



If the construction geometry becomes distracting, you can use the **Show construction geometry** option in the **Sketcher Options** dialog box to hide it. (By default, construction geometry is displayed when you start the Sketcher.) If preselection is enabled, the cursor will still snap to items associated with hidden construction geometry, such as the intersection of a line with a construction line. Customization of the display of construction geometry applies to the current sketch and is stored along with the sketch.

1. From the bottom of the Sketcher toolbox, select the customization tool .
   The **Sketcher Options** dialog box appears.
2. From the **General** tabbed page, toggle **Show construction geometry**.
   When **Show construction geometry** is on, Abaqus/CAE displays construction geometry in the sketch.
3. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Construction geometry*
- *Creating construction geometry*
- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Limiting the projection of coplanar entities

When a sketch-based feature is created or edited, you can set a limit for the number of coplanar entities that Abaqus/CAE automatically projects from the background. By default, Abaqus/CAE automatically projects 300 coplanar entities. The limit you choose for projected coplanar entities applies to all sketches in the Abaqus/CAE session.

1. From the bottom of the Sketcher toolbox, select the customization tool .
   The **Sketcher Options** dialog box appears.
2. From the **General** tabbed page, enter a value in the **Max coplanar entities to project** text field.
   If this value is exceeded, a warning message is issued and no entities are automatically projected.
3. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Setting the maximum number of recorded sketching operations

Abaqus/CAE records your changes to the current sketch in a cache that enables you to undo and redo several consecutive sketching changes. By default, the undo cache supports up to 10 levels of undo, but you can increase or decrease this number. You might want to increase the number of levels if your sketching operations require large-scale changes, and you might want to decrease this number to improve system performance for large, complex sketches.

The maximum number of undo levels applies to all sketches in the Abaqus/CAE session. Abaqus/CAE clears the undo cache each time you change the current sketch or switch to a different module.

1. From the bottom of the Sketcher toolbox, select the customization tool .

   The **Sketcher Options** dialog box appears.
2. From the **General** tabbed page, enter a value from 0 to 100 in the **Max level for sketch undo** text field.
3. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Customizing the Sketcher*
- *Undoing and redoing sketching actions*

Customizing the format and use of dimensions in the Sketcher

When you add *dimensions* to your sketch, the default dimension text height is 12 points. You can change the text height by using the arrows in the **Dimension text height** field on the **Dimensions** tabbed page of the **Sketcher Options** dialog box. The dimension text is independent of the current sheet size and zoom settings; that is, the text remains at the size you chose regardless of the apparent size of the dimensioned feature.

In addition, you may want to change the number of decimal places to match the dimensions of the feature you are creating. You can use the **Decimal places** field to control the number of decimal places displayed in each dimension. To reduce clutter while editing a sketch, you can use the **Show dimensions** option to hide the dimensions. You can also use the **Show parameter names** option to display the names of dimensions that have been converted to parameters instead of their numerical values. Displaying parameter names allows you to view dimensions that have been replaced by parametric expressions and helps you to create relationships between various sketch dimensions. Finally, you can select the types of dimensions that can be created automatically with the **Auto-Dimension** tool.

Dimension text height and decimal place settings apply to the sketch you are working on and are stored along with the sketch. When you create a new sketch, Abaqus/CAE uses the default dimension text height and number of decimal places for the new sketch. All other dimension options are stored for the remainder of the current Abaqus/CAE session.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .

   The **Sketcher Options** dialog box appears.
2. Click on the **Dimensions** tab.

   Abaqus/CAE displays the **Dimensions** tabbed page.
3. If desired, toggle **Show dimensions**.

   When **Show dimensions** is on, Abaqus/CAE displays dimensions in the sketch.
4. If desired, toggle **Show parameter names**.

   When **Show parameter names** is on, Abaqus/CAE shows the names of dimensions that have been converted to parameters. (For more information, see *Adding and editing parameters*.)
5. Next to the **Dimension text height** label, click the arrows to increase or decrease the height of the dimension text.

   Specify the height in points; the height can vary from 8 to 30.
6. Next to the **Decimal places** label, click the arrows to increase or decrease the number of decimal places that will be included in the dimension text.

   The number of decimal places displayed can vary from one to six.
7. In the lower portion of the page, toggle the dimension types available for use with the **Auto-Dimension** tool. (For more information, see *Automatically dimensioning a sketch*.)
8. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Constraining, dimensioning, and parameterizing a sketch*
- *Customizing the Sketcher*
- *How Sketcher customization options are initialized and saved*

Customizing the use of constraints in the Sketcher

Constraints create non-numerical relationships that control the geometry of a sketch. As you edit a sketch, Abaqus/CAE applies solution methods to avoid violating the constraints. Several methods are available for use either when entities are *dragged* or when they are edited using other sketcher tools.



Note:

The constraint solution method does not prevent sketch modification tools from overriding applied constraints. For example, you can use the **Translate** tool to move a line that has a **Fixed** constraint.

In addition, you can use the **Show constraints** option to hide the constraints and reduce clutter in the sketch and the **Add constraints during entity creation** option to add the constraints that are implied during sketching. You can select the linear and angular tolerance and the types of constraints that can be created automatically with the **Auto-Constrain** tool.

Constraint customization options apply to the entire Abaqus/CAE session. They are not saved with individual sketches or with the current model.

1. From the bottom of the Sketcher toolbox, select the Sketcher customization tool .

   The **Sketcher Options** dialog box appears.
2. Click on the **Constraints** tab.
3. Select the desired constraint solution methods for use while dragging sketch entities or during other types of editing. The following selections are available:

Standard

This is the default solution method for editing. Abaqus/CAE moves or modifies the geometry by the minimum amount to satisfy the constraints and dimensions. Abaqus/CAE also attempts to maintain a rigid body appearance of the uninvolved geometries during the move.

Weighted

Abaqus/CAE moves or modifies the geometry by the minimum amount to satisfy the constraints and dimensions. The rigid body appearance of the uninvolved geometries are not considered during the move.

Minimum move

This is the default solution method for dragging. Abaqus/CAE moves the minimum number of entities to satisfy the constraints.

Relaxation

Abaqus/CAE moves the geometry using a numerical solution technique to minimize the sum of the squares of the movements. This solution method results in more geometries being moved compared to other methods.

4. If desired, toggle **Show constraints**.

   When **Show constraints** is on, Abaqus/CAE displays constraint symbols in the sketch.
5. If desired, toggle **Add constraints during entity creation**.

When **Add constraints during entity creation** is on, Abaqus/CAE adds the constraints that are implied during sketching. For example, if you are sketching a line for which the preselection symbol indicates Abaqus/CAE will apply a horizontal constraint and **Add constraints during entity creation** is on, the horizontal constraint is added to the line.

6. In the lower portion of the page, toggle the constraint types available for use with the **Auto-Constrain** tool. (For more information, see *Automatically constraining a sketch*.)
7. After you have chosen the desired customization options, click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *How Sketcher customization options are initialized and saved*
- *Customizing the Sketcher*
- *Constraining, dimensioning, and parameterizing a sketch*

Managing images in the Sketcher background

You can display an image in the background of the Sketcher to help you draw and align objects in your sketch. This image can be positioned in any location in the Sketcher background and can be stretched or compressed horizontally or vertically. When you resize the viewport, Abaqus/CAE resizes the background image proportionally.

The Sketcher background image appears in the Sketch module only. You can also display a different image that appears in the viewport background in every Abaqus/CAE module, including the Sketch module; see *Working with background images and movies in viewports* for more information. When both the Sketcher and the module-wide background images are toggled on, Abaqus/CAE displays the Sketcher background image on top when you enter the Sketch module.

Before you can display an image in the background of the Sketch module or any other module, you must add the image to your Abaqus/CAE session. To add an image from the **Sketcher Options** dialog box, click **Create**; then enter a name for the image and provide its location. Images in your session are available in all modules and persist for your session only; they are not saved to the model database or output database.

Abaqus/CAE supports background images in the following formats: Bitmap (`.bmp`), PNG (`.png`), TIFF (`.tif`), XPM (`.xpm`), PCX (`.pcx`), ICO (`.ico`), TGA (`.tga`), and RGB (`.rgb`).

1. From the bottom of the Sketcher toolbox, select the Sketcher options tool .
   The **Sketcher Options** dialog box appears.
2. Click the **Image** tab.
3. Toggle **Show image** to display or hide the Sketcher background image.
4. Select an image file to display:
   - To display an image that has been defined in your session, expand the **Image name** list and select the image name.
   - To add a new image, click **Create**; then enter a name and specify a file location in the dialog box that appears. You can either enter the file location directly in the **File Name** field or click  to navigate to it in the **Select Image File** dialog box.
5. Stretch or compress the background image using one of the following techniques:

Scaling the image along either axis

You can stretch or compress the Sketcher background image by specifying **X scale** and **Y scale** values. When these scale values are equal to 1 (the default value), the image retains its original scale. You can increase the scale value to stretch the Sketcher background image along the selected axis, or you can decrease the value to compress the image along the selected axis.

Calibrate image size

You can scale the size of the Sketcher background image by calibrating the distance between two points on the image. Click **Calibrate**, click two points in the viewport; then enter in the Prompt area the desired distance between the two points.

After you calibrate an image, Abaqus/CAE reflects the changes in the **X scale** and **Y scale** fields.

6. Reposition the Sketcher background image using one of the following techniques:
   - Enter the Sketch coordinates to which you want to move the lower left corner of the Sketcher background image into the **Origin X** and **Origin Y** fields.

- Click **Pick**, then click a point on the Sketcher background image.

  Abaqus/CAE moves this point of the image to the grid origin and populates the **Origin** fields with the new origin location.



Note:

The lower left corner of the image is used to fix the position of the background image. If you change the scaling after positioning the background image, the center of the image will move.

7. Drag the **Translucency** slider to the level of translucency with which you want to display the Sketcher background image. Background images are opaque by default, and you can select a value from 0.00 (transparent) to 1.00 (opaque).
8. Click **OK** to apply your changes and to close the **Sketcher Options** dialog box.

Additional information

- *Working with background images and movies in viewports*
- *Customizing the Sketcher*

Sketching simple objects

This section describes how to use the Sketcher tools to draw simple objects.

In this section:

- *Sketching an isolated point*
- *Sketching lines and polygons*
- *Sketching rectangles*
- *Sketching circles*
- *Sketching arcs using a center point and two endpoints*
- *Sketching arcs through three points*
- *Sketching arcs tangent to a line*
- *Sketching ellipses*
- *Sketching fillets between two lines*
- *Sketching splines*

Sketching an isolated point

Use the point tool from the Sketcher toolbox to draw a single isolated point. You can use the resulting point as a reference, and you can create dimensions between the point and vertices on your sketch.

1. From the Sketcher toolbox, select the point tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click at the desired location of the point.

   The point appears.
3. To create more points, repeat the previous step.
4. When you have finished creating points, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

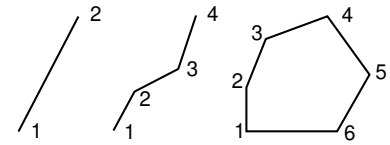
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching lines and polygons

Use the line tool from the Sketcher toolbox to draw lines, connected lines, or polygons.

The following figure shows how you draw lines, connected lines, and polygons by clicking the locations shown, in the order indicated below:



Line Connected Line Polygon

You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

1. From the line tools in the Sketcher toolbox, select the connected lines tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. To construct a simple line, click the two end points. To construct a connected line or a polygon, click each vertex.

    **Tip:** If necessary, you can use the text box in the prompt area to enter the precise coordinates of the vertices of the line. For more information on precisely defining the line, see *Specifying precise geometry*.

   The line or polygon appears as you click a vertex or enter the coordinates.

3. To complete the line or polygon, click mouse button 2.

    **Tip:** If you make a mistake while constructing a connected line or a polygon, click the **Undo** tool  in the Sketcher toolbox to delete the most recent line segment. If you make a mistake in an earlier segment, you can delete the incorrect segments using the **Delete** tool  and redraw them with the line tool.

4. To create more lines or polygons, repeat the above steps beginning with Step 2.
5. When you have finished creating lines and polygons, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

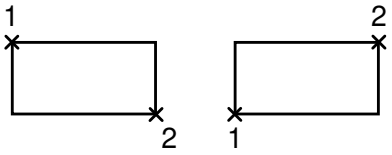
- *Sketching simple objects*
- *Creating construction geometry*

- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching rectangles

Use the rectangle tool from the Sketcher toolbox to draw rectangles.

To draw a rectangle, click at any two opposite corners as indicated by the numbering in the following figure.



You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

1. From the line tools in the Sketcher toolbox, select the rectangle tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click the desired locations of any two opposite corners of the rectangle.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of the corners of the rectangle. For more information on precisely defining the rectangle, see *Specifying precise geometry*.

   The rectangle appears, aligned with the current Sketcher grid, as you move the cursor or enter the coordinates.
3. To create more rectangles, repeat the previous step.
4. When you have finished creating rectangles, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

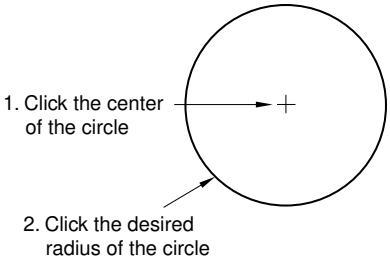
Additional information

- *Realigning the sketch grid*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching circles

Use the circle tool from the Sketcher toolbox to draw circles based on a center point and any arbitrary point on the circumference of the circle.

An example of using the circle tool from the Sketcher toolbox is shown here:



You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

1. From the Sketcher toolbox, select the circle tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click at the desired location of the center of the circle.
3. Move the cursor to a point on the circumference of the circle.

   Abaqus/CAE displays a preview of the circle that would be created using the current cursor position.
4. Click any point on the circumference of the desired circle.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of the center and the point on the circumference of the circle. For more information on precisely defining the circle, see *Specifying precise geometry*.

   Abaqus/CAE draws the circle.
5. To create more circles, repeat the above steps beginning with Step 2.
6. When you have finished creating circles, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

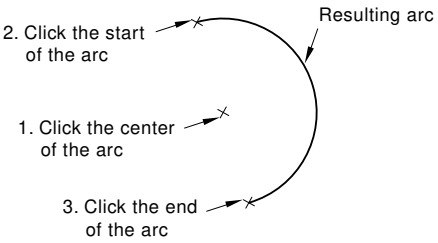
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching arcs using a center point and two endpoints

Use the Center with Two Endpoints arc tool from the Sketcher toolbox to draw arcs using a center point and two endpoints.

The following figure shows the resulting arc:



An arc that forms part of an analytical *rigid surface* cannot subtend an angle greater than 180°. If necessary, append two arcs to create an arc that subtends an angle of more than 180°. There is no such limitation for deformable bodies or discrete rigid surfaces.

You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

1. From the arc tools in the Sketcher toolbox, select the Center with Two Endpoints arc tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click at the center of the desired arc.
3. Click the first endpoint to define the radius of the arc.

    **Tip:** If necessary, you can use the text box in the prompt area to enter the precise coordinates of the center and the endpoints of the arc. For more information on precisely defining the arc, see *Specifying precise geometry*.

   Abaqus/CAE draws a circle showing the radius of the arc as you move the cursor from the center of the arc to its first endpoint.
4. Move the cursor clockwise from the first endpoint to draw the arc in a clockwise direction. Move the cursor counterclockwise from the first endpoint to draw the arc in a counterclockwise direction. Click the second endpoint to define the length of the arc. If you start drawing the arc and then decide to change its direction, you must return to the first endpoint and move the cursor in the desired direction toward the second endpoint.

   Step 2 and Step 3 define the center of the arc and the radius of the arc, respectively. The point that you select in Step 4 defines only the length of the arc, and the point may not lie on the arc. If you want the arc to pass through a vertex of the sketch, you should select that vertex when you click the first endpoint in Step 3.

    **Tip:** If necessary, click the **Previous** button  to reverse the selection of the endpoints.
5. To create more arcs, repeat the above steps beginning with Step 2.

6. When you have finished creating arcs, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

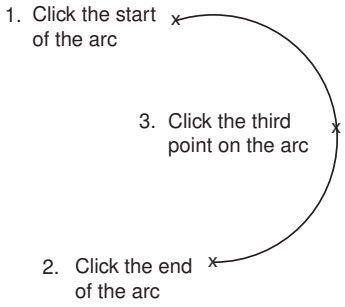
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching arcs through three points

Use the Through Three Points arc tool from the Sketcher toolbox to draw arcs through three noncolinear points.

The following figure shows the resulting arc:



When you draw an arc using the Through Three Points method, you should consider the same angle size and point selection prerequisites as described in *Sketching arcs using a center point and two endpoints*.

1. From the arc tools in the Sketcher toolbox, select the Through Three Points arc tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Click the two points that you want to use as the endpoints of the arc.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of the endpoints. For more information on precisely defining the arc, see *Specifying precise geometry*.

   Abaqus/CAE draws a sample arc with a predetermined radius as you select the second endpoint of the arc. You change this sample radius when you select the third point.

3. Move the cursor to the third point on the arc, and click mouse button 1.

   Abaqus/CAE draws an arc joining the endpoints and passing through the selected point.

   Tip: If necessary, click the **Previous** button  to reverse the selection of either endpoint. Clicking one enables you to reverse the selection of the second endpoint; clicking twice reverses the selection of the first endpoint as well.

4. To create more arcs, repeat the above steps beginning with Step 2.
5. When you have finished creating arcs, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

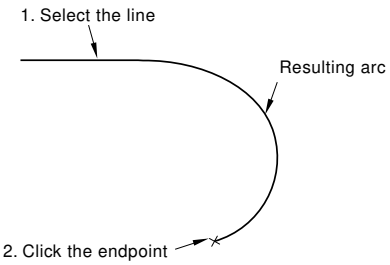
Additional information

- *Sketching simple objects*

- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching arcs tangent to a line

Use the tangent arc tool from the Sketcher toolbox to draw an arc that is tangent to a line or curve at a selected point and ends at a second selected point, as illustrated in the following figure:



You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, including those that define arcs, see *Dragging Sketcher objects*.

1. From the arc tools in the Sketcher toolbox, select the tangent arc tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the point along the line or curve. The arc will be tangent-continuous at this point.

   If you start the arc at the endpoint of a line, a spline, or a second arc, the new arc is tangent-continuous at the endpoint. If you start the arc at a point in space (where there is no endpoint), the arc is tangent to the positive *X*-axis. If more than one line or curve meets at the point, the arc will be tangent to the one that was created first.

3. The radius of the arc changes as you move the cursor. Click at the desired endpoint.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of the tangent point and the endpoint of the arc. For more information on creating the desired arc, see *Specifying precise geometry*.

4. To create more arcs, repeat the above steps beginning with Step 2.
5. When you have finished creating tangent arcs, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

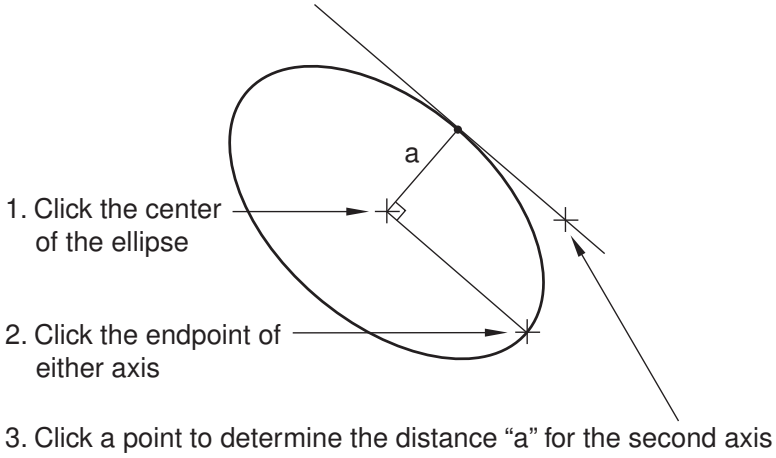
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching ellipses

Use the ellipse tool from the Sketcher toolbox to draw ellipses based on a center point, an axis endpoint, and an arbitrary point whose distance from the first axis determines the length of the second axis.

An example of using the ellipse tool from the Sketcher toolbox is shown here:



You should take care positioning points while sketching because this positioning can affect the quality of your mesh. The center point and the first axis point in the sketch become vertices of the part you are creating. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

You cannot edit the size of an ellipse by moving its vertices. To edit the size of an ellipse, create a radial dimension and then edit that dimension. The radial dimension for an ellipse includes both the major and minor radius.

1. From the Sketcher toolbox, select the ellipse tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click at the desired location of the center of the ellipse.
3. Move the cursor to the endpoint of the first axis of the ellipse.

   Abaqus/CAE displays an ellipse that could be created using the current cursor position to define the major axis and rotation.
4. Click a point to define the endpoint of one axis. The point that you select can be either the major or minor axis endpoint.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of the center and the first axis endpoint of the ellipse. For more information on precisely defining geometry, see *Specifying precise geometry*.

   Abaqus/CAE sets the rotation of the ellipse and the endpoint of the first axis.
5. Click a point to define the length of the second axis. The point you select defines the distance from the first axis to the end of the second axis. The point does not need to lie on the second axis, and it cannot lie along the first axis since this would create a second axis of length 0.

   Abaqus/CAE creates the ellipse.
6. To create more ellipses, repeat the above steps beginning with Step 2.

7. When you have finished creating ellipses, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

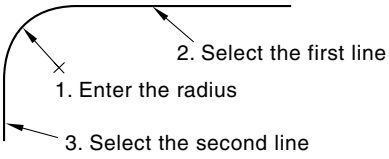
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching fillets between two lines

Use the fillet tool from the Sketcher toolbox to draw fillets between two lines or circles.

Enter the radius of the fillet and select the two lines or circles as shown here:



Construction geometry illustrates how you can create a fillet tangent to two construction circles.

You should take care positioning points while sketching because this positioning can affect the quality of your mesh. Points in the sketch become vertices of the part you are creating or modifying. In turn, when you mesh your model in the Mesh module, Abaqus/CAE converts these vertices into fully constrained seeds and places nodes at their location. For information on how to subsequently move vertices, see *Dragging Sketcher objects*.

If you create a fillet and subsequently move the selected lines or circles, Abaqus/CAE will move the fillet and maintain the tangency.

1. From the Sketcher toolbox, select the fillet tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. In the text box that appears in the prompt area, enter the radius of the desired fillet.
3. Select the two lines or circles (not arcs) to which the fillet must remain tangent.

   The fillet appears between the two lines or circles.

   Tip: When you select a line or circle from the sketch, Abaqus/CAE uses the cursor position to determine the location of the fillet. To create the desired fillet, you should position the cursor close to the expected location of the fillet when making a selection.
4. To create more fillets of the same radius, repeat the previous step.
5. When you have finished creating fillets, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

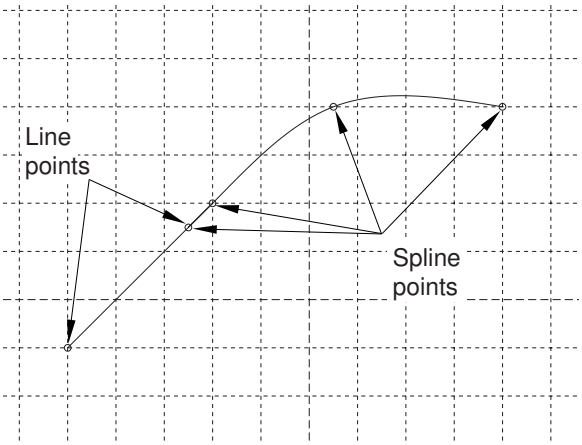
Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Sketching splines

Use the spline tool from the Sketcher toolbox to draw a spline curve connecting a series of points. Abaqus/CAE calculates the shape of the curve using a cubic spline fit between all points along the spline; in addition, the first and second derivatives of the spline are continuous. As you create the spline, you can influence the shape of the curve by placing the vertices closer or farther apart. However, you cannot add or remove vertices from a spline after you have drawn it; you must delete the spline and create a new one with the desired number of vertices. You can use the **Modify** tools to move the original vertices. Only the endpoints of a spline become vertices of the part you are creating or modifying; intermediate points of the spline do not appear outside the Sketcher.

If you want the spline to start tangent to an existing line, at the start of the spline create two adjacent vertices that are colinear with the line. This method is useful for creating smooth rigid surfaces, as illustrated in the following figure:



When you create an analytical *rigid surface*, you sketch a sequence of lines, arcs, and parabolas to define its profile. To create a parabolic curve, sketch a spline defined by only three vertices.

1. From the Sketcher toolbox, select the spline tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. To construct a spline, click each vertex.

   Tip: If necessary, you can use the text box in the prompt area to enter the precise coordinates of each vertex of the spline. For more information on creating the desired spline, see *Specifying precise geometry*.

   The spline appears as you click each vertex or enter each coordinate, and Abaqus/CAE adjusts the curve to maintain a cubic spline between all points.

   Tip: If you make a mistake while constructing a spline, you can click the **Previous** button  to step back to the previous vertex. Alternatively, you can click the **Undo** button in the Sketcher toolbox to delete the entire spline.

3. To complete the spline, click mouse button 2.
4. To create more splines, repeat the above steps beginning with step 2.
5. When you have finished creating splines, do one of the following:

- Click mouse button 2 anywhere in the Abaqus/CAE window.
- Select any other tool in the Sketcher toolbox.
- Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating construction geometry

This section describes each of the Sketcher tools used to create construction geometry. Construction geometry is used to help you create and align objects in your sketch and to define the axis of rotation for revolved solids and surfaces.

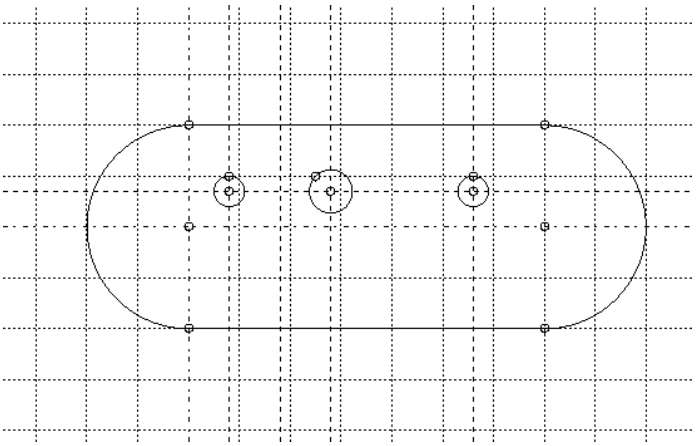
In this section:

- *Creating a horizontal construction line*
- *Creating a vertical construction line*
- *Creating an oblique construction line*
- *Creating angled construction lines*
- *Creating a construction circle*
- *Setting sketch components as construction geometry*

Creating a horizontal construction line

Use the horizontal construction line tool from the Sketcher toolbox to help position and align objects along a horizontal line.

Horizontal construction lines are created parallel to the *X*-axis of the Sketcher grid. Rotating the Sketcher grid does not affect existing horizontal construction lines, but new horizontal construction lines are aligned with the rotated grid. The following figure illustrates how a horizontal construction line and a set of vertical construction lines can be used to align circle centers. (Dashed lines indicate construction geometry.)



You can also use a horizontal construction line to define the axis of rotation for revolved solids and surfaces.

1. From the construction tools in the Sketcher toolbox, select the horizontal construction tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   The horizontal construction line moves vertically as you move the cursor around the Sketcher sheet. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click a point that will lie on the horizontal construction line. Alternatively, you can type its *X*–*Y* coordinates in the text field that appears in the prompt area. (The *X*-coordinate is arbitrary since it is ignored.)
3. To create additional horizontal construction lines, repeat the previous step.
4. When you have finished creating horizontal construction lines, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

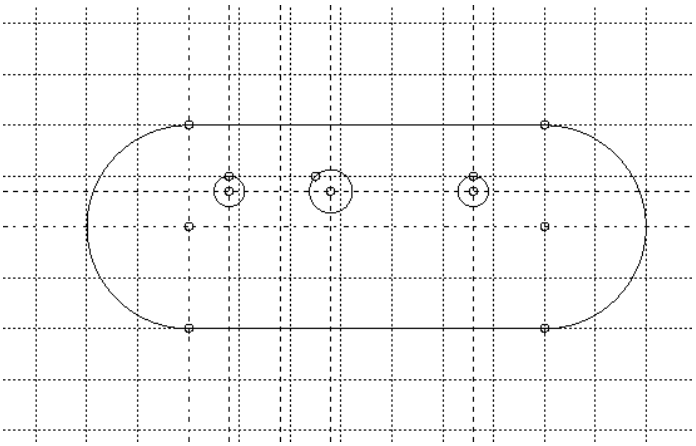
Additional information

- *Construction geometry*
- *Realigning the sketch grid*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating a vertical construction line

Use the vertical construction line tool from the Sketcher toolbox to help position and align objects along a vertical line.

Vertical construction lines are created parallel to the *Y*-axis of the Sketcher grid. Rotating the Sketcher grid does not affect existing vertical construction lines, but new vertical construction lines are aligned with the rotated grid. The following figure illustrates how a horizontal construction line and a set of vertical construction lines can be used to align circle centers. (Dashed lines indicate construction geometry.)



You can also use a vertical construction line to define the axis of rotation for revolved solids and surfaces.

1. From the construction tools in the Sketcher toolbox, select the vertical construction tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   The vertical construction line moves horizontally as you move the cursor around the Sketcher sheet. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click a point that will lie on the vertical construction line. Alternatively, you can type its *X*–*Y* coordinates in the text field that appears in the prompt area. (The *Y*-coordinate is arbitrary since it is ignored.)
3. To create additional vertical construction lines, repeat the previous step.
4. When you have finished creating vertical construction lines, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

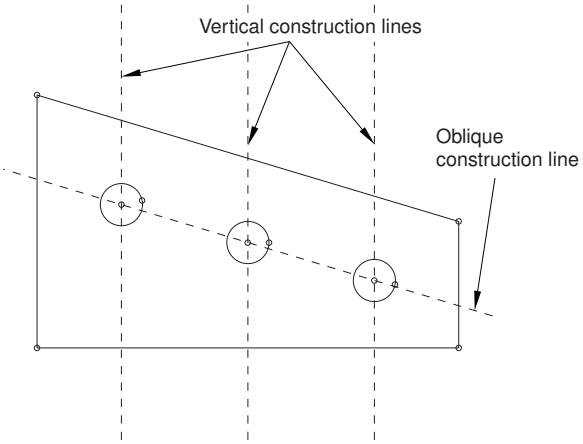
Additional information

- *Construction geometry*
- *Realigning the sketch grid*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating an oblique construction line

Use the oblique construction line tool from the Sketcher toolbox to help position and align objects along an arbritrary oblique line defined by two points.

The following figure illustrates how an oblique construction line and a set of vertical construction lines can be used to align circle centers:



You can also use an oblique construction line to define the axis of rotation for revolved solids and surfaces.

1. From the construction tools in the Sketcher toolbox, select the oblique construction tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click two points that will lie on the oblique construction line. Alternatively, you can type their *X*–*Y* coordinates in the text field that appears in the prompt area.

   The oblique construction line appears when you select the first point. The line rotates about this point until you select the second point.
3. To create additional oblique construction lines, repeat the previous step.
4. When you have finished creating oblique construction lines, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

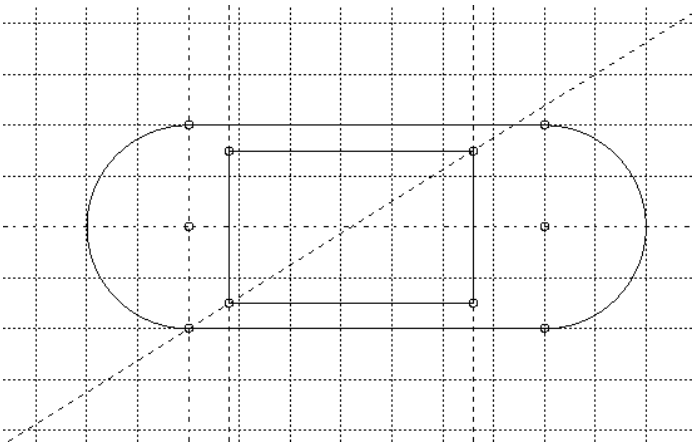
Additional information

- *Construction geometry*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating angled construction lines

Use the angled construction line tool from the Sketcher toolbox to help position and align objects along a line running at a specified angle to the *X*-axis of the Sketcher grid.

Rotating the Sketcher grid does not affect existing angled construction lines, but new angled construction lines are measured from the *X*-axis of the rotated grid. The following figure illustrates how an angled construction line and a pair of vertical construction lines can be used to position the vertices of a rectangle. (Dashed lines indicate construction geometry.)



You can also use an angled construction line to define the axis of rotation for revolved solids and surfaces.

1. From the construction tools in the Sketcher toolbox, select the angled construction line tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. In the text box that appears in the prompt area, enter an angle value in degrees. Positive angles are measured counterclockwise from the horizontal axis, and negative angles are measured clockwise from the horizontal axis.

   The construction line moves as you move the cursor around the Sketcher sheet.
3. Click a point that will lie on the angled construction line. Alternatively, you can type its *X*–*Y* coordinates in the text field that appears in the prompt area.
4. To create additional angled construction lines, repeat the above steps beginning with Step 2.
5. When you have finished creating angled construction lines, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

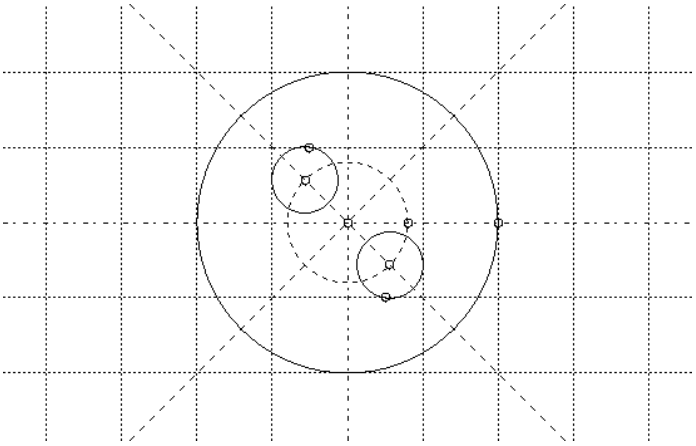
- *Construction geometry*
- *Realigning the sketch grid*
- *Sketching simple objects*
- *Creating construction geometry*

- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating a construction circle

Use the construction circle tool from the Sketcher toolbox to help position and align objects around a circle.

The following figure illustrates how a construction circle and a pair of angled construction lines can be used to position the centers of two circles. (Dashed lines indicate construction geometry.)



1. From the construction tools in the Sketcher toolbox, select the construction circle tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click at the desired location of the center of the construction circle. Alternatively, you can type the *X*–*Y* coordinates of the circle center in the text field that appears in the prompt area.

   The radius of the construction circle changes as you move the cursor around the Sketcher sheet.
3. Click a point that will lie on the circumference of the construction circle, or type its *X*–*Y* coordinates in the prompt area.

   Abaqus/CAE creates the construction circle using the point you specified.
4. To create more construction circles, repeat the above steps beginning with Step 2.
5. When you have finished creating construction circles, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Construction geometry*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Setting sketch components as construction geometry

Use the **Set as Construction** tool from the Sketcher toolbox to convert a component of your sketch into construction geometry. The **Unset Construction** tool reverses this process for items that had been converted to construction geometry. You can unset construction geometry components only if they were converted to construction geometry using the **Set as Construction** tool; items that were added to the sketch as construction geometry cannot be unset.

Additional information

- *Construction geometry*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Set items in a sketch as construction geometry

1. From the construction tools in the Sketcher toolbox, select the **Set as Construction** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select the sketch item or items that you want to set as construction geometry. You can drag-select, **[Shift] + Click**, or **[Ctrl] + Click** to select multiple items.

   Abaqus/CAE indicates the items you select by coloring them red.
3. To complete the conversion, click **Done** in the prompt area or click mouse button 2 anywhere in the Abaqus/CAE window.

   Abaqus/CAE displays the items you selected with dashed lines, indicating that they are now construction geometry.
4. When you have finished selecting geometry to convert to construction geometry, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Unset construction geometry items

1. From the construction tools in the Sketcher toolbox, select the **Unset Construction** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select one or more construction geometry items. You can drag-select, **[Shift] + Click**, or **[Ctrl] + Click** to select multiple items. All construction geometry is displayed in the sketch with dashed lines.

   Abaqus/CAE indicates the items you select by coloring them red.
3. To complete the conversion, click **Done** in the prompt area or click mouse button 2 anywhere in the Abaqus/CAE window.

   Abaqus/CAE displays the items you selected with solid lines, indicating that they are not construction geometry.

4. When you have finished selecting construction geometry items that you want to revert, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Constraining, dimensioning, and parameterizing a sketch

This section describes each of the Sketcher tools used to add constraints, dimensions, and parameters to sketches. You can use these tools to create relationships between individual entities in a sketch or to create a fully constrained and parametric sketch.

In this section:

- *Automatically constraining a sketch*
- *Automatically dimensioning a sketch*
- *Adding individual constraints*
- *Adding individual dimensions*
- *Adding and editing parameters*
- *Creating parametric equations*

Automatically constraining a sketch

Adding constraints to a sketch helps you finalize the shape of a sketched feature. Abaqus/CAE can apply constraints to your sketch automatically. You can use the sketcher options to select the types of constraints that Abaqus/CAE can add to the sketch. (For more information, see *Customizing the use of constraints in the Sketcher*.)

1. From the tools in the Sketcher toolbox, select the auto-constrain tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the entities to be constrained.

    **Note:**

   You can select the entire sketch; Abaqus/CAE will not duplicate any existing constraints.

3. Click **Done** in the prompt area.

   Abaqus/CAE creates the constraints on the sketch. Sketch modification tools such as translate, rotate, scale, and mirror can override constraints.

4. If desired, use the Delete tool to remove any unwanted constraints. For more information, see *Deleting Sketcher objects*.

Additional information

- *Controlling sketch geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Customizing the use of constraints in the Sketcher*

Automatically dimensioning a sketch

Adding dimensions to a sketch helps you finalize the size and shape of a sketched feature. Abaqus/CAE can apply dimensions to your sketch automatically. You can use the sketcher options to select the types of dimensions that Abaqus/CAE can add to the sketch. (For more information, see *Customizing the format and use of dimensions in the Sketcher*.)

1. From the tools in the Sketcher toolbox, select the auto-dimension tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select the entities to be dimensioned.

   Note:

   You can select the entire sketch; Abaqus/CAE will not duplicate any existing dimensions.
3. Toggle on **Auto-Constrain** if you also want Abaqus/CAE to create constraints automatically for the selected entities.
4. Click **Done** in the prompt area.

   Abaqus/CAE creates the dimensions—and constraints, if selected—on the sketch.
5. If desired, use the drag tool to move dimensions that appear within the sketched feature and use the delete tool to remove any unwanted dimensions or constraints. For more information, see *Dragging Sketcher objects*, and *Deleting Sketcher objects*, respectively.

Additional information

- *Controlling sketch geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Customizing the format and use of dimensions in the Sketcher*
- *Customizing the use of constraints in the Sketcher*

Adding individual constraints

A constraint is a logical relationship between entities in a sketch. Constraints control Sketcher geometry by removing degrees of freedom from the sketch. The various constraint types are described in *Using constraints to control sketch geometry*. To change constrained geometry, you must either work within the defined constraints or modify the constraints to accept your changes. You can also constrain a sketch automatically using the **Auto-Constrain** tool. For more information, see *Automatically constraining a sketch*.

1. From the tools in the Sketcher toolbox, select the constraint tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE opens the **Add Constraint** dialog box.
2. Select the type of constraint that you want to add to the sketch, and click **Apply**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. Select the entity or entities to create the constraint.

   Abaqus/CAE creates the constraint and displays the appropriate constraint symbol on the sketch. Sketch modification tools such as translate, rotate, scale, and mirror can override constraints.
4. To create additional constraints, repeat the above steps beginning with Step 2.
5. When you have finished creating constraints, click **Cancel** to close the **Add Constraint** dialog box.

Additional information

- *Using constraints to control sketch geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Customizing the use of constraints in the Sketcher*

Adding individual dimensions

Dimensions are a specialized form of constraints; where constraints create a logical relationship between Sketcher entities, dimensions create a numerical relationship. Dimensions add clarity to your sketches and allow you to position objects precisely. Use the dimension tool from the Sketcher toolbox to add individual dimensions to a sketch. You can also dimension a sketch automatically using the **Auto-Dimension** tool. For more information, see *Automatically dimensioning a sketch*.

The dimension tool is interactive; Abaqus/CAE uses preselection and the Sketcher entities that you select to display a preview of the dimension that will be created. If the dimension type shown is not the type that you want to create, move the cursor to another location. Once you accept the dimension type and position, you can enter a new dimension value.



Note:

The interactive function of the dimension tool is available only if preselection is active. Preselection is toggled on by default in the **Sketcher Options** dialog box. (For more information , see *The Sketcher cursors and preselection*.)

1. From the tools in the Sketcher toolbox, select the dimension tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the first entity to dimension.

   Abaqus/CAE prompts you according to your selection:

Vertex

Abaqus/CAE prompts you to select another vertex or line for the dimension. Select a line to dimension the distance between the vertex and the line. Select a vertex to dimension the horizontal, vertical, or oblique distance between the selected points—the type of dimension Abaqus/CAE creates is based on the position of the second vertex and the cursor position.

Abaqus/CAE creates a preview of the dimension that will be created and its location based on the cursor position. Horizontal or vertical dimensions values between two vertices are preceded by an H or V, respectively.

Line

Abaqus/CAE prompts you to select a location for the default length dimension. To create a different type of dimension, move the cursor over other entities in the sketch; Abaqus/CAE creates a preview of the appropriate dimension type for the relation between the selected line and the preselected entity. Click on the desired entity to accept the displayed type of dimension.

Circle or circular arc

Abaqus/CAE prompts you to select a location for the radial dimension. No other entity selections are available.

3. Move the cursor to the location where you would like the dimension to appear; click when you are satisfied with the appearance of the dimension.
4. If desired, edit the dimension value in the prompt area; press `Enter` or click mouse button 2 to accept the dimension value.

Abaqus/CAE creates the dimension on the sketch. If necessary, vertices are moved to comply with the edited dimension value and any other constraints in the sketch.

5. To create additional dimensions, repeat the above steps beginning with Step 2.
6. When you have finished creating dimensions, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Controlling sketch geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Editing dimensions*
- *Specifying precise geometry*
- *Modifying objects by changing dimensions or adding parameters*
- *Customizing the format and use of dimensions in the Sketcher*

Adding and editing parameters

Parameters are constraints that have variable names. They can have either numerical values or values based on the result of mathematical expressions involving other parameters—parametric equations. Parameters can be associated with dimensions in the current sketch, or they can be numerical constants and expressions independent of the sketch. The value of parameters associated with dimensions directly affect the sketch, whereas independent parameters affect the sketch only if you use them to create expressions for parameters that are in the sketch.

You add and edit parameters using the **Parameter Manager**. You can use the **Dimension** button to create new parameters by selecting dimensions, or you can add independent parameters by typing them in blank lines in the manager.

1. From the tools in the Sketcher toolbox, select the **Parameter Manager** tool $f(x)$. For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   The **Parameter Manager** appears.

   Tip: You can also open the **Parameter Manager** by selecting **Edit->Parameter Manager** from the main menu bar or by using the **f(x)** button in the **Edit Dimension** dialog box.

2. Click **Dimension**, and select dimensions from the sketch to associate them with new parameters.

   Abaqus/CAE adds the new parameters to the **Parameter Manager** using default names of the form `dimension_x`.

3. Click on a cell in the **Name** column to edit an existing name, or add a new parameter name by editing a blank cell.

   If the selected parameter is associated with a dimension, Abaqus/CAE highlights the parameter in the sketch.

4. Click on a cell in the **Expression** column to edit it directly in the cell or to select it for editing with the **Expression Builder**.

   You must define all the parameters that you want to use in an expression prior to defining the expression.

   Tip: Use the buttons on the right side of the manager to insert, rearrange, or delete rows.

5. If desired, use the **Expression Builder** to edit the expression. For detailed instructions, see *Creating parametric equations*.
6. Click **Apply** in the **Parameter Manager** to implement your changes in the sketch.

   Abaqus/CAE updates the sketch with any new expression values and, if **Show parameter names** is toggled on in the **Sketcher Options**, displays the revised parameter names.

Additional information

- *Specifying precise geometry*
- *Modifying objects by changing dimensions or adding parameters*
- *Using parameters to control sketch geometry*
- *Editing dimensions*
- *Customizing the format and use of dimensions in the Sketcher*

Creating parametric equations

Parametric equations are mathematical expressions relating different quantities in a sketch. The **Expression Builder** helps you create parametric equations using the parameters in your sketch. For example, if you want the width of a Sketcher object to be twice its length, you can *associate the dimensions* with parameters and use the **Expression Builder** to create the parametric equation $width = 2 * length$.

You can use the following operators to create parametric equations:

Mathematical operations:

+	Add.
–	Subtract.
*	Multiply.
/	Divide.
1/A	Divide 1 by the parameter, value, or expression.
abs(A)	Take the absolute value of the parameter, value, or expression.
sqrt(A)	Take the square root of the parameter, value, or expression.

Trigonometric operations:

cos(A)	Take the cosine of the parameter, value, or expression.
acos(A)	Take the arccosine of the parameter, value, or expression.
cosh(A)	Take the hyperbolic cosine of the parameter, value, or expression.
sin(A)	Take the sine of the parameter, value, or expression.
asin(A)	Take the arcsine of the parameter, value, or expression.
sinh(A)	Take the hyperbolic sine of the parameter, value, or expression.
tan(A)	Take the tangent of the parameter, value, or expression.
atan(A)	Take the arctangent of the parameter, value, or expression.
tanh(A)	Take the hyperbolic tangent of the parameter, value, or expression.

Logarithmic and exponential operations:

exp(A)	Take the exponential of the parameter, value, or expression.
log(A)	Take the natural log of the parameter, value, or expression.
log10(A)	Take the base 10 log of the parameter, value, or expression.

1. With an expression highlighted in the **Parameter Manager**, click **Expression Builder**. Abaqus/CAE opens the **Expression Builder** dialog box.
2. From the **Operators** list, select the desired operation. The operator appears within the expression window.

3. From the **Parameter Name** choices, click the name of the parameter on which to operate and click **Add to Expression**. You can choose from all parameters that appear above the parameter that you are defining in the **Parameter Manager**.

   The parameter name appears within the expression window.

4. Repeat Steps 2 and 3 as needed to complete the expression. In some cases you may need to reposition the cursor to correctly place the next portion of the expression.
5. When you are finished, click **OK** to save your changes and to close the dialog box.

   Abaqus/CAE updates the expression and the current value in the **Parameter Manager**.

Additional information

- *Adding and editing parameters*
- *Editing dimensions*
- *Customizing the format and use of dimensions in the Sketcher*

Editing dimensions

Dimensions in Abaqus/CAE are used to precisely control sizes and distances, to annotate reference values, and to create parameters that are, in turn, used to define equations relating different pieces of geometry. You can edit any dimension in a sketch as follows:

- Change the dimension value. Changing a dimension value modifies the associated Sketcher geometry.
- Designate the dimension as a reference dimension. Reference dimensions indicate quantities whose values are controlled elsewhere in the same sketch or dimensions of reference geometry that are controlled in another sketch.
- Associate the dimension with a parameter. Parameters control Sketcher geometry by defining relationships between different pieces of geometry, including reference geometry.

The effects of modifying a dimension depend on how the dimension is used in the sketch. For example, if a radius has an equal radius constraint, editing the radius dimension will change all of the constrained radii. Likewise, if a dimension is associated with a parameter, editing its value will change the value of all the parameters that refer to it. For more information on constraints, dimensions, and parameters, see *Controlling sketch geometry*.

1. From the tools in the Sketcher toolbox, select the dimension modification tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select the dimension you want to change.

   Abaqus/CAE highlights the selected dimension and opens the **Edit Dimension** dialog box.
3. Enter a new value for the dimension.

   Note:

   You cannot edit the value field for reference dimensions or for parameters whose values are defined using an equation.
4. Toggle **Reference** to switch between a standard dimension whose value is used to control the sketch geometry and a reference dimension that reads the current value from the geometry.
5. Click **f(x)** to associate the dimension with a new parameter or to edit a previously associated parameter.

   Abaqus/CAE opens the **Parameter Manager** for editing. For more information, see *Adding and editing parameters*.
6. When you are finished, click **Apply** in the **Edit Dimension** dialog box to update the sketch.

   Abaqus/CAE updates the dimension and geometry as required. If your edits result in an overconstraint, the conflicting constraints, dimensions, and parameters are indicated in magenta (for more information, see *Fully constrained geometry*).
7. To edit more dimensions, repeat the above steps beginning with Step 2.
8. When you have finished editing dimensions, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Specifying precise geometry*
- *Controlling sketch geometry*
- *Constraining, dimensioning, and parameterizing a sketch*
- *Moving or copying sketch geometry*

Adding reference geometry

When you sketch the profile of a feature, Abaqus/CAE projects onto the sketch sheet the part edges and vertices that are in the same plane as the sketch plane. In addition, Abaqus/CAE projects all datum axes and datum points onto the sketch sheet. These projected lines and points are called reference geometry, and you can use them for reference while sketching. If you want to use other edges and vertices, including orphan element edges and orphan nodes, you must project them onto the sketch plane as reference geometry by selecting **Add->References** from the main menu bar. Alternatively, if you want to create new sketch edges from existing model edges, you can select **Add->Edges** from the main menu bar (for more information, see *Projecting edges onto a sketch*).

1. From the Sketcher toolbox, select the **Project References** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select the edges, vertices, orphan element edges, and orphan nodes you want to project onto the sketch plane as reference geometry.
3. Click mouse button 2 when you have finished making selections.

   Abaqus/CAE creates isolated points for the projected reference vertices and nodes and creates reference edges for the geometric and orphan element edges.

Additional information

- *Reference geometry*
- *Projecting edges onto a sketch*

Projecting edges onto a sketch

When you sketch the profile of a feature, you can create new edges by projecting existing edges from the part onto the sketch sheet. You can use projected edges to duplicate the shapes of existing features as they appear from the current sketch plane. To project part edges, including orphan element edges, onto your sketch; select **Add->Edges** from the main menu bar.

1. From the Sketcher toolbox, select the **Project Edges** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. If desired, toggle off **Constrain to background** in the prompt area to project the edges as independent objects.

   If projected edges are constrained to the background, their shape and position depend on the original feature edges. If projected edges are unconstrained, changes to the original feature do not affect them—you can edit the projected edges and vertices independently from the original feature.
3. Select the edges you want to project into the sketch plane.
4. Click mouse button 2 to indicate you have finished selecting edges.

   Abaqus/CAE creates the new edges in the sketch.

Additional information

- *Reference geometry*
- *Adding reference geometry*

Moving or copying sketch geometry

This section describes how to use the translate, rotate, mirror, and scale tools to move or copy Sketcher geometry.

In this section:

- *Translating Sketcher objects along a vector*
- *Rotating Sketcher objects about a point*
- *Enlarging or reducing Sketcher objects*
- *Moving or copying Sketcher objects about a mirror line*

Translating Sketcher objects along a vector

Use the translate tool from the Sketcher toolbox to move or copy Sketcher objects—lines, arcs, circles, ellipses, fillets, or splines—along a specified vector. You can define the beginning and the end of the translation vector by selecting from the sketch or by typing the *X*- and *Y*-coordinates of each end.

1. From the move and copy tools in the Sketcher toolbox, select the translate tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.
2. Select **Copy** or **Move** in the prompt area.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. Select all the objects that you want to translate. You can select sketch and construction geometry to move, or you can select sketch, construction, and reference geometry to copy; dimensions cannot be translated.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."
4. Click mouse button 2 to indicate that you have finished selecting objects.
5. Select a start position for the translation vector, or enter coordinates in the prompt area.
6. Select an end position for the translation vector, or enter coordinates in the prompt area.

   Abaqus/CAE moves or copies the selected objects as specified.
7. If desired, repeat Steps 5 and 6 to continue moving the previously selected objects.

   Regardless of whether you originally moved or copied the selected objects, subsequent translation vectors only move the objects from their position at the end of the previous step.
8. When you have finished translating objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Rotating Sketcher objects about a point

Use the rotate tool from the Sketcher toolbox to rotate Sketcher objects—lines, arcs, circles, ellipses, fillets, or splines—through a specified angle about a specified point. You can choose to move the original objects or create copies and rotate them to the new location.

1. From the move and copy tools in the Sketcher toolbox, select the rotate tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.
2. Select **Copy** or **Move** in the prompt area.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. Select all the objects that you want to rotate. You can select sketch and construction geometry to move, or you can select sketch, construction, and reference geometry to copy; dimensions cannot be rotated.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."
4. Click mouse button 2 to indicate that you have finished selecting objects.
5. Select a point for the center of rotation, or enter coordinates in the prompt area.
6. Enter the angle of rotation in the prompt area.

   Abaqus/CAE moves or copies the objects as specified.
7. If desired, repeat Steps 5 and 6 to continue moving the previously selected objects.

   Regardless of whether you originally moved or copied the selected objects, subsequent rotations only move the objects from their position at the end of the previous step.
8. When you have finished rotating objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Enlarging or reducing Sketcher objects

Use the scale tool from the Sketcher toolbox to enlarge or reduce Sketcher objects—lines, arcs, circles, ellipses, fillets, or splines—about a specified point. You can choose to move the original objects or to create a scaled copy in the new location. If you choose to move objects, your changes are controlled by any constraints in the sketch and by the constraint solution method in the sketch options (for more information, see *Customizing the use of constraints in the Sketcher*). If you choose to create a copy, the movement of the copy is unrestricted.

1. From the move and copy tools in the Sketcher toolbox, select the scale tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.
2. Select **Copy** or **Move** in the prompt area.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. Select all the objects that you want to scale. You can select sketch and construction geometry to move, or you can select sketch, construction, and reference geometry to copy; dimensions cannot be scaled.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."
4. Click mouse button 2 to indicate that you have finished selecting objects.
5. Select a center point about which to scale, or enter coordinates in the prompt area.
6. Enter the scale factor—values greater than `1.0` enlarge the moved or copied objects, values less than `1.0` reduce them.

   Abaqus/CAE keeps the selected point stationary in the viewport and enlarges or reduces the objects around it according to the scale factor.
7. If desired, repeat Steps 3 through 6 to continue scaling objects.

   Note:

   Subsequent scale operations are of the same type—**Copy** or **Move**—as you originally selected. To change the type, end the procedure and select the scale tool again.
8. When you have finished scaling objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Moving or copying Sketcher objects about a mirror line

Use the mirror tool from the Sketcher toolbox to mirror Sketcher objects—lines, arcs, circles, ellipses, fillets, or splines—about a specified line. You can choose to move the original objects or to create a mirror copy in the new location. If you choose to move objects, your changes are controlled by any constraints in the sketch and by the constraint solution method in the sketch options (for more information, see *Customizing the use of constraints in the Sketcher*). If you choose to create a copy, Abaqus/CAE applies a mirror constraint between the original objects and the copies so that they will remain mirror images during any subsequent sketch edits.

1. From the move and copy tools in the Sketcher toolbox, select the mirror tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.
2. Select **Copy** or **Move** in the prompt area.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
3. Select an existing straight line in the sketch as the mirror line. You can select any straight edge from sketch, construction, or reference geometry.
4. Select all the objects that you want to mirror. You can select sketch and construction geometry to move, or you can select sketch, construction, and reference geometry to copy; dimensions cannot be mirrored.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."
5. Click mouse button 2 to indicate that you have finished selecting objects.

   Abaqus/CAE moves or copies the objects as specified.
6. If desired, repeat Steps 4 and 5 to move or copy more objects about the same mirror line.
7. When you have finished moving and copying objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Modifying objects

This section describes how to use the Sketcher tools to move and resize Sketcher objects.

In this section:

- *Dragging Sketcher objects*
- *Modifying Sketcher objects by trimming or extending edges*
- *Modifying Sketcher objects by automatically trimming edges*
- *Modifying Sketcher objects by splitting edges*
- *Modifying Sketcher objects by merging edges*

Dragging Sketcher objects

Use the **Drag** tool from the Sketcher toolbox to quickly move or resize a Sketcher object or to reposition dimension text. You can drag only one vertex, line, or dimension at a time. The object or dimension text that you drag snaps to the grilines when the **Snap to grid** option is selected; see *Turning snapping on or off* for more information.

If you drag a vertex or line, Abaqus/CAE modifies connected Sketcher objects according to the applied constraints—including dimensions and parameters—and according to the constraint solution method applied to drag operations. By default, the **Minimum move** constraint solution method, in which Abaqus/CAE moves the minimum number of entities to satisfy the constraints, is used (for more information, see *Customizing the use of constraints in the Sketcher*).

1. From the modify tools in the Sketcher toolbox, select the **Drag** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Handles appear on all vertices in the viewport. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Click and drag the selected vertex, edge, or dimension text to a new location.

   The selected object moves with the cursor unless constraints in the sketch do not allow the desired movement.
3. When you have finished dragging objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Modifying, copying, and offsetting objects*
- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Modifying Sketcher objects by trimming or extending edges

Use the trim and extend tool from the Sketcher toolbox to trim or extend individual edges of Sketcher objects. You can trim or extend an edge at any intersection between the edge and other sketch, reference, or construction geometry. You can also trim or extend an edge at an extended intersection—an intersection that would exist only if another edge in the sketch was extended. You cannot trim or extend a fillet; if you trim or extend an edge that defines an endpoint of a fillet, Abaqus/CAE deletes the fillet.

1. From the modify tools in the Sketcher toolbox, select the trim and extend tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select an edge near the portion that you want to keep.

   You can select any sketched or imported edge except for circles and ellipses. Circles and ellipses have only one edge, they cannot be extended or trimmed unless you first make them into one or more arcs.

   Tip: Use the Auto-trim tool to remove part of a circle or an ellipse, or use the split tool to split it into arcs. For more information, see *Modifying Sketcher objects by automatically trimming edges*, and *Modifying Sketcher objects by splitting edges*, respectively.

   You cannot extend spline curves. If you trim or split a spline curve, Abaqus/CAE removes the control points to retain the exact shape of the original curve; therefore, you can no longer edit the shape of the curve.

3. Move the cursor over other edges in the sketch.

   As you move the cursor, Abaqus/CAE highlights the edge that would be selected at the current cursor location. Abaqus/CAE also creates a preview of the extended or trimmed portion of the first edge by highlighting it a different color from the portion that will remain unchanged.

4. Click to select the highlighted edge.

   Abaqus/CAE trims or extends the first edge to meet the intersection point (or potential intersection point) with the second edge. Abaqus/CAE also splits the second edge if the intersection point is between its endpoints.

5. To trim or extend more edges, repeat the above steps beginning with Step 2.
6. When you have finished trimming and extending edges, do one of the following:

   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Modifying Sketcher objects by automatically trimming edges

Use the Auto-trim tool from the Sketcher toolbox to delete unwanted portions of sketched edges. Edges are deleted between the two “trim points” closest to the cursor. The trim points include:

- Intersections with other edges, construction geometry, or reference geometry
- Endpoints
- Points used to define the circumference of circles

If the endpoints are the only trim points on an edge, Abaqus/CAE deletes the entire edge.

You cannot trim a fillet; if you trim an edge that defines an endpoint of a fillet, Abaqus/CAE deletes the fillet.

1. From the modify tools in the Sketcher toolbox, select the Auto-trim tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Move the cursor over the edges in the sketch.

   As you move the cursor, Abaqus/CAE highlights the section that would be removed.

3. Click to remove the highlighted edge segment.

   Unlike when you use **Trim/Extend**, Abaqus/CAE does not split the intersecting edges.

   Note:

   If you trim a spline curve, Abaqus/CAE removes the control points to retain the exact shape of the original curve; therefore, you can no longer edit the shape of the curve.

4. When you have finished trimming edges, do one of the following

   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Modifying Sketcher objects by splitting edges

Use the split tool from the Sketcher toolbox to split an individual edge of a Sketcher object into multiple edges. You can split an edge at any intersection between the edge and other sketch, reference, or construction geometry. If you use other sketch geometry to split an edge, Abaqus/CAE splits both edges at the intersection point. You can also split an edge at an extended intersection—an intersection that would exist only if another edge in the sketch was extended.

1. From the modify tools in the Sketcher toolbox, select the split tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the edge that you want to split; you can select any sketched or imported edge in Abaqus/CAE.

   Note:

   If you split a spline curve, Abaqus/CAE removes the control points to retain the exact shape of the original curve; therefore, you can no longer edit the shape of the curve.

3. Move the cursor over other edges in the sketch.

   As you move the cursor, Abaqus/CAE highlights the edge that would be selected at the current cursor location. Abaqus/CAE also circles the split point—the intersection point between the edge to be broken and the edge at the cursor position. If there is more than one possible split point between the selected edges, Abaqus/CAE circles the point that is closest to the cursor position.

4. Click to accept the indicated split point.

   Abaqus/CAE splits the first edge, creating two edges that share the split point as one endpoint. Abaqus/CAE also splits the second edge if it is sketch geometry and the split point lies between its endpoints.

   Note:

   If you split a circle or an ellipse, Abaqus/CAE uses the vertex on the edge as a second split point to create two arcs. If the split point indicated in the previous step is the same as the edge point, a warning appears in the message area and the curve is not broken.

5. To split more edges, repeat the above steps beginning with Step 2.
6. When you have finished splitting edges, do one of the following
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Modifying Sketcher objects by merging edges

Small gaps can occur in Sketcher objects when you import sketch geometry or when you use the Project Edges tool on edges that are not in a plane parallel to the sketch plane. The **Merge** tool from the Sketcher toolbox closes these small gaps by moving one or more vertices near the gap to a point on the other side.

Note:

When you merge edges, Abaqus/CAE can close the gap by moving one vertex to coincide with the other or by moving both vertices to a location between them. If you want to control the changes to your sketch, you can constrain the line or point that you want to retain its current location before performing the merge operation.

Abaqus/CAE enables you to choose the vertex that you want to remain stationary, and the merge operation moves any vertices within a specified tolerance radius across the gap to share the same point as your selected vertex. *Figure 1* shows a simple merge operation, closing the upper gap between two sketcher objects.

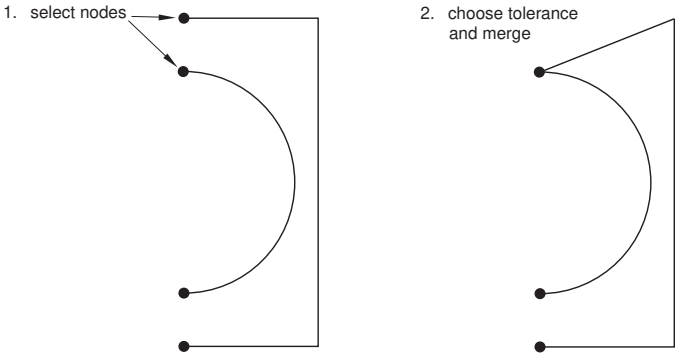


Figure 1: Merging edges

Note:

The Merge tool is intended only for closing small gaps. Merging edges to close larger gaps may change the sketch geometry significantly. If you want to move elements in a sketch over a large distance, use the drag tool to move vertices in your sketch to a new location. For more information, see *Dragging Sketcher objects*.

1. From the modify tools in the Sketcher toolbox, select the Merge tool. For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select at least two vertices near the small gap.
3. Click **Done** in the prompt area.

   Abaqus/CAE prompts you for a **Tolerance** number for this merge operation.
4. Enter a value for the tolerance, or accept the default value of 0.001. Then either press Enter, or click mouse button 2.

   Abaqus/CAE allows you to increase the tolerance, but you should avoid merges that change the sketch geometry significantly.

   If any vertices are within the selected tolerance, Abaqus/CAE merges them to the location of the vertex you selected in Step 2.
5. To merge more edges, repeat the above steps beginning with Step 2.
6. When you have finished merging edges, do one of the following:

- Click mouse button 2 anywhere in the Abaqus/CAE window.
- Select any tool in the Sketcher toolbox.
- Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Repairing short edges, gaps, and overlaps

This section describes how to use the Sketcher tools to repair short edges in a sketch and to remove gaps and overlaps from a sketch.
These tools are intended for the repair of imported sketches, but you can use them to repair geometry in a native sketch as well. You should remove the gaps and overlaps in your sketch before repairing its short edges.

In this section:

- *Removing gaps and overlaps*
- *Repairing short edges*

Removing gaps and overlaps

Use the **Remove gaps and overlaps** tool to repair imported sketches that include gaps or overlaps.

Gaps or overlaps can occur when you import a sketch into Abaqus/CAE from a CAD system that measures sketch geometry using different tolerance values than Abaqus/CAE. The sketch in *Figure 1* includes a gap in the upper left and a small overlap at the midpoint of the top line.

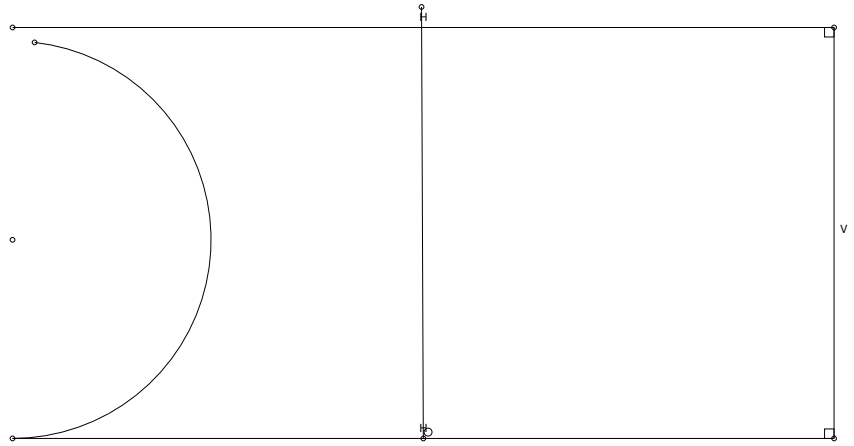


Figure 1: Simple imported sketch with a gap and an overlap.

You can repair the gap and overlap in this example by selecting the edges and specifying a tolerance value. When the size of the gap or the length of the overlap is less than the specified tolerance, Abaqus/CAE merges the vertices to close the gap.

Note:

During the removal of a gap or overlap, Abaqus/CAE can merge vertices by moving one vertex to coincide with the other or by moving both vertices to a location between them. If you want to control the changes to your sketch, you can constrain the line or point that you want to retain its current location before performing the gap/overlap removal.

In practice, you should remove gaps or overlaps from your sketch using an iterative process. Start by selecting the edges adjacent to the gaps or overlaps you want to remove and use the default length tolerance value of 0.001 for your first attempt. If all of the gaps or overlaps are not closed, reselect the edges and increase the tolerance. Repeat this process until Abaqus/CAE merges the selected vertices to close the gaps or overlaps.

1. From the modify tools in the Sketcher toolbox, select the **Remove gaps and overlaps** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the edges between which you want to remove gaps and overlaps. You can **[Shift] + Click** or **[Ctrl] + Click** to specify edges individually, or you can drag-select to specify all of the edges within an area in your sketch.

   Abaqus/CAE colors the selected edges red.

3. Click mouse button 2 anywhere in the Abaqus/CAE window.

   Abaqus/CAE prompts you to select a tolerance value.

4. For your first attempt, accept the default tolerance value and click mouse button 2 anywhere in the Abaqus/CAE window. Abaqus/CAE removes the gaps and overlaps that are smaller than the default tolerance.
5. If you still have gaps and overlaps remaining that you want to remove, perform the following steps:
   a. Reselect the edges between which you want to remove gaps and overlaps.
   b. Increase the tolerance value in the prompt area.
   c. Click mouse button 2 anywhere in the Abaqus/CAE window.

      Abaqus/CAE removes the selected gaps or overlaps that are smaller than the default tolerance.

   d. Continue selecting edges and increasing the tolerance until Abaqus/CAE has removed all the gaps and overlaps that you want to remove from your model.
6. When you have finished removing edges and overlaps, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Modifying objects*

Repairing short edges

Use the **Repair short edges** tool from the Sketcher toolbox to remove selected short edges from your sketch. Abaqus/CAE provides two methods for edge removal: you can either remove every edge that you select, or you can select multiple edges as candidates for removal and specify a length tolerance value. Edges shorter than the tolerance length are removed from the sketch.

If you remove short edges by specifying length tolerance values, you should perform the removals using an iterative process. Start by selecting the edges that you want to remove and use the default length tolerance value of 0.001 for your first attempt. If the short edges are not removed, reselect them and increase the tolerance. Repeat this process until the short edges are removed from the sketch.

You should repair the short edges in your sketch only after removing its gaps and overlaps. For more information, see *Removing gaps and overlaps*.

1. From the modify tools in the Sketcher toolbox, select the **Repair short edges** tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Specify the edges that you want to remove or, if you are specifying a length tolerance value, the edges you want to consider for removal. You can **[Shift] + Click** or **[Ctrl] + Click** to specify edges individually, or you can drag-select to specify all of the edges within an area in your sketch.

   Abaqus/CAE colors the selected edges red.

3. If you are removing short edges without specifying a tolerance, click **Done** in the prompt area.

   Abaqus/CAE removes the selected edges from the sketch.

4. If you are removing edges shorter than a specified length tolerance value, toggle on **Specify tolerance** in the prompt area and click **Done**. For your first attempt, accept the default tolerance value and click **Done** in the prompt area.

   Abaqus/CAE removes from the sketch any of the selected edges that are shorter than the length tolerance value.

5. To continue the removal of short edges by length tolerance value, perform the following steps:
   a. Reselect the edges that you want to consider for removal, and click **Done**.
   b. Leave **Specify tolerance** toggled on, and click **Done** again.
   c. Increase the length tolerance value in the prompt area.
   d. Click mouse button 2 anywhere in the Abaqus/CAE window.

      Abaqus/CAE removes from the sketch any of the selected edges that are shorter than the length tolerance value.

   e. Continue selecting edges and increasing the tolerance until Abaqus/CAE has removed all the short edges that you want to remove from your model.

6. When you have finished repairing short edges in your sketch, do one of the following:

   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any other tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Modifying, copying, and offsetting objects*

- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*

Creating patterns, offsetting, and deleting objects

This section describes how to use the Sketcher tools to copy and delete Sketcher objects.

In this section:

- *Creating linear patterns of objects*
- *Creating radial patterns of objects*
- *Offsetting the edges of Sketcher objects*
- *Deleting Sketcher objects*

Creating linear patterns of objects

Use the linear pattern tool from the Sketcher toolbox to create a linear pattern of selected Sketcher objects. You can create a pattern that extends in one direction (for example, horizontally or vertically), or you can create a pattern that extends in two directions (for example, both horizontally and vertically). You can specify the following:

- The number of copies to create in each direction, including the selected objects. You can create a maxium of 1000 copies.
- The spacing between each copy along the specified direction.
- A line that defines the direction along which Abaqus/CAE generates the copies.

By default, Abaqus/CAE creates three copies in the positive *X*-direction and two copies in the positive *Y*-direction. The default spacing is 10% of the sketch sheet size. If you change the default settings, Abaqus/CAE retains those values while you work on the sketch. However, after you exit the Sketcher, Abaqus/CAE reverts to the original defaults. You cannot edit a pattern after you create it.

1. From the Sketcher toolbox, select the linear pattern tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the objects that you want to copy. Both sketch and *construction geometry* can be copied; *reference geometry* and dimensions cannot be copied.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."

3. Click mouse button 2 to indicate that you have finished selecting objects.

   Abaqus/CAE displays the **Create Linear Pattern** dialog box.

4. From the **Create Linear Pattern** dialog box, configure the pattern in **Direction-1** (by default, **Direction-1** is the *X*-direction in the current grid rotation):

   a. Click the arrows to the right of **Number** to increase or decrease the number of copies to create, including the selected objects. The number of copies in the Sketch updates when you click the arrows and provides a preview of the setting.

      Alternatively, you can type in a number and press **[Enter]** to preview the setting. You can enter any number between 1 and 1000. If you enter a value of 1, Abaqus/CAE displays only the selected objects and does not create any copies of the selected objects; in effect, you are disabling copies in **Direction-1**.

   b. Enter the **Spacing** between each copy along the specified direction.

   c. By default, Abaqus/CAE creates the copies in a horizontal direction. If you want to change the direction in which Abaqus/CAE creates the copies, click **Angle** and select a line from the Sketch to define the new direction. You must pick a straight edge or a straight construction line. Abaqus/CAE calculates the spacing between each copy along the direction that you specify.

   d. By default, Abaqus/CAE creates the copies in the positive *X*-direction. Click **Flip** to reverse the direction in which Abaqus/CAE creates the copies.

5. To create copies in a second direction, enter a **Number** between than 1 and 1000 and specify the **Spacing**, the **Angle**, and the **Flip** direction for **Direction-2** (by default, **Direction-2** is the *Y*-direction in the current grid rotation). You must enter a **Number** greater than 1 for at least one direction.

6. In most cases you will want to preview the linear pattern that Abaqus/CAE will create as you enter values in the **Create Linear Pattern** dialog box. However, if you choose to create a large number of copies, the preview capability may impact the performance of the Sketcher. In this case you should toggle off the **Preview** button.
7. To copy more objects, repeat the above steps beginning with Step 2.
8. When you have finished copying objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Copying sketch objects to create patterns*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Creating radial patterns of objects

Use the radial pattern tool from the Sketcher toolbox to create a radial pattern of selected Sketcher objects. You can specify the following:

- The number of copies to create in the radial pattern, including the selected objects. You can create a maxium of 1000 copies.
- The total angle between the original objects and the last copy in the pattern.
- The position of the center point of the circular pattern.

By default, Abaqus/CAE creates five copies around a complete circle (360°), and the center of the circle is located at the origin of the sketch. If you change the default settings, Abaqus/CAE retains those values while you work on the sketch. However, after you exit the Sketcher, Abaqus/CAE reverts to the original defaults. You cannot edit a pattern after you create it.

1. From the Sketcher toolbox, select the radial pattern tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select the objects that you want to copy. Both sketch and *construction geometry* can be copied; *reference geometry* and dimensions cannot be copied.

   Tip: To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."

3. Click mouse button 2 to indicate that you have finished selecting objects.

   Abaqus/CAE displays the **Create Radial Pattern** dialog box.

4. From the **Create Radial Pattern** dialog box, configure the radial pattern:

   a. Click the arrows to the right of **Number** to increase or decrease the number of copies to create, including the selected objects. The number of copies in the Sketch updates when you click the arrows and provides a preview of the setting.

      Alternatively, you can type in a number and press **[Enter]** to preview the setting. You can enter any number between 2 and 1000.

   b. Enter the **Total angle** between the original objects that you selected and the final copy. The angle must be between –360° and +360°. A positive angle corresponds to a counterclockwise direction.

   c. By default, the center point of the circular pattern is the origin of the sketch. Click **Center** to change the location of the center point, and select a point from the sketch to define the new center point.

5. In most cases you will want to preview the radial pattern that Abaqus/CAE will create as you enter values in the **Create Radial Pattern** dialog box. However, if you choose to create a large number of copies, the preview capability may impact the performance of the Sketcher. In this case you should toggle off the **Preview** button.
6. To copy more objects, repeat the above steps beginning with Step 2.
7. When you have finished copying objects, do one of the following:

   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.

- Click the cancel button  in the prompt area.

Additional information

- *Copying sketch objects to create patterns*
- *Modifying objects*
- *Undoing and redoing sketching actions*

Offsetting the edges of Sketcher objects

Use the offset tool from the Sketcher toolbox to offset edges. You select the edges, enter the offset distance, and choose the offset direction. Abaqus/CAE applies the offset distance perpendicular to each point along the edges to create the offset.

Note:

Edges containing features with dimensions of similar size to the offset distance will lose their details in the offset copy.

1. From the Sketcher toolbox, select the offset tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.
2. Select a single group of connected edges to offset. Only sketch geometry can be offset; *construction geometry*, *reference geometry*, and dimensions cannot be offset.

   Tip: To select a group of edges that are connected end-to-end, choose **by chain** from the prompt area (for more information, see *Using the chain method to select edges in the Sketcher*). If necessary, you can then use the other selection methods in Abaqus/CAE to enhance your selection. For more information, see *Selecting objects within the viewport*."
3. Click mouse button 2 to indicate that you have finished selecting objects.
4. Specify an offset distance or accept the default value displayed in the prompt area.

   Abaqus/CAE displays a preview of the offset edges in the viewport.
5. Use the buttons that appear in the prompt area to accept or change the offset direction.
6. To offset more edges, repeat the above steps beginning with Step 2.
7. When you have finished offsetting edges, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Deleting Sketcher objects

Use the delete tool from the Sketcher toolbox to remove Sketcher objects: lines, arcs, circles, fillets, splines, points, constraints, dimensions, or parameters.

1. From the Sketcher toolbox, select the delete tool  . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

2. Select all the objects you want to delete. Select a filter from the prompt area to limit the available selections to **Geometry**, **Dimensions** (including parameters), or **Constraints**; the default filter, **Any**, does not limit your selection.

   Both sketch and *construction geometry* can be deleted; *reference geometry* cannot be deleted.

    **Tip:** To select more than one object, hold down the **[Shift]** key as you click each object or drag a rectangle around the objects. To unselect an object, use **[Ctrl] + Click**. For more information, see *Selecting objects within the viewport*."

3. When you have finished selecting objects, click mouse button 2.

   Abaqus/CAE deletes the selected objects and any associated dimensions or constraints.

    **Tip:** To restore accidentally deleted objects, click the *undo* tool  in the Sketcher toolbox.

4. To delete more objects, repeat the above steps beginning with Step 2.
5. When you have finished deleting objects, do one of the following:
   - Click mouse button 2 anywhere in the Abaqus/CAE window.
   - Select any tool in the Sketcher toolbox.
   - Click the cancel button  in the prompt area.

Additional information

- *Sketching simple objects*
- *Creating construction geometry*
- *Modifying objects*
- *Creating patterns, offsetting, and deleting objects*
- *Undoing and redoing sketching actions*

Undoing and redoing sketching actions

The Sketcher toolbox contains tools to undo and redo sketch actions.

Use the **Undo** tool  or the **Redo** tool  from the Sketcher toolbox to undo your previous operation or redo your most recently undone operation. For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

Depending on your previous action, clicking the undo tool can have the following effect:

- Remove the object just created. Objects include sketch geometry (lines, arcs, circles, ellipses, fillets, splines, or points), construction geometry, or dimensions.
- Restore the sketch to its state prior to the editing operation just performed. Editing operations include copying entities, deleting entities, moving vertices, trimming edges, extending edges, breaking edges, and changing dimensions.
- Remove a retrieved stand-alone sketch, provided you have not positioned the sketch. Once you have positioned the retrieved sketch, clicking the undo tool moves the origin of the retrieved sketch back to the origin of the current sketch.

You can undo and redo several actions in a row. For example, if you sketch a circle and dimension its radius, your first undo operation removes the dimension and your second undo operation removes the circle. You can restore the circle by clicking the redo tool and restore its dimension by clicking the redo tool a second time. Abaqus/CAE stores a history of actions for the current sketch, and you can specify the maximum number of sketching operations to store (see *Setting the maximum number of recorded sketching operations*, for more information).

The **Undo** and **Redo** tools do not perform view manipulation operations. For example, if you sketch a circle, magnify the view, and click the undo tool, Abaqus/CAE deletes the circle (it does not resize the view).

Additional information

• *Deleting Sketcher objects*

Resetting the view

The Sketcher toolbox contains a tool for resetting the view.

Use the reset view tool  from the Sketcher toolbox to return to the original view. For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

If you are unsure of the part's or assembly's orientation relative to the sketch plane, you can use the view manipulation tools to examine the sketch plane and the object on which you are sketching. When you click the reset view tool, Abaqus/CAE restores the view to the original view orientation displayed when you entered the Sketcher.

Additional information

- *Using the view manipulation tools*

Managing stand-alone sketches

This section describes how to manage stand-alone sketches while working in the Sketch module.

In this section:

- *Managing stand-alone sketches*
- *Creating stand-alone sketches*
- *Saving the current sketch as a stand-alone sketch*
- *Adding stand-alone sketches*

Managing stand-alone sketches

Stand-alone sketches are stored in your model and are independent of any particular part or the assembly. You can create a stand-alone sketch while working in the Sketch module, or you can save the current sketch as a stand-alone sketch while sketching. You can add a stand-alone sketch while creating or editing a feature or a partition, and an added sketch overlays the current sketch so that their origins are coincident.

You use the Sketch module to manage the stand-alone sketches defined in your model. To create, retrieve, copy, rename, and delete stand-alone sketches while working in the Sketch module, use one of the following:

- The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** items listed under the **Sketch** menu on the main menu bar. Each item contains a submenu listing all the sketches in the current model.
- The **Sketch Manager** dialog box. The **Sketch Manager** contains functions identical to those listed under the **Sketch** menu, but with a convenient browser that lists all the sketches available in the current model. To display the **Sketch Manager** dialog box, select **Sketch->Manager** from the main menu bar.



Note:

The **Sketch** menu and the **Sketch Manager** are available only from the Sketch module. When you are sketching a feature in the Part module, for example, the **Sketch** menu and the **Sketch Manager** are not available.

Additional information

- *Managing stand-alone sketches*
- *Stand-alone sketches*

Creating stand-alone sketches

Stand-alone sketches are stored in your model and are independent of any particular part or the assembly. To create a new stand-alone sketch in the current viewport, select **Sketch->Create** from the main menu bar.

1. From the main menu bar, select **Sketch->Create**.

   The **Create Sketch** dialog box appears.

   Tip: You can also create a stand-alone sketch by clicking the sketch create tool  in the Sketch module toolbox.

2. In the **Create Sketch** dialog box, enter a name for the sketch. For information on valid names, see *Using basic dialog box components*.
3. In the **Create Sketch** dialog box, enter the approximate size of the sketch.

   The size that you enter is used by Abaqus/CAE to calculate the size of the sheet and the spacing of its grid. The approximate size should reflect the largest dimension of the sketch and must be between 10^{5} and 10^{-3} units. Abaqus/CAE does not require specific units, but the units must be consistent throughout the model.
4. Click **Continue** to close the **Create Sketch** dialog box.

   Abaqus/CAE starts the Sketcher and displays a square sheet on which you sketch; the width and height of the sheet are approximately equal to the value that you entered in the previous step. If you later find that your sketch extends beyond the edge of the sheet, you can change the sheet size using the Sketcher *customization options*.
5. When you have finished sketching, click **Done** in the prompt area.

Additional information

- *Managing stand-alone sketches*
- *Stand-alone sketches*

Saving the current sketch as a stand-alone sketch

While you are sketching, you can save the current sketch as a stand-alone sketch. Stand-alone sketches are maintained independent of any features; they can be subsequently retrieved into the Sketcher and will overlay any existing geometry.

Abaqus/CAE saves only items within the sketch. *Reference geometry* is not saved. Similarly, dimensions between vertices and lines from reference geometry are not saved.

1. From the Sketcher toolbox, select the save as tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.
2. In the text field that appears in the prompt area, enter the name of the sketch.

   Abaqus/CAE saves the sketch and returns to the Sketcher.

Additional information

- *Managing stand-alone sketches*
- *Stand-alone sketches*

Adding stand-alone sketches

While you are sketching, you can add a stand-alone sketch so that its geometry overlays the current sketch. Abaqus/CAE positions the added stand-alone sketch so that its origin is coincident with the origin of the current sketch.

1. From the Sketcher toolbox, select the add sketch tool . For a diagram of the tools in the Sketcher toolbox, see *The Sketcher tools*.

   Abaqus/CAE displays the **Add Sketch** dialog box.

2. Select the sketch to add from the list of stand-alone sketches and click **OK** to close the **Add Sketch** dialog box.

   Abaqus/CAE does the following:

   - Positions the added sketch so that its origin is coincident with the origin of the current sketch.
   - Resizes the view so that both the current sketch and the added sketch are visible.

3. Use the buttons that appear in the prompt area to indicate how you will locate the added sketch.

   - Select **Translate** to translate the added sketch along a specified vector. You can define the beginning and the end of the translation vector by either selecting from the sketch or by typing the *X*- and *Y*-coordinates of each end.
   - Select **Rotate** to rotate the added sketch through a specified angle about a specified point. You can specify the point at the center of rotation by selecting from the sketch or typing the *X*- and *Y*-coordinates. The angle of rotation you specify must be between 360 and −360 degrees.
   - Select **Mirror** to mirror the added sketch about a selected line. You must specify the mirror line by selecting an existing straight line (not a line in the added sketch).
   - Select **Scale** to scale the spacing between the vertices of the added sketch by a specified amount about a specified point. You can specify the point by selecting from the sketch or by typing the *X*- and *Y*-coordinates.

   A preview of the added sketch is modified and positioned according to your specification.

4. If desired, repeat Step 3 to continue modifying or repositioning the added sketch.
5. Select **Done** in the prompt area.

   Abaqus/CAE adds the stand-alone sketch to the existing geometry.

Additional information

- *Managing stand-alone sketches*
- *Stand-alone sketches*

Modeling techniques

This part discusses modeling techniques in Abaqus/CAE that span multiple Abaqus/CAE modules and that define special engineering features in an Abaqus/CAE model.

In this section:

- *Adhesive joints and bonded interfaces*
- *Bolt loads*
- *Composite layups*
- *Connectors*
- *Continuum shells*
- *Co-simulation*
- *Display bodies*
- *Eulerian analyses*
- *Fasteners*
- *Fracture mechanics*
- *Gaskets*
- *Imperfection*
- *Inertia*
- *Load cases*
- *Midsurface modeling*
- *Skin and stringer reinforcements*
- *Springs and dashpots*
- *Submodeling*
- *Substructures*

Adhesive joints and bonded interfaces

This section provides information on how to model adhesive joints and bonded interfaces.

In this section:

- *Modeling adhesive joints and bonded interfaces*
- *Embedding cohesive elements in an existing three-dimensional mesh*
- *Creating a model with cohesive elements using geometry and mesh tools*
- *Defining tie constraints between the cohesive layer and the surrounding bulk material*
- *Assigning cohesive modeling data*

Modeling adhesive joints and bonded interfaces

You can create a model using cohesive elements to model adhesive joints, fracture at bonded interfaces, and gaskets.

You can model:

- Adhesive joints – two components connected by a glue-like material that has a finite thickness.
- Fracture at bonded interfaces – crack propagation in glue material that is very thin and for all practical purposes may be considered to be of zero thickness.
- Gaskets (limited capability) – seal between two components. No specialized gasket behavior (typically defined in terms of pressure versus closure) is available. If you want to model special gasket behavior, use special-purpose gasket elements as described in *Gaskets*.

For more information, see *About Cohesive Elements*.

The two main approaches that you can use to include cohesive elements in your model are

- embedding one or more layers of cohesive elements in the mesh of an existing model; or
- creating the analysis model using the geometry and mesh tools.

You can model the connection at the interface between the cohesive layer and the surrounding bulk material by sharing nodes or by defining a tie constraint. The tie-constraint approach allows you to model the cohesive layer using a finer discretization than that of the bulk material and may be more desirable in certain modeling situations. For more information, see *Defining tie constraints between the cohesive layer and the surrounding bulk material*.

Like gasket elements, cohesive elements have an orientation associated with them. This orientation defines the thickness direction of the elements, and it should be consistent throughout the cohesive layer. Swept or offset meshing techniques should be used to generate the mesh in the cohesive layer because these tools produce meshes that are oriented consistently. You can also use the bottom-up sweep meshing method, but you must sweep in the direction of the element thickness to maintain the correct orientation. You should create a single layer of solid elements to model the cohesive region. The use of more than one layer through the thickness could produce unreliable results and is not recommended. You can generate the mesh in the surrounding bulk material using any of the mesh tools because these elements do not need to be oriented.

The general procedure for modeling adhesive joints and bonded interfaces involves the following steps:

1. Create a model, and assign a cohesive element type to the cohesive region in the Mesh module using one of the following approaches:
   - *Embedding cohesive elements in an existing three-dimensional mesh*
   - *Creating a model with cohesive elements using geometry and mesh tools*
2. In the Property module, define a material and a cohesive section that refers to the material, and assign the cohesive section to the cohesive region. See *Assigning cohesive modeling data* for more information.
3. To model progressive damage and failure of cohesive layers as discussed in *Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*, include the desired damage initiation and damage evolution information in the material definition (select **Mechanical->Damage for Traction Separation Laws->*damage type***). For more information, see *Defining damage*.

Embedding cohesive elements in an existing three-dimensional mesh

You can use the following procedure to embed a layer of cohesive elements:

1. In the Mesh module, use the solid offset mesh tool in the Edit Mesh toolset to embed elements in an existing mesh (see *Editing the entire mesh*). This approach generates a layer of hexahedral or wedge elements that share nodes with the surrounding bulk material. The offset mesh tool generates elements that are oriented consistently with a stack direction that is aligned with the offset direction. When prompted to select the element faces from which the offset mesh will be generated, select the interior element faces using the procedure described in *Selecting interior surfaces*.

   Note: When generating an embedded layer of elements, the thickness should be much less than that of the adjacent elements because the nodes are displaced when the offset layer is generated.

   You can create an offset mesh from only three-dimensional element faces. As a result, you can create only hexahedral- and wedge-shaped cohesive elements using an offset mesh. For example, you cannot create quadrilateral cohesive elements by offsetting from element edges.

2. In the Mesh module, use the element type assignment tool to assign the cohesive element type to the cohesive region. See *Element type assignment* for more information.

If your existing mesh is a native mesh, you should create a mesh part prior to adding cohesive elements. For more information, see *Creating a mesh part*.

If you want to use a finer mesh in the cohesive layer, you should construct the cohesive layer as a separate part. You should separate the mesh in the surrounding bulk material into two regions with the appropriate spacing to accommodate the cohesive layer. You can mesh the cohesive layer using the methods described in *Creating a model with cohesive elements using geometry and mesh tools*, and tie the cohesive layer to the surrounding bulk material. For more information, see *Defining tie constraints between the cohesive layer and the surrounding bulk material*.

Creating a model with cohesive elements using geometry and mesh tools

You can use the following procedure to create a model with cohesive elements using geometry and mesh tools:

1. In the Part module, define the geometry of the model. You should define the geometric region that represents the cohesive layer as a solid, even if the thickness of the layer is close to zero. To avoid numerical problems, it is recommended that you model the geometry using a value of 10^{-4} or greater for the thickness. If the actual thickness of the layer is less than this value, you should specify the actual thickness in the **Initial thickness** field of the cohesive section editor as described in *Creating cohesive sections*
2. In the Mesh module, mesh the surrounding bulk material. You can use any of the mesh tools to mesh the surrounding bulk material. See *Mesh generation* for more information.
3. In the Mesh module, mesh the cohesive region using one of the following methods:

Two- and three-dimensional models

Top-down swept or bottom-up meshing technique. You can assign the top-down swept meshing technique or the bottom-up meshing technique to mesh the cohesive region. The bottom-up meshing technique is available only for three-dimensional models (for more information, see *Bottom-up meshing*). Regardless of the meshing technique that you choose, you must sweep, extrude, or revolve the mesh in the thickness direction of the element to produce the correct element orientation. For a complex cohesive region, you may need to partition the model to create a group of sweep regions that you can align consistently. For more information, see *Selecting a meshing technique*, and *Specifying the sweep path*.

Three-dimensional models

Convert the cohesive region into a shell region, and use the offset meshing technique.

1. Convert the solid part to a shell using the **From solid** shell tool in the Part module.
2. Isolate a collection of faces that represents an idealized shell of the part using the **Remove faces** tool in the Geometry Edit toolset.
3. Mesh the simplified model with shell elements, and create a mesh part.
4. Use the mesh part to generate an offset mesh of solid hexahedral or wedge elements. The elements will be oriented through the thickness of the part, and you can verify this with the Query toolset.

For detailed instructions, see *Generating layers of solid elements offset from an existing mesh*.

4. In the Mesh module, use the element type assignment tool to assign the cohesive element type to the cohesive region. See *Element type assignment* for more information.

For example, *Figure 1* illustrates the layered composite specimen that is used in the benchmark problem *Delamination analysis of laminated composites*. An Abaqus Scripting Interface script that reproduces the composite specimen model using Abaqus/CAE is provided with this problem.

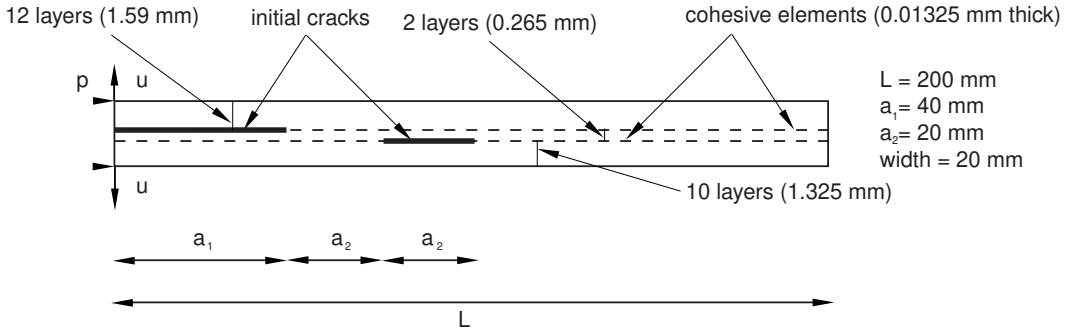


Figure 1: Model geometry for the Alfano delamination problem.

Defining tie constraints between the cohesive layer and the surrounding bulk material

If you want to model the cohesive layer using a mesh that is finer than the adjacent bulk material mesh, the cohesive layer should be generated as a separate mesh and tied to the bulk material using tie constraints (see *Defining tie constraints*). You should create a shell geometric model to represent the surface on one side of the cohesive layer and mesh that model with the desired mesh density. You can use this mesh to create a mesh part from which you can generate the offset mesh. For more information, see *Modeling with Cohesive Elements*.

You should use care when tying zero-thickness cohesive layers to the surrounding bulk material mesh. Use the following procedure to avoid problems:

1. Use the solid offset mesh tool to create the layer of cohesive elements along with the top and bottom surfaces. Abaqus/CAE appends **TopSurf** and **BottomSurf** to the surface name prefix when it creates the two surfaces on either side of the layer of cohesive elements.
2. Use the stack orientation query tool to distinguish the top (brown) and bottom (purple) faces of the cohesive elements.
3. Tie the surfaces of the surrounding bulk material mesh to the appropriate top and bottom cohesive surfaces. When you are prompted to select a surface from the cohesive surface, click **Surfaces** from the right side of the prompt line and select the appropriate surface. For example, select ***surface name*-BottomSurf** to tie the bottom (purple) face of the cohesive layer to the adjacent surface of the bulk material mesh.

Assigning cohesive modeling data

In the Property module, you can complete the cohesive modeling data as follows:

1. Create a cohesive section to define the section properties of adhesive layers at the interface between two bonded parts. You can model a negligibly thin layer of adhesive, a finite-thickness adhesive layer, or a gasket by selecting the constitutive behavior of the cohesive section as **Traction Separation**, **Continuum**, or **Gasket**, respectively. For more information, see *Creating cohesive sections*.
2. Define a material for the cohesive region. If you select the **Continuum** or the **Gasket** response in the cohesive section editor, you can use the material models available in the Property module to define the continuum material properties. Abaqus/CAE does not support the material models for the **Traction Separation** response. In this case you must add the material definition using the **Keywords Editor**, as described in *Adding unsupported keywords to your Abaqus/CAE model*.
3. Assign the cohesive section to the cohesive region. For more information, see *Assigning a section*.

Bolt loads

This section explains how to model tightening forces or length adjustments in bolts or fasteners.

In this section:

- *Understanding bolt loads*
- *Creating and editing bolt loads*

Understanding bolt loads

Bolt loads model tightening forces or length adjustments in bolts or fasteners.

Figure 1 shows a container (A) that is sealed by tightening the bolts that hold the lid, which places the gasket under pressure.

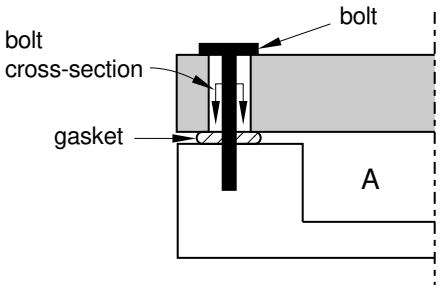


Figure 1: Modeling a pre-tensioned bolt.

You can model the tension in the tightened bolts by applying a bolt load to each one in the first step of the analysis. You define the load in terms of either a concentrated force or a prescribed change in length, and you apply the load across a bolt cross-section surface that you specify. In later steps you can modify the load to prevent further length changes so that the bolt acts as a standard, deformable component responding to other loadings on the assembly.

When you create a bolt load, you must specify the following:

A surface that defines the bolt cross-section

Abaqus/CAE applies the bolt load across the cross-section surface that you specify. The surface that defines the bolt cross-section must cut through the bolt geometry. Abaqus/CAE creates an “interior” surface at that location.

If you are working with bolt part instances made from native or imported geometry, Abaqus/CAE can create an interior surface by partitioning the bolt shank surface, or you can partition the bolt at the location where you want the cross-section to be defined. For example, a partition is selected as the bolt cross-section in *Figure 2*. For more information, see *The Partition toolset*.

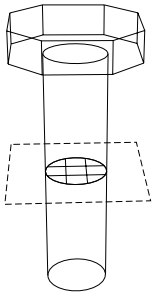


Figure 2: Using a partition to specify the bolt cross-section.

If you are working with orphan mesh elements, you must specify the cross-section surface by selecting element faces. For example, element faces define a cross-section surface for the mesh part shown in *Figure 3*.

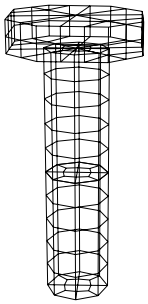


Figure 3: Using element faces to specify the bolt cross-section.

The element faces need to be selected from elements on only one side of the pre-tension section. You can use display groups to remove selected elements from the viewport to reveal the element faces of the cross-section surface. For more information on selecting surfaces, see *Selecting objects within the viewport*. For detailed information on selecting surfaces on wire part instances, see *Specifying a particular side or end of a region*.

Note:

You can apply bolt loads only to three-dimensional solid, two-dimensional solid, and three-dimensional wire part instances. Bolt loads on two-dimensional and axisymmetric wire part instances are unsupported.

A bolt axis

If you are defining a bolt load on a solid region, the surface normal of the surface that defines the bolt cross-section is used as the bolt axis by default. You can select a datum axis to indicate a different bolt axis direction (it need not be normal to the cross-section). If you are defining a bolt load on a wire region, the bolt axis direction is always assumed to be the direction of the tangent to the wire at the bolt cross-section.

Abaqus/CAE uses both the cross-section surface that you specify and the bolt axis to define the pre-tension section data and the pre-tension reference node used by Abaqus/Standard; see *Prescribed Assembly Loads* for more information. You can create the pre-tension section at the part level or the assembly level. To create the pre-tension section at the part level, the selected region must belong to a dependent part instance. If you create the pre-tension section at the part level, the bolt load is defined for all of the instances from the same part.

A method for applying the loading

When you create a bolt load, you must choose one of the following loading methods:

- Apply a force to the bolt. This method models tightening the bolt so that it carries a specified load.
- Adjust the bolt length. This method models tightening the bolt until its free length has changed by the specified value.
- Fix the bolt at its current length. This method is available only if you have already created the load in the first analysis step and are now editing it in a subsequent analysis step. This method allows the bolt length to remain unchanged so that the force in the bolt can change according to the response of the model.

A magnitude for the chosen method

If you are applying a force to the bolt, you must enter the force magnitude. If you are adjusting the bolt length, you must enter the length change.

You can create a bolt load only in the first analysis step, but you can modify the loading method or the magnitude of the load in subsequent steps. For example, you can apply a specific tension in the first step and then change the method in the second step to fix the bolt length.

You can obtain data from a bolt load using the field and history output request editors in the Step module. In the **Domain** section of the editor, select **Bolt load** and choose the desired bolt load from the menu that appears. For more information, see *Creating an output request*.

For detailed information on creating a bolt load, see *Creating and editing bolt loads*.

Creating and editing bolt loads

You can create bolt loads to model tightening forces or length adjustments in bolts or fasteners.

1. From the main menu, select **Load->Create**.

   Abaqus/CAE displays the **Create Load** dialog box.

2. In the **Create Load** dialog box, do the following:
   a. From the **Category** list, select **Mechanical**.
   b. From the **Types for Selected Step** list, select **Bolt Load**, and click **Continue**.

3. Select the method to specify the surface that defines the bolt cross-section.

   - Select **Interior Surface** to use a previously created interior surface.
   - Select **Bolt Shank Surface** to create an interior surface by partitioning the bolt shank surface.

4. If you selected the **Interior Surface** method, do the following:
   a. Select the interior surface or wire segment that indicates the location of the bolt load.

      - If you are working with native or imported geometry, use the mouse to select the interior surface or wire segment in the viewport. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face or edge. For more information, see *Selecting objects within the current viewport*.

        Tip: If you are unable to select the desired faces or edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

      - If you are using orphan mesh elements, you must select element faces to specify the interior surface. You can use display groups to remove selected elements from the viewport to reveal the element faces of the cross-section surface. For more information, see *Using display groups to display subsets of your model*.

      When you have finished selecting, click mouse button 2.

   b. Choose the side on which the surface is defined using the techniques described in *Specifying a particular side or end of a region*. The side you select determines which elements are adjusted to produce the desired tightening load or length adjustment (see *Prescribed Assembly Loads* for details).

   Abaqus/CAE displays the bolt load editor.

5. If you selected the **Bolt Shank Surface** method, do the following:
   a. Select the bolt shank face to partition. You can only select a cylindrical face.
   b. In the prompt area, enter the cut position value to define the partition location.
   c. Click **Done**.

   Abaqus/CAE creates an interior surface by partitioning the selected face and displays the bolt load editor.

6. Click the arrow next to the **Method** field and select the loading method of your choice from the list that appears.
7. In the **Magnitude** field, enter the force magnitude (for the **Apply force** method) or the change in length (for the **Adjust length** method).



Note:

The **Fix at current length** method becomes available if you edit the load in a step that follows the step in which you create the load. If, while editing the load, you change the method to **Fix at current length**, the **Magnitude** field becomes unavailable.

8. If desired, specify an amplitude. (See *The Amplitude toolset* for more information.)
9. By default, the surface normal of the surface that defines the bolt cross-section is used as the bolt axis. If desired, you can define a different bolt axis direction.
   a. Select **Specify**.
   b. Select a datum axis.

   - To create a datum axis, click  and select two points in the viewport that define the axis. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
   - Click , and select a datum axis that is aligned with the bolt centerline.

10. The pre-tension section is written at the assembly level. Toggle on **Pre-tension section at part level** to write the pre-tension section at the part level (applicable only for dependent part instances). If you selected the **Pre-tension section at part level** option, the bolt load is defined for all of the instances from the same part.
11. Click **OK** to create the load and to exit the editor.

    Arrows appear in the viewport that represent the bolt load that you just created. For more information, see *Understanding symbols that represent prescribed conditions*.

Additional information

- *Creating and modifying prescribed conditions*
- *Understanding symbols that represent prescribed conditions*
- *Prescribed Assembly Loads*

Composite layups

This section provides information on modeling composite layups with Abaqus/CAE.

In this section:

- *An overview of composite layups*
- *Creating a composite layup*
- *Understanding composite layups and orientations*
- *Understanding composite layups and distributions*
- *Requesting output from a composite layup*
- *Viewing a composite layup*

An overview of composite layups

Figure 1 illustrates a single composite layup that contains three plies. Each ply is composed of a homogenous material of uniform thickness, with fibers oriented along a single orientation. However, a ply can also be an isotropic material, such as a foam core.

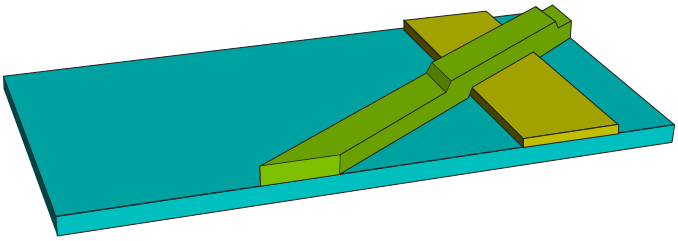


Figure 1: A composite layup.

A ply represents a single piece of material that is placed in a mold during the composites manufacturing process. A composite layup can contain a different number of plies in different regions. For example, the composite layup in *Figure 1* includes regions that contain a single ply, regions where two plies overlap, and a region where three plies overlap.

Figure 2 shows the same model that is depicted in *Figure 1* but defined using composite shell sections. The number of plies cannot change within a composite shell section. Therefore, four composite shell sections, each with a constant number of plies, are required to define this simple model. Section 1 contains a single ply, sections 2 and 3 each contain two plies, and section 4 contains three plies.

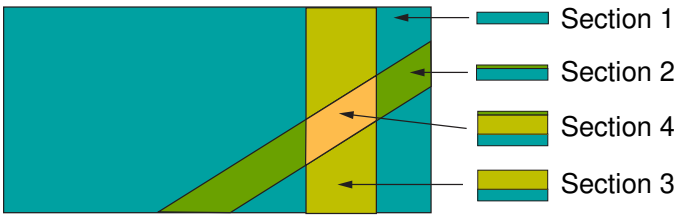


Figure 2: Composite shell sections.

Composite layups in Abaqus/CAE are designed to help you manage a large number of plies in a typical composite model. In contrast, composite sections are a product of finite element analysis and may be difficult to apply to a real-world application. Unless your model is relatively simple and all plies cover the same region, you will find it increasingly difficult to define your model using composite sections as you increase the number of plies. It can also be cumbersome to add new plies or remove or reposition existing plies.

The procedure for creating a composite layup with Abaqus/CAE mirrors the procedure for creating a real composite part—you start with a basic shape and then you add plies of different materials and thicknesses to selected regions and orient the plies to provide the greatest strength in a particular direction. The Abaqus/CAE composite layup editor allows you to easily add a ply, choose the region to which it is applied, specify its material properties, and define its orientation. You must specify ply names that are unique throughout the entire model to ensure the correct display of ply-based results. You can also read the definition of the plies in a layup from the data in a text file. This is convenient when the data are stored in a spreadsheet or were generated by a third-party tool. You can also suppress plies in your layup and experiment with the effect of adding and removing plies of different orientations.

It is important to specify the correct orientation of the fibers in a ply. Abaqus/CAE allows you to define a reference orientation for the layup as well as a reference orientation for each ply in the layup. In addition, you can specify the direction of the fibers in a ply relative to the reference orientation of the ply. By default, the coordinate system of a layup is the same as the coordinate system of the part; similarly, the coordinate system of a ply is the same as the coordinate system of the layup. Orientation definition is described in more detail in *Understanding composite layups and orientations*.

Using a composite layup to model a yacht hull, illustrates how you can analyze a complex three-dimensional model using the composite layup capability in Abaqus/CAE.

Creating a composite layup

Abaqus/CAE allows you to define composite layups for three types of elements: conventional shells, continuum shells, and solids. For detailed instructions, see the following sections:

- *Creating conventional shell composite layups*
- *Creating continuum shell composite layups*
- *Creating solid composite layups*

You create a composite layup in the Property module. The composite layup editor provides a table that you use to define the plies in the layup. For each ply you specify the ply's name, material, thickness, and orientation, as well as the number of integration points and the region of the model to which the ply is assigned. The composite layup editor's ply table is shown in *Figure 1*.

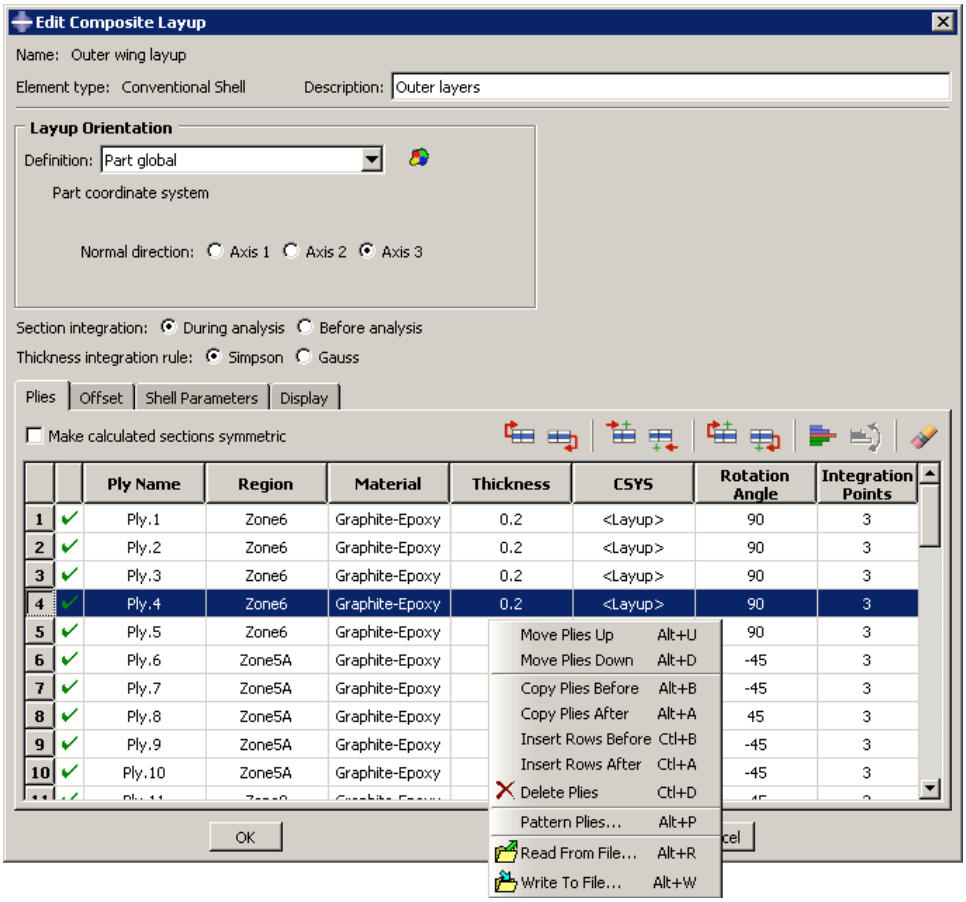


Figure 1: The ply table in the composite layup editor.

The ply table provides several options that make it easier for you to define a layup containing many plies; for example, the ply table allows you to do the following:

- Move or copy selected plies up or down in the table.
- Suppress or delete selected plies.
- Pattern a group of selected plies.
- Read ply data from or write ply data to an ASCII file. The data can define all the plies in the layup or only a subset of the plies in the layup.

The ability to suppress plies allows you to easily experiment with different configurations of plies in the composite layup and to see the effect on the results of an analysis of the model. For more information, see *Using the ply table when defining a composite layup*.

If you are defining a composite layup whose plies are symmetric about a central core, you only need to enter the bottom half of the layup in the ply table. When you apply the symmetry option, Abaqus automatically creates additional plies in the generated sections by repeating the entered plies (including the central ply) in reverse order.

You will probably have to partition your model to create the regions to which you will assign plies. You should create the partitions before you create the layup and start defining plies. You can select a region directly from the part in the current viewport, or you can create a named set that refers to the region and select the named set. Abaqus/CAE preserves any existing ply regions if you decide to add partitions and plies after you have defined a composite layup; for example, you can add plies to a layup that model a rib providing additional stiffness to a region. The region to which you assign a ply can be either Abaqus/CAE geometry or a mesh. You can combine plies assigned to geometry and plies assigned to a native mesh in a single composite layup.

Continuum shell and solid composite layups are expected to have a single element through the entire thickness across a combination of all the regions specified in the composite layup. Each single element through the thickness contains the multiple plies that you defined in the ply table. If the region to which you assign your continuum shell or solid composite layup contains multiple elements through the thickness, each element will contain all of the plies that you defined in the ply table and the analysis results will not be as expected.

If your model contains multiple continuum shell or solid elements through the thickness of a region, you can obtain correct results by defining a separate composite layup for each layer of elements. You can define a composite layup for each layer by selecting the elements of a native Abaqus/CAE mesh or the orphan elements when you specify the region of the layup. You must create a layup for each layer of elements.

If you have plies that overlap in your composite layup, you must enter the plies in the ply table in the order that they appear in the overlapping region. *Figure 2* illustrates a simple example of an overlapping region and the corresponding order of the plies in the ply table. The first ply in the ply table represents the bottom ply in the layup.

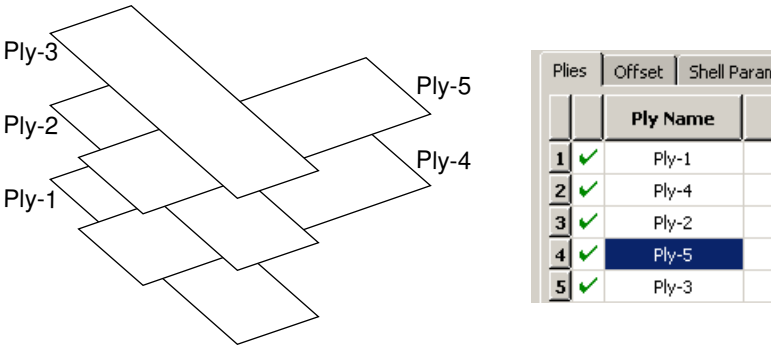


Figure 2: The order of the plies in the ply table.

Under certain circumstances, Abaqus cannot determine the orientations of the composite ply for conventional shells. This can occur, for example, if your ply makes a sharp transition through an angle of 90°, and/or your part is aligned with one or more planes of the global coordinate system. To help you diagnose the problem region, Abaqus/CAE draws a collapsed coordinate system indicating regions for which the user-selected coordinate system and the geometric normal cannot be resolved into a valid orientation for display purposes. If this situation arises, you may be able to complete the analysis by splitting the ply into multiple plies at the transitions and by assigning orientations to each new ply.

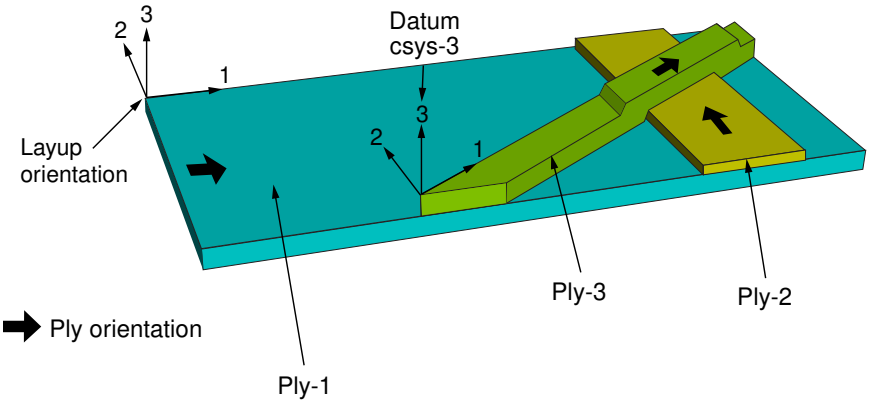
If you apply a composite layup and assign a section assignment to the same region, Abaqus/CAE uses only the properties from the composite layup during the analysis. If you apply two or more composite layups to regions that overlap, Abaqus/CAE uses the properties of the last layup, where “last” refers to the alphabetical order of the names of the composite layups. The default color coding in the Property module uses yellow coloring to indicate a region with

overlapping composite layups and section assignments or to indicate a region with multiple overlapping composite layups.

Understanding composite layups and orientations

The orientation of the fibers within each ply of a composite layup plays an important role in determining the physical qualities of your model; however, defining this orientation in a model based on a real-world application is not straightforward.

Composite layups in Abaqus/CAE make the process more manageable by deriving the orientation of the fibers from three parameters that are relative to each other—the layup orientation, the ply orientation, and an additional rotation—as shown in *Figure 1*.



		Ply Name
1	✓	Ply-1
2	✓	Ply-2
3	✓	Ply-3

	CSYS	Rotation Angle	Integra Point
	<Layup>	0	3
	<Layup>	90	3
	Datum csys-3.3	0	3

Figure 1: Determining the ply orientation.

Layup orientation

The layup orientation defines the base or reference orientation of all the plies in the layup. In conventional and continuum shell layups, Abaqus projects the specified orientation onto the surface of the shell, aligning the direction of the layup orientation that you choose with the shell normal. In solid composite layups the orientation is not projected.

Abaqus/CAE provides several options for defining the layup orientation:

- **Part global**: By default, the layup orientation is the same as the orientation of the part.
- **Coordinate system**: You can create and select a datum coordinate system that defines the orientation.
- **Discrete orientation**: You can create a discrete orientation that provides an orientation value for each mesh element to define the orientation.
- **Discrete field**: You can create and select an orientation discrete field that defines a spatially varying orientation.
- **User-defined**: You can define the orientation in user subroutine *ORIENT*. This option is valid only for Abaqus/Standard analyses.
- **Normal direction**: For all options except **User-defined**, you can choose which axis defines the approximate normal direction of the composite layup.

- **Additional rotation**: If you choose a **Coordinate system**, **Discrete orientation**, or **Discrete field** to define the layup orientation, you can specify an angle (in degrees) that defines an additional rotation about the specified normal direction for the entire layup. You can use a scalar discrete field to specify a spatially varying additional rotation angle.

Ply orientation

The ply orientation defines the relative orientation of each ply. In conventional and continuum shell layups Abaqus projects the specified ply orientation onto the surface of the shell so that the ply normal direction is aligned with the shell normal and the layup stacking direction. In solid composite layups the plies are created with respect to the layup stacking direction, and the unprojected ply orientation defines the material orientation within a ply (see *Figure 2*).

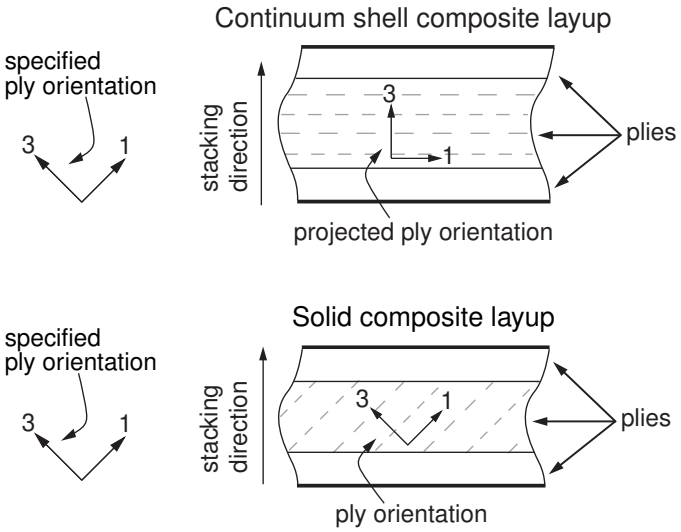


Figure 2: Plies and ply orientations for shell and solid composite layups.

Abaqus/CAE calculates the ply orientation from a combination of two variables that you can specify—the coordinate system (CSYS) and a rotation angle about the normal direction.

In cases where Abaqus/CAE attempts to draw coordinate systems for ply orientations in composite layups at singularity points of the system (i.e., points for which the user-selected coordinate system and the geometric normal from either geometry or elements cannot be resolved into a valid orientation for display purposes in Abaqus/CAE), the coordinate system will be drawn collapsed.

If the layup orientation is specified using a discrete field, no display is available for either the layup or ply orientation. For continuum shell elements, Abaqus/CAE does not project the displayed orientations onto the midplane surface. In both of these cases, you can perform a data check and view the output database in the Visualization module to verify the orientations. For more information, see *Performing a data check on a model*.

Coordinate system

Abaqus/CAE provides the following options for defining the coordinate system of the ply:

- **Layup**: By default, the coordinate system of the ply is the same as the coordinate system of the layup.
- **CSYS**: You can create and select a datum coordinate system that defines the coordinate system of the ply. If you choose to use a coordinate system for a ply, it overrides the layup orientation for that ply.

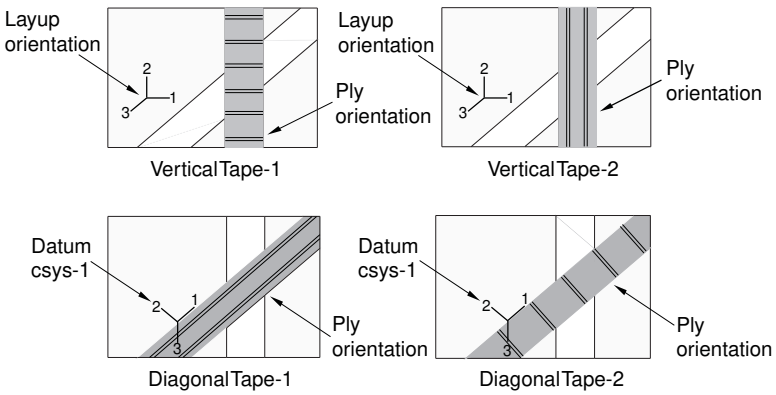
  You can also select the axis of the coordinate system that defines the normal direction of the ply. The axis that you choose appears as the last digit in the **CSYS** column in the layup table. For example,

`Datum csys1.3` indicates that you chose `Datum csys-1` to define the coordinate system of the ply, and you chose the `3-axis` to define the normal.

Rotation angle

The rotation angle defines the orientation of the fibers within each ply relative to the ply's coordinate system. For example, in a typical composite the fibers might be oriented at −45° or +90° relative to the coordinate system. You can also use a scalar discrete field to specify a fiber orientation that varies spatially across the ply. If you specify a rotation angle, the ply is rotated counterclockwise about the coordinate system normal, and the angle is measured relative to the 1-axis.

Figure 3 shows the orientation of four plies in a composite layup and the corresponding entries in the layup table.



		Ply Name	CSYS	Rotation Angle
1	✔	VerticalTape-1	<Layup>	0
2	✔	VerticalTape-2	<Layup>	90
3	✔	DiagonalTape-1	Datum csys-1.3	0
4	✔	DiagonalTape-2	Datum csys-1.3	90

Figure 3: Determining the orientation of each ply.

Abaqus/CAE determines the ply orientation as follows:

- You selected the layup orientation to define the coordinate system of **VerticalTape-1** and entered a rotation angle of 0°. The resulting ply orientation is along the 1-axis of the layup orientation.
- You selected the layup orientation to define the coordinate system of **VerticalTape-2** and entered a rotation angle of 90°. The resulting ply orientation is a rotation of 90° counterclockwise about the 3-axis (the normal direction) of the layup orientation. The rotation angle is measured relative to the 1-axis.
- You selected **Datum csys-1** to define the coordinate system of **DiagonalTape-1** and entered a rotation angle of 0°. The resulting ply orientation is along the 1-axis of **Datum csys-1**.
- You selected **Datum csys-1** to define the coordinate system of **DiagonalTape-2** and entered a rotation angle of 90°. The resulting ply orientation is a rotation of 90° counterclockwise about the 3-axis (the normal direction) of the datum coordinate system. The angle is measured relative to the 1-axis.

Understanding composite layups and distributions

A discrete field that is used by a composite layup is called a distribution. Because distributions are applied to specific elements and nodes, you can use a distribution only after you have meshed the part.
A discrete field is a spatially varying field where values are associated with a node or an element. The Discrete Field toolset allows you to create and manage discrete fields in Abaqus/CAE. In most cases you will use a third-party preprocessor that operates on meshed parts to create a distribution that can be applied to a composite layup. For more information, see *The Discrete Field toolset*.

You can use a distribution for the following:

- To define a spatially varying local coordinate system that specifies the overall orientation of the composite layup.
- To define an additional rotation for the layup orientation that is varying spatially across the layup.
- To define an additional rotation for a ply orientation that is varying spatially across the ply. You can use a distribution to define the ply orientation only in an Abaqus/Standard analysis.
- To define an overall shell thickness that varies spatially across a conventional shell composite layup.
- To define a ply thickness that varies spatially across a ply in a conventional shell composite layup. You can use a distribution to define the ply thickness only in an Abaqus/Standard analysis.
- To define a nodal thickness that varies spatially across a conventional shell composite layup.
- To define an offset that varies spatially across a conventional shell composite layup. You can use a distribution to define the offset only in an Abaqus/Standard analysis.

Requesting output from a composite layup

When you create a composite layup that will be integrated during the analysis, you can specify the number of integration points in each ply of the layup. By default, for shell sections integrated during the analysis Abaqus/CAE creates three integration points for each ply of a shell or continuum shell composite layup and one integration point for each ply of a solid composite layup. For pre-integrated shell sections, Abaqus/CAE creates three integration points for each ply in the layup, and you cannot change this value. *Figure 1* shows the integration point numbering for a composite layup with three plies and three integration points per ply.

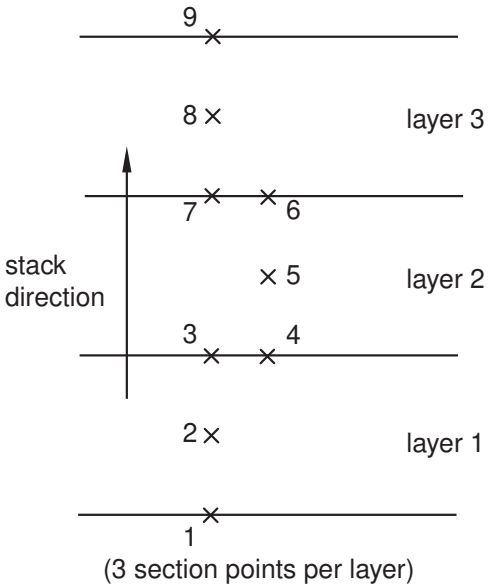


Figure 1: The integration point numbering for a composite layup with three plies.

If you do not create an output request, Abaqus writes field output data from only the top and bottom of a composite layup (the highest and lowest integration points), and no data are generated from the other plies. Therefore, if your model contains a composite layup and you want data from individual plies or interior integration points, you must create a new output request or edit the default output request and specify the composite layup from which variables will be output.

When you create an output request, Abaqus/CAE by default writes field and history data to the output database from only the middle section point in each ply of a composite layup. To change the default behavior, you can use the field and history output request editors to edit the output request and change the domain to a composite layup. You can then request field or history data from the top, middle, and/or bottom section point of each ply, from all section points, or from specified section points. For more information, see *Modifying field output requests*, and *Modifying history output requests*. You can request the following:

Selected

Abaqus/CAE writes field data to the output database from the selected section points of each ply (top, middle, and/or bottom). If a ply has an even number of section points and you request output from the middle section point, Abaqus/CAE generates data from the higher of the two section points that span the middle of the ply. For example, if a ply has four section points and you request output from the middle section point, Abaqus/CAE generates data from the third section point.

All

Abaqus/CAE writes field data to the output database from all of the section points of all of the plies.

Specify

Abaqus/CAE writes field data to the output database from specified section points. Section points are numbered sequentially from the bottom of the bottom ply to the top of the top ply, where the bottom ply is the first ply in the layup. For example, if you want output from the middle section point of each ply shown in *Figure 1*, you would enter `2,5,8`.

For more information, see *Understanding output requests*.

Viewing a composite layup

You can view a composite layup while you are creating it or after it has been analyzed. Abaqus/CAE provides the following tools for ply-based visualization of a composite layup:

Ply stack plot

A ply stack plot is a graphical representation of the plies from the selected region of a composite layup or a composite section. *Figure 1* illustrates a ply stack plot.

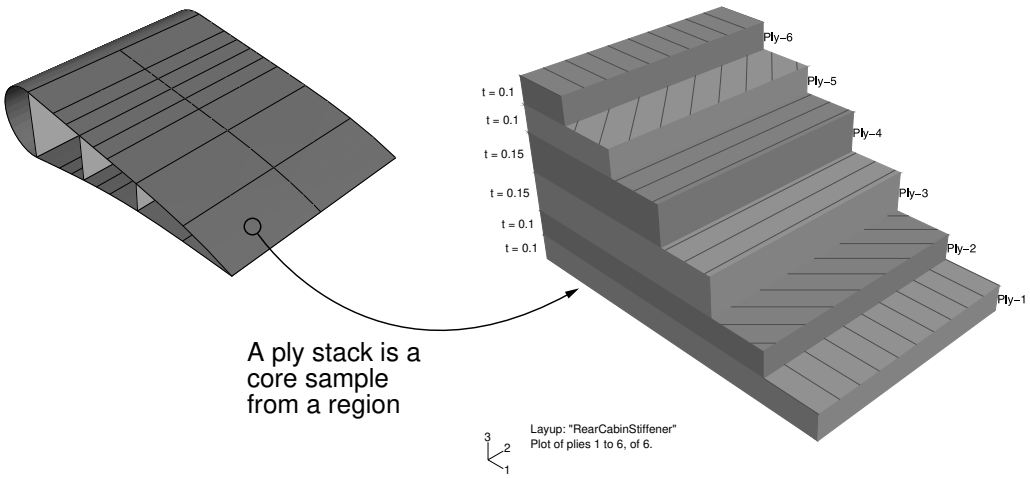


Figure 1: A ply stack plot.

The staircase appearance does not indicate ply drop-offs in the layup; it is simply a graphical representation that allows you to see the number of plies in the layup and, for example, the relative thickness of a ply, the material from which it is constructed, and the orientation of its fibers. If your composite layup contains many plies, the ply stack plot can become confusing and hard to interpret. To make the ply stack plot more readable, Abaqus/CAE provides options that allow you to view only a manageable number of plies.

The triad in a ply stack plot represents the element orientation system, and it shows the shell normal or stacking direction (the 3-direction) and the 1- and 2-directions for the plies. The fibers are always drawn in the 1–2 plane at an angle with respect to the 1-direction. In solid composite layups the fibers in a ply do not always run parallel to the 1–2 plane (e.g., if the 3-direction of the ply orientation and the element stacking direction are not aligned). In this situation the fibers in the ply stack plot are not a true depiction of the fibers in the layup, but rather a graphical representation of the rotation angles in the layup definition: the angle drawn in the ply stack plot is the rotation angle specified in the ply table measured about the element stacking direction axis. For more information, see *Understanding composite layups and orientations*.

Ply stack plots do not draw fibers for ply orientations that are based on a user-defined coordinate system or a rotation angle distribution. Abaqus/CAE displays an asterisk (for coordinate systems) or a caret (for rotation angle distributions) on such plies to signify that it cannot draw lines in the 1–2 plane that accurately represent the fiber direction for the ply. Similarly, if you use a discrete field distribution to define a ply's thickness, Abaqus/CAE draws the ply in the ply stack plot based on the average thickness of the other uniform plies in the layup and displays the discrete field name next to the plot (if thickness labels are turned on).

You can display a ply stack plot in the Property module after you have created a composite layup. You can also display a ply stack plot in the Visualization module after you have analyzed a model with a composite layup. For more information, see *Viewing a ply stack plot*.

Ply-based results

After you have analyzed a model with a composite layup, you can view a contour plot of results from individual plies of the layup, or you can view an envelope contour plot that looks for maximum or minimum values across all of the plies in the layup.

Results from individual plies

Ply-based results show the data from a selected ply in a composite layup. For a given ply, you can view data from the bottom, middle, or top of the ply, or you can view data from both the top and bottom plies. *Figure 2* illustrates a contour plot of strain (E11) from selected plies of a composite layup.

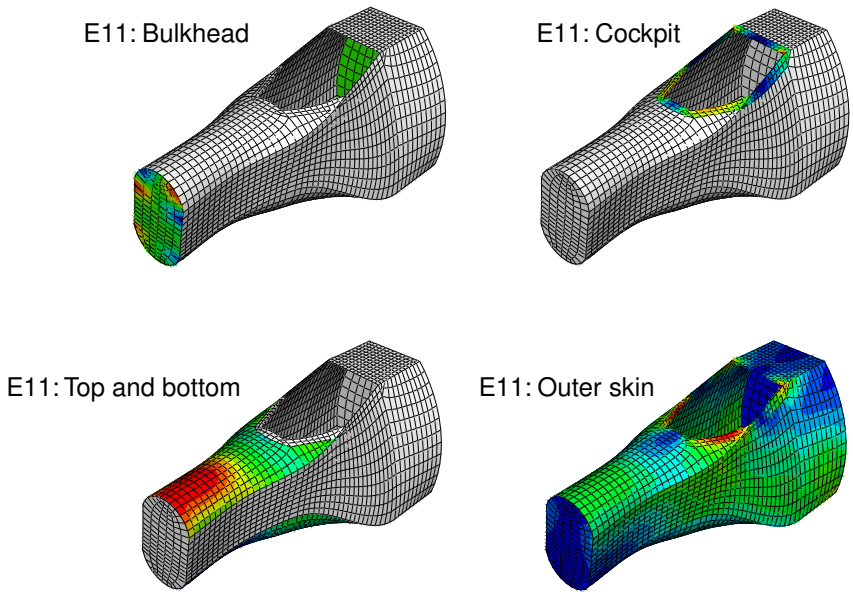


Figure 2: Contour plots from selected plies.

If the same ply name is used in multiple composite layup definitions within a model, viewing data for this ply displays results for all layups in which the ply name appears. To limit the results to a single ply within a single layup, you must first use a display group to limit the display to the individual composite layup (see *Selection methods for creating or editing a display group*), then plot the results for a section point within the desired ply (see *Selecting section point data by ply*).

Contour plots displaying output at both the top and bottom plies of a composite layup vary in appearance depending on the type of layup. In a conventional shell composite layup the two contours appear as a single double-sided shell with different contours on each side. In a continuum shell composite layup or solid composite layup the two contours appear as distinct single-sided contours at each section point location. *Figure 3* illustrates the difference between contour plots of conventional shell and continuum shell composite layups.

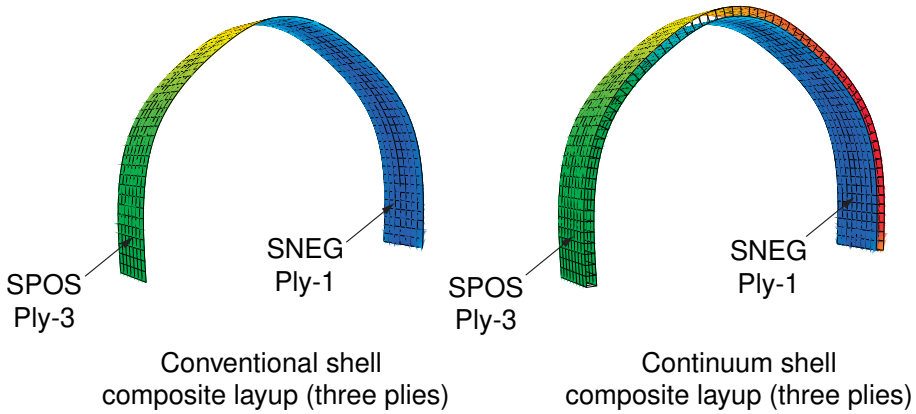


Figure 3: Contour plots of conventional shell and continuum shell composite layups.

Each layup contains three plies, and each plot is showing the output from the top of the top ply and from the bottom of the bottom ply. Stress (S11) is plotted in both cases. For more information, see *Selecting section point data by category*.

Envelope plots

An envelope plot allows you to view a contour plot of the highest or lowest value of a variable in your model, regardless of the ply in which it is occurring. The plies in an envelope plot that correspond with the extreme values are called the critical plies. You can choose the criterion that Abaqus/CAE applies (absolute maximum, maximum, or minimum) and the position in the ply at which Abaqus/CAE checks the value (integration point, centroid, element nodal).

For example, even though your composite layup can include a large number of plies, you can view only the critical plies and determine the highest strain that is occurring in each element of your model. You can decide if you want to reduce the strain in the critical plies by increasing the number of plies in a particular region or by reorienting existing plies. *Figure 4* shows the value of the strain (E11) in the critical plies of the model.

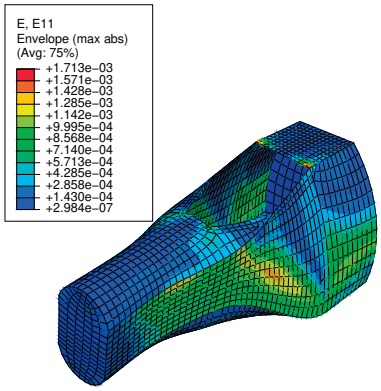


Figure 4: The value of the strain in the critical plies.

In addition, you can use the contour plot options to display a quilt contour plot where the color of each element indicates which ply is the critical ply, as shown in *Figure 5*.

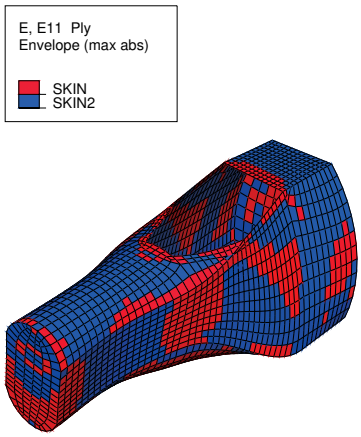


Figure 5: The critical plies.

Using a combination of the plots shown in *Figure 4* and *Figure 5*, you can determine both the value of the strain in the critical ply and the location of the critical ply in the layup. For more information, see *Selecting the field output to display*, and *Understanding how contour values are computed*.

If you do not create an output request, Abaqus/CAE writes field output data from only the top and bottom of a composite layup, and no data are generated from the other plies. To create an envelope plot that examines all of the plies in your model, you must create a new output request or edit the default output request and specify the composite layup and the plies and section points from which variables will be output.

In many cases, even though you change the position in the ply at which Abaqus/CAE checks the value, the same ply experiences the extreme value, and the contour plot does not change. However, the contour plot might change because the critical ply changed if:

- you have a small number of plies, and the results are changing rapidly through the thickness of the ply; or
- the material is nonlinear, and its stiffness changes abruptly under some conditions.

Through thickness X–Y plots

After you have analyzed a composite layup and determined which regions contain the critical plies, you can view the behavior of the plies across the layup with a through thickness X–Y plot. You can create an *X–Y* data object by reading field output results from the section points in a selected element of a shell region of your model. If you select an element in a composite layup, Abaqus/CAE plots data for each section point in each ply across the entire thickness of the layup. *Figure 6* illustrates a through thickness plot of the strain in the fiber direction through 13 plies of a composite layup. The strain is discontinuous because the orientation of the fiber changes between plies.

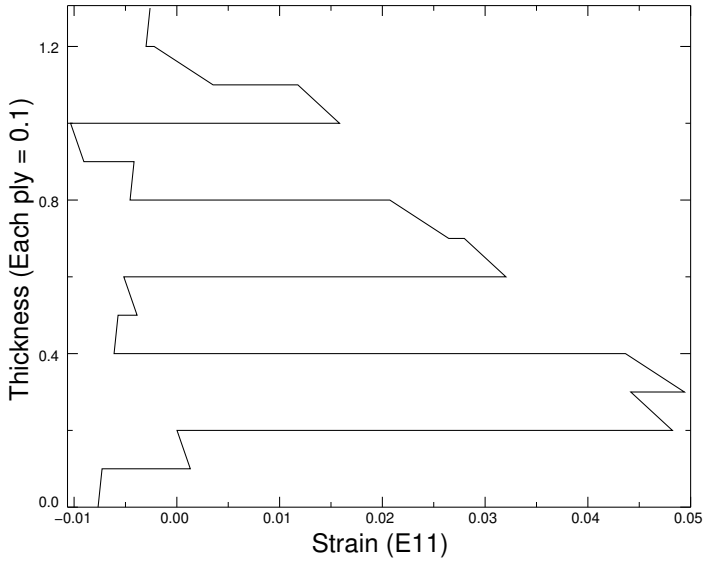


Figure 6: A through thickness X–Y plot.

For more information, see *Reading X–Y data through the thickness of a shell*.

Color coding

You can use color coding in all of the Abaqus/CAE modules to change the color of individual layups and plies. If you choose to color code by plies, Abaqus/CAE displays only one ply in a region, which by default is the last ply (in alphabetical order). To view a different ply, you can deactivate selected plies. For more information, see *Coloring geometry and mesh elements*.

Display groups

In the Visualization module you can use display groups to view the elements in only selected layups or plies. For more information, see *Selection methods for creating or editing a display group*.

Using a composite layup to model a yacht hull, illustrates how you can create and analyze a complex three-dimensional composite model and how you can use Abaqus/CAE to view the behavior of individual plies in the layup.

Connectors

This section provides information on how to model connectors.

In this section:

- *Modeling connectors*
- *What is a connector?*
- *What is a connector section?*
- *What is a CORM?*
- *What are connector behaviors?*
- *Creating the connector geometry, connector sections, and connector section assignments*
- *What is the relationship between reference points and connectors?*
- *Defining connector orientations in connector section assignments*
- *Requesting output from connectors*
- *Applying connector loads and connector boundary conditions*
- *Displaying connectors and connector output in the Visualization module*

Modeling connectors

The procedure for modeling and using connectors in Abaqus/CAE involves the following general steps:

1. Create reference points and datum coordinate systems to use when modeling connectors.
2. Create assembly-level wire features.
3. Create connector sections to define the connection types, connector behavior, and section data.
4. Create a connector section assignment that associates the connector section with the wires that you select and that specifies the orientations for the first and second points of the wires that you select.
5. In the Load module, prescribe connector loads and connector boundary conditions to simulate connector actuations and constrain material flow.
6. In the Step module, create field and history output requests for connectors.
7. In the Visualization module, plot connector output results; control the display of connector section assignments, wire endpoints, connection types, and current local orientations; and animate the time history of wire endpoints and local orientations.

What is a connector?

Connectors allow you to model mechanical relationships between two points in an assembly. The connection can be simple, such as a link, or the connection can impose more complicated constraints, such as constant velocity joints. The connector geometry is modeled using an assembly-level wire feature that contains one or more wires. The wires may connect two points in an assembly or connect a point to ground. You create a connector section that specifies the type of connection and connector behaviors, such as spring-like elasticity behavior. To complete the connector modeling, you create a connector section assignment that associates a connector section with the wires that you select and that specifies the orientations for the first and second points of the wires that you select. For more information on connectors, see *About Connectors*.

For example, *Figure 1* illustrates the crank mechanism that is used in the example problem *Crank mechanism*.

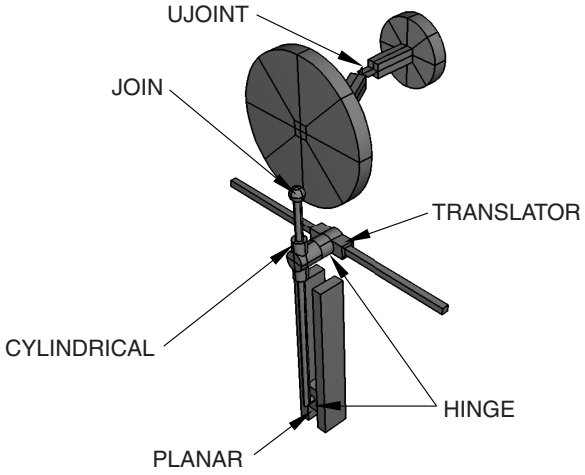


Figure 1: A crank mechanism modeled with connectors.

The model transmits a rotational motion through two universal joints and then converts the rotation into translational motion of two slides. An Abaqus Scripting Interface script that reproduces the crank mechanism model using Abaqus/CAE is provided with this example. The crank mechanism is modeled using nine part instances attached to each other through eight connectors modeled in Abaqus/CAE.

What is a connector section?

A connector section defines the connection type and may include connector behavior and section data. Abaqus provides several connection types—basic types, assembled types, complex types, and MPC types.

Basic types

Basic connection types include translational types and rotational types. Translational types affect translational degrees of freedom at both endpoints of the wires to which the connector section is assigned and may affect rotational degrees of freedom at the first or both endpoints of the wires. Rotational types affect only rotational degrees of freedom at both endpoints of the wires.

Assembled types

Assembled connection types are predefined combinations of basic connection types. For example, a HINGE connection type combines a JOIN connection type (translational) and a REVOLUTE connection type (rotational) to join the position of two points and provide a revolute constraint between their rotational degrees of freedom.

Complex types

Complex connection types affect a combination of degrees of freedom in the connection and cannot be combined with other connection types. They typically model highly coupled physical connections.

MPC types

MPC connection types are used to define multi-point constraints between two points.



Note:

Abaqus/CAE writes complete connector section data to the input file only when a connector section has a connector section assignment in the model database. If you plan to import a model from an input file and obtain connector sections, you must make sure that all of the connector sections have assignments in the model database.

For an overview of the connection types that are available in Abaqus/CAE, see *Understanding connector sections and functions*. For a description of each connection type and the equivalent basic connection types that define the kinematic constraints of assembled type connections, see *Connection Types* and *General Multi-Point Constraints*.

What is a CORM?

CORM is an abbreviation for component of relative motion: relative displacements and rotations local to a connector. Available components of relative motion are relative displacements and rotations that are not kinematically constrained. Depending upon the connection type, some of the components may be available and some may be constrained. When you are creating or modifying a connector section, Abaqus/CAE displays the available and constrained components of relative motion in the editor for the specified connection type. In addition to applying behaviors to the components of relative motion, you can prescribe connector loads and connector boundary conditions to the available components of relative motion of a connector (see *Applying connector loads and connector boundary conditions*). For more information on creating connector sections, see *Connector section editors*.

What are connector behaviors?

After you specify a connection type for a connector section, you can define behaviors for the components of relative motion. You can specify the following connector behaviors:

- Elasticity
- Damping
- Friction
- Plasticity
- Damage
- Stop
- Lock
- Failure
- Reference length
- Integration (Abaqus/Explicit analyses only)

For more information on behaviors, see *Connector behaviors*, and *Connector Behavior*.

Creating the connector geometry, connector sections, and connector section assignments

You create the assembly-level wire features, connector sections, and connector section assignments in the Interaction module to model a connector. Abaqus/CAE provides two methods for modeling connectors: you can use the **Connector Builder** to perform all the steps involved in creating a single connector, including creation of the wire feature, connector section, connector section assignment, and any desired reference points and datum coordinate systems; or you can create multiple connectors by creating the wires, connector sections, and connector section assignments in separate dialog boxes in the Interaction module. If you choose the latter technique, you should create the desired reference points and datum coordinate systems before you begin modeling the connectors.

The Model Tree, shown in *Figure 1*, is helpful for understanding the organization of the reference points, datum coordinate systems, and wire features that you created in assembly-related modules, as well as connector sections and connector section assignments that you created in the Interaction module. You can use the Query toolset in the Interaction module to obtain connector assignment information for selected wires.

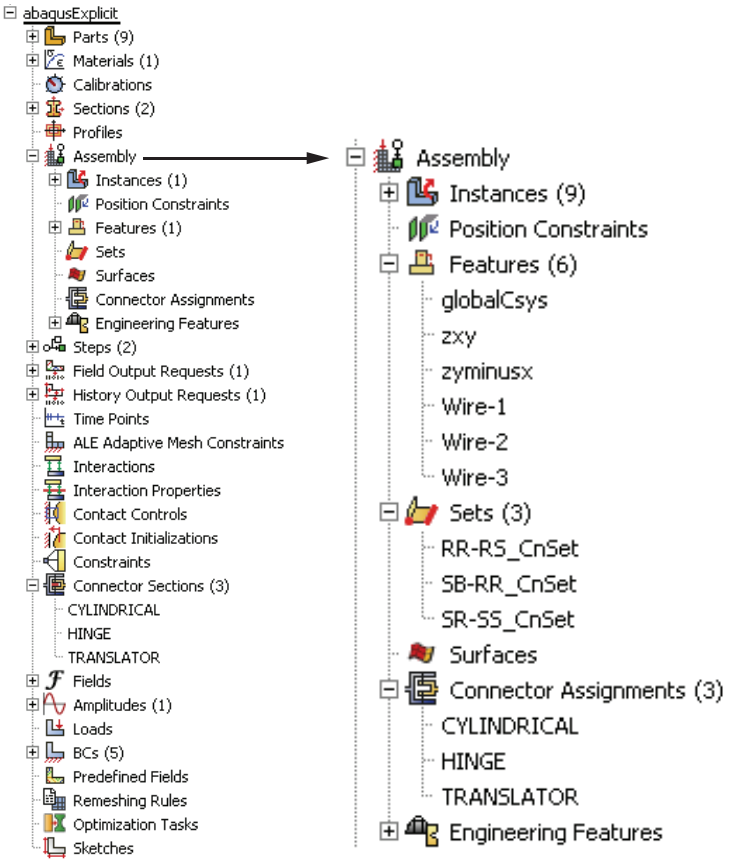


Figure 1: Wire features, connector sections, and connector section assignments in the Model Tree.

For detailed instructions, see the following sections:

- *Selecting a process for defining connector geometry*
- *Creating a single connector*
- *Creating or modifying wire features for multiple connectors*
- *Creating connector sections*
- *Creating and modifying connector section assignments*

- *Using the Query toolset to obtain connector assignment information*

What is the relationship between reference points and connectors?

The points that you can use to define an assembly-level wire feature can be one of the following:

- Vertex or node of a part instance
- Reference point of a part instance or of an assembly
- Ground

Using a reference point is often more convenient than using a point or node on the assembly because the same reference point can be used by a rigid body or coupling constraint. In the Part module you can use the Reference Point toolset to create one reference point on each part; additional reference points can be created in the Assembly module, Interaction module, and Load module.

You can use the Model Tree to rename reference points with more descriptive feature names. Descriptive names allow you to more easily associate reference points and wires in a complex model, as shown in *Figure 1*.

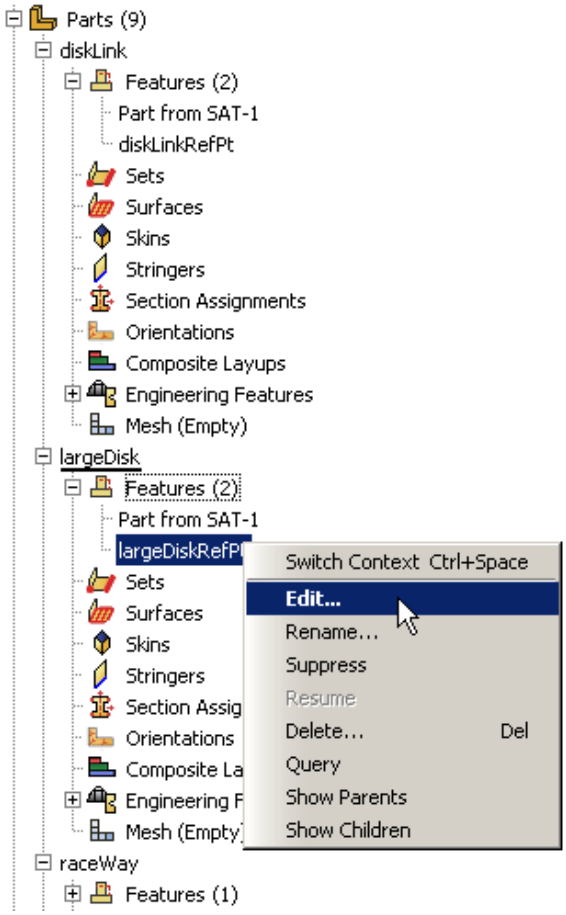


Figure 1: The Model Tree for the crank mechanism.

If you want to use different feature names and labels for a reference point on multiple instances of the same part, you must create the reference points in an assembly-related module. You can create a node on a mesh using the Edit Mesh toolset and then select the node to define an endpoint of an assembly-level wire; however, the node is not listed in the Model Tree and has no label in the viewport. For more information on creating and renaming reference points, see *The Reference Point toolset*. For more information on creating a node on a mesh, see *Manipulating nodes*.

Each reference point or node that defines an endpoint of an assembly-level wire needs to be associated with the model geometry, even if they are created in the Part module. In many cases you will create a rigid body constraint or a coupling

constraint (to distribute forces to an area instead of to a single point) between the reference point or node and a part instance. As a result, the motion of the wire endpoint will be constrained to the motion of the part instance. You create constraints in the Interaction module. For more information, see *Understanding constraints*.

Defining connector orientations in connector section assignments

When you model a connector, you may need to specify the orientation of the connector. Orientations for a connector may be required, optional, or not applicable, depending upon the connection type. In most cases the orientation will not be the same as the global coordinate system, and you must create a datum coordinate system that defines the connector's local orientation before you create a connector section assignment. Connector orientation requirements are described in *Connection Types*.

You can create datum coordinate systems in the Part module, Property module, Assembly module, Interaction module, Load module, and Mesh module. You can name a datum coordinate system when you create it. When you create a connector section assignment, you can select a datum coordinate system from the current viewport or you can select from a list of coordinate system names in a dialog box. Again, the Model Tree is helpful for understanding the organization of your datum coordinate systems. Datum coordinate systems are described in *Methods for creating a datum coordinate system*.

Requesting output from connectors

You can request field and history output for connectors by selecting a set that contains wires as the domain from which the output will be generated. In the Step module choose **Sets** as the domain type in the field output request editor or history output request editor; then select a set containing wires from the list of sets available. For more information, see *Creating and modifying output requests*.

Applying connector loads and connector boundary conditions

In the Load module you can apply a connector force or connector moment to the available components of relative motion of a connector to simulate connector actuation. Similarly, you can prescribe a connector displacement, connector velocity, or connector acceleration for the available components of relative motion of a connector. When you create these connector loads or boundary conditions, you select the wires to which you want to apply the prescribed condition. The best approach for selecting wires is to use the default geometry set name for the wire feature (see *Creating or modifying wire features for multiple connectors*, for more information). The wires that you select must be associated with a connector section assignment. If you select multiple wires, you must ensure that the connector sections assigned to the wires in the connector section assignments have the available components of relative motion for which you want to define forces or moments. If there are insufficient available components of relative motion for the connector force or connector moment, a message appears asking you to select different wires or to change the connection type.

You can also apply a connector material flow boundary condition to the endpoints of wires that are associated with connector section assignments. For more information, see *Creating loads*, and *Creating boundary conditions*.

Displaying connectors and connector output in the Visualization module

Abaqus/CAE displays connectors modeled using connector section assignments in the Visualization module. You can control the display of connectors using display groups. Each connector is listed as an element set in the output database. For more information on display groups, see *Creating or editing a display group*.

You can also use the **Entity Display** options in the **ODB Display Options** dialog box to control the display of symbols that represent connectors. You can control the following:

- The highlighting of wire endpoints
- The display of local orientation axes for connectors
- The display of connector type labels
- The size of the displayed symbols

For information on the symbols that represent connectors modeled in Abaqus/CAE, see *Understanding symbols that represent interactions, constraints, and connectors*. For more information on controlling the display of the symbols, see *Controlling the display of model entities*.

For example, you can use display groups to display four part instances and three connector element sets of the crank mechanism, as shown in *Figure 1*.

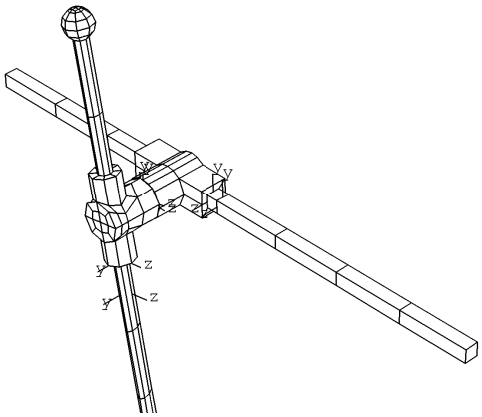


Figure 1: Selected part instance, connector, and connector symbol display in the Visualization module for the crank mechanism.

The highlighting of the wire endpoints and the display of connector type labels have been toggled off so that only the local connector orientations are displayed. You can produce a time history animation to view the changes in the connector orientations during the mechanism operation. For more information, see *Time history animation*.

For information on plotting connector output results, see *Contouring analysis results*, *Plotting analysis results as symbols*, and *X–Y plotting*.

The kinematics of connection types is formulated using an intrinsic coordinate system. The basis vectors of the intrinsic coordinate system are aligned with the directions associated with the components of relative motion. For some connection types (such as the CARTESIAN connection type), the intrinsic coordinate system aligns with the orientation directions at node *a* (see *Connection Types* for more information on connector orientation directions).

The components of connector vector output are resolved with respect to different coordinate systems depending on whether the output requested is field output or history output. For connector field output, the components of the vector are resolved with respect to the orientation directions at node *a*. However, if either of the two nodes is ground, the output of the axial connector is CTF1. For connector history output and connector field outputs with the "_LOCAL" suffix (for example, CTF_LOCAL, CTM_LOCAL), the components of the vector are resolved with respect to the intrinsic coordinate system. Therefore, except for the connection types whose intrinsic coordinate systems align with

the orientation directions at node *a*, plots of field output (symbol plots and contour plots) and history plots display different values for the requested connector vector components; the resultant is not affected by the choice of the coordinate system used to resolve the components.

Additional information

- *Connectors*
- *Understanding contour plotting*
- *Understanding symbol plotting*

Continuum shells

This section provides information on how to model continuum shells.

In this section:

- *Modeling continuum shells*
- *Meshing parts with continuum shell elements*

Modeling continuum shells

You use conventional shell parts to model structures in which the thickness is significantly smaller than the other dimensions, and you define the thickness in the Property module when you create the section.
In contrast, you assign continuum shell elements to solid parts, and Abaqus determines the thickness from the geometry of the part. From a modeling point of view continuum shell elements look like three-dimensional continuum solids, but their kinematic and constitutive behavior is similar to conventional shell elements. For example, conventional shell elements have displacement and rotational degrees of freedom, while continuum solid elements and continuum shell elements have only displacement degrees of freedom. For more information, see *About Shell Elements* and *Choosing a Shell Element*. *Figure 1* illustrates the differences between a conventional shell and a continuum shell element.

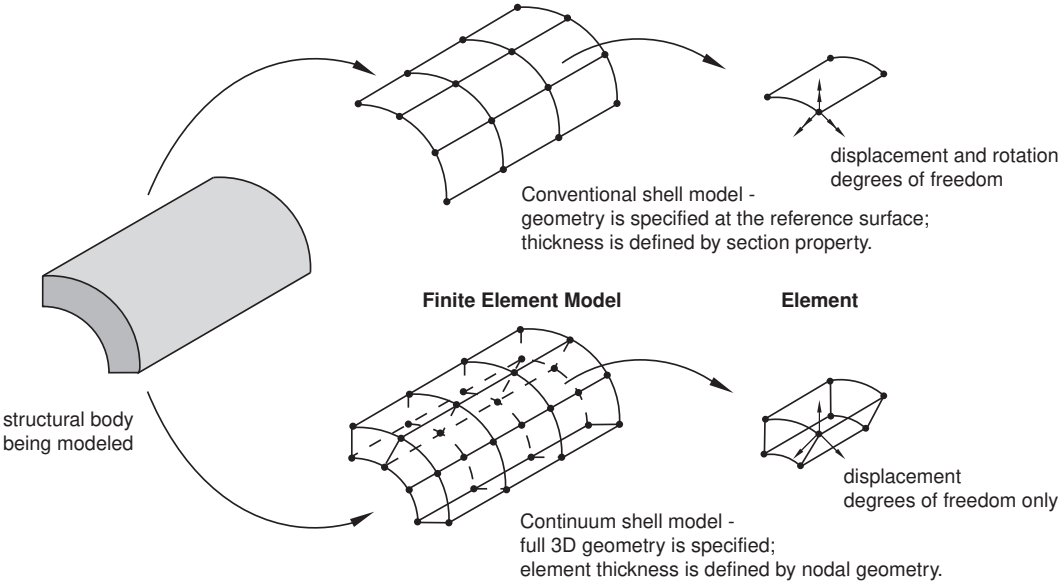


Figure 1: Conventional versus continuum shell element.

The general procedure for modeling continuum shells in three-dimensional space involves the following steps:

1. In the Part module, define the solid geometry.
2. In the Property module, assign a shell section to any solid regions to which you will assign continuum shell elements in the Mesh module. You must specify the thickness of a shell section; however, Abaqus uses this thickness only to estimate certain section properties, such as hourglass stiffness. Abaqus uses the actual thickness, based on the element nodal geometry, during the analysis. If the thickness of the solid region varies along its length, you should provide an approximate value of the thickness. For more information, see *Using a Shell Section Integrated during the Analysis to Define the Section Behavior*.
3. In the Mesh module, query the mesh stack orientation. If necessary, assign a stack orientation so that the continuum elements are aligned consistently from the bottom to the top of the stack. See *Applying a mesh stack orientation*, for more information.
4. In the Mesh module, assign a continuum shell element type to the region, and mesh the region with hexahedral or wedge elements. These are the only elements that can be stacked to form a continuum shell mesh.

Meshing parts with continuum shell elements

You use continuum shell elements to model shell-like solids with greater accuracy than conventional shell elements, as described in *About Shell Elements*. In addition, although you model a continuum shell with hexahedral- or wedge-shaped elements, the element formulations are still more computationally efficient than solid continuum elements.

When you are generating elements that will be assigned to continuum shell elements, the elements in the mesh must be oriented consistently. For example, *Figure 1* shows a swept mesh generated in the direction of the sweep path. The generated elements are stacked in the direction of the sweep path; however, if you plan to use continuum shell elements, the elements should be stacked through their thickness.

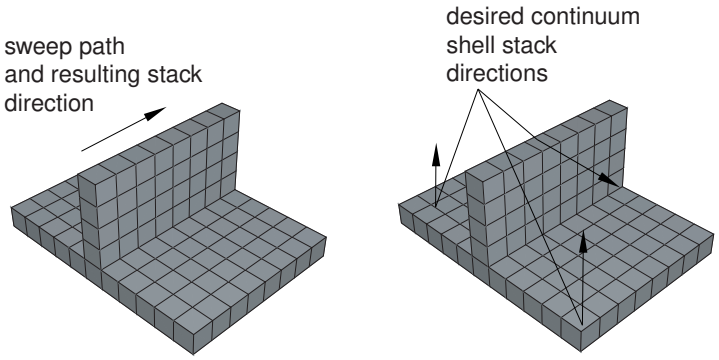


Figure 1: The resulting stack direction is not correct for continuum shell elements.

You can use the Query toolset to determine which faces are designated as the bottom face and the top face and to look for inconsistent orientations between elements. For more information, see *Obtaining general information about the model*. In some cases, you can partition the model and change the direction of the sweep path to obtain the correct orientation. Alternatively, you can assign an orientation unrelated to the sweep path. For more information, see *Applying a mesh stack orientation*.

Co-simulation

This section explains how to model and run a co-simulation in Abaqus/CAE.

In this section:

- *Overview of co-simulation*
- *What is a co-simulation?*
- *Linking and excluding part instances for a co-simulation*
- *Ensuring matching nodes at the interface regions*
- *Specifying the interface region and coupling schemes*
- *Identifying the models involved and specifying job parameters*
- *Viewing the results of the co-simulation*

Overview of co-simulation

The procedure for modeling and running a co-simulation in Abaqus/CAE involves the following general steps:

1. Create the models in a single model database.
2. Optionally, link part instances between models and exclude the linked instances from the analyses.
3. Optionally, ensure matching nodes at the interface regions.
4. In each model, create a co-simulation interaction to specify the interface region and coupling schemes.
5. Create a co-execution to identify the two models involved and specify the job parameters for each analysis.
6. Submit the co-execution to perform the co-simulation.
7. View the results of the co-simulation using overlay plots.

What is a co-simulation?

The co-simulation technique is a multiphysics capability that provides run-time coupling of Abaqus analysis programs. You can divide a model into multiple domains and use different analysis programs to obtain solutions for each domain. For Abaqus/Standard to Abaqus/Explicit co-simulation, each Abaqus analysis operates on a complementary section of the model domain where it is expected to provide the more computationally efficient solution. For example, Abaqus/Standard provides a more efficient solution for light and stiff components, while Abaqus/Explicit is more efficient for solving complex contact interactions.

You define the interface region across which fields are exchanged and the coupling schemes. For more information, see *Structural-to-Structural Co-Simulation*.

Additional information

• *Preparing an Abaqus Analysis for Co-Simulation*

Linking and excluding part instances for a co-simulation

For a co-simulation you may want to see the “whole” model involved in the co-simulation in a single viewport. To achieve this, you can link part instances in one model to part instances in the other model. For example, you can link part instances from the Abaqus/Standard model to part instances in the Abaqus/Explicit model. In this case, the linked part instances in the Abaqus/Standard model cannot be edited, their position is determined solely by the position of the part instance in the Abaqus/Explicit model, and they are available only for display purposes. Any changes to the part instances in the Abaqus/Explicit model are automatically updated in the linked part instances. You must exclude the linked part instances from the appropriate analysis; for example, the part instances in the Abaqus/Explicit model that are linked to the Abaqus/Standard model must be excluded from the Abaqus/Explicit analysis. For more information, see *Linking part instances between models*, and *Excluding part instances from an analysis*.

Ensuring matching nodes at the interface regions

You may have dissimilar meshes in regions shared in the model definitions. For Abaqus/Standard to Abaqus/Explicit co-simulation, you can improve solution stability and accuracy in some cases by ensuring that you have matching nodes at the interface (see *Dissimilar Mesh-Related Limitations*). This section describes the recommended modeling practices for ensuring matching nodes at the interface regions.

In general, you will create a skin or a stringer (depending on whether the interface region is a face or an edge) on the part that contains the interface region in the Abaqus/Explicit model, perform a variety of modeling techniques, and obtain a part instance to use to define a tie constraint in the Abaqus/Standard model.

Detailed instructions are provided in the following procedure.

1. In the Abaqus/Explicit model:
   a. In the Property module display the part that contains the interface region. If the interface region is a face, create a skin on the face. If the interface region is an edge, create a stringer on the edge. For more information, see *Skin and stringer reinforcements*.
   b. If the part is geometry based, mesh the part.
   c. Create a mesh part (even if you are working with a mesh part).
   d. Delete all of the elements in the newly created mesh part other than those on the skin or stringer. In addition, delete the associated unreferenced nodes using the Edit Mesh toolset.
2. In the Abaqus/Standard model:
   a. Copy the mesh part containing the skin or stringer from the Abaqus/Explicit model, and create an instance of the newly copied part.
   b. To simplify region selection procedures, create a named set or surface that contains the mesh part.
   c. In the Interaction module, create a tie constraint specifying the copied mesh part (using the named set or surface) as the main region and the interface region on the Abaqus/Standard model as the secondary region.
   d. In the Interaction module, define a Standard-Explicit co-simulation interaction and specify the mesh part (using the named set or surface) as the interface region. For more information, see *Specifying the interface region and coupling schemes*.
3. In the Abaqus/Explicit model:
   a. Delete the mesh part that contains the skin or stringer.
   b. Delete the skin or stringer from the part geometry.
   c. If the part is geometry based, remesh the part.
   d. In the Interaction module, define a Standard-Explicit co-simulation interaction and specify the interface region in the original Abaqus/Explicit part as the interface region.
4. Continue with the co-simulation procedure as described in *Overview of co-simulation*.

Specifying the interface region and coupling schemes

In each model, you define a co-simulation interaction to specify the interface region (region for exchanging data) and coupling schemes for the co-simulation. Only one co-simulation interaction can be active in each model, and the settings in each co-simulation interaction must be the same in each model. For more information, see *Defining a Standard-Explicit co-simulation interaction*.

Identifying the models involved and specifying job parameters

For co-simulation, the analyses are executed in synchronization with one another using the same functionality as that used for executing a single analysis job. In the Job module, you create a co-execution to identify the two analysis jobs involved in the co-simulation and to specify job parameters for each analysis, and you submit the co-execution to submit both jobs for analysis. For more information, see *Creating, editing, and manipulating co-executions*.

Viewing the results of the co-simulation

To display the results from the co-simulation, you can use the overlay plot functionality to display data from both output databases in the same viewport. For more information, see *Overlaying multiple plots*.

Display bodies

This section provides information on how to model display bodies.

In this section:

- *What is a display body?*
- *Should I mesh a display body?*
- *Displaying display bodies in the Visualization module*

What is a display body?

A display body is a part instance that will be used for display only. The part instance can be any instance containing geometry, mesh elements, or a combination of geometry and mesh. You do not have to mesh the part, and the part is not included in the analysis; however, when you view the results of the analysis, the Visualization module displays the part along with the rest of your model. In effect, a display body provides a more realistic view of your model in the Visualization module without the computational expense of including the part instance in the analysis. A display body behaves like a rigid part and does not deform. You cannot apply prescribed conditions, such as constraints, loads, or boundary conditions, to a display body.

You can associate the motion of a display body with selected control points (one point or three points), or you can specify that the display body remain fixed during the analysis. If the display body follows a single point, the display body will translate and rotate based on the translations and rotations of the single point. If the display body follows three points, the display body will translate and rotate based on the translation of the three points. For more information, see *Display Body Definition*.

For example, *Figure 1* shows a backhoe arm modeled with connectors.

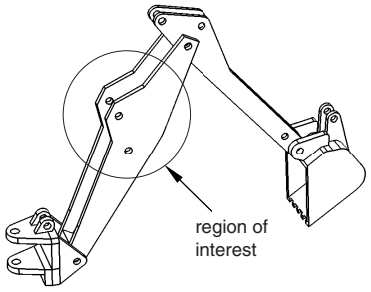


Figure 1: A backhoe arm modeled with connectors.

The region of interest is the main arm of the backhoe. The bucket interacts with the rest of the backhoe model only through the connectors and the mass and inertia of the bucket. As a result, the bucket can be modeled as a display body.

You create a display body by applying a display body constraint to a part instance. For the backhoe arm model you would first create the bucket part instance in the Part module. In addition, you would create a point part and position it at the centroid of the bucket in the assembly, as shown in *Figure 2* (see *Point parts*); a reference point is automatically created for the point part.

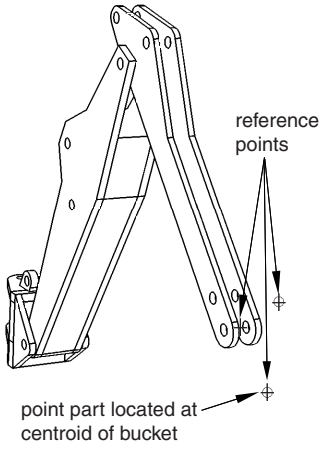


Figure 2: The mass and inertia of the bucket are modeled with a point part.

The other two reference points in the assembly serve as endpoints for a HINGE and an AXIAL connector; see *Connectors* for more information on connectors. In the Interaction module you would create a rigid body constraint to constrain the endpoints and the point part to move as a single rigid entity, as shown in *Figure 3*; see *Defining rigid body constraints*.

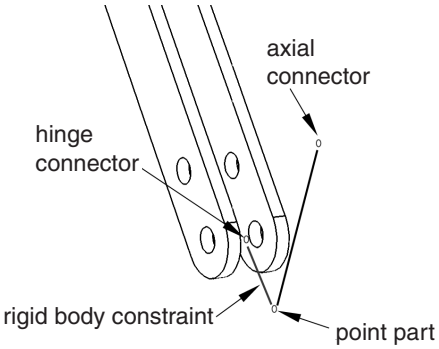


Figure 3: A rigid body constraint constrains the point part to the endpoints.

In addition, you would create a display body constraint to constrain the bucket part instance to the point part; see *Defining display body constraints*.

You do not have to mesh the bucket part instance, and it is not included in the analysis of the backhoe model. However, the bucket still appears when you view the results of the analysis in the Visualization module, as shown in *Figure 4*.

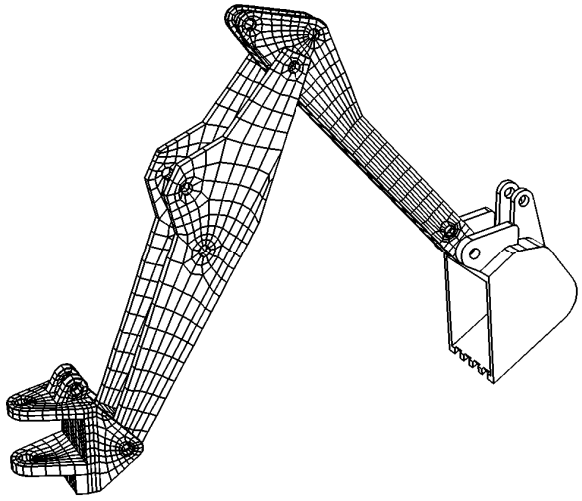


Figure 4: The bucket is modeled as a display body.

Display bodies are useful if your model contains a part that cannot be meshed and need not be analyzed. In general, an imported part that has invalid geometry cannot be used by Abaqus/CAE unless you choose to ignore the invalidity of the part (for more information, see *Working with invalid parts*). If you apply a display body constraint to the invalid part, you can continue to use the part in your model, although the part will not be included in the analysis. For more information, see *What is a valid and precise part?*.

Should I mesh a display body?

You do not have to mesh a display body. However, to display the part instance in the Visualization module, Abaqus/CAE computes a triangular mesh that models the surfaces of the part instance. This internal mesh is written to the input file and to the resulting output database; however, it is used only to render the display and is not included in the numerical analysis. By default, the Visualization module displays only the free edges of the meshed display body part instance, although you can change the visible edges displayed.

Similarly, if you do mesh a display body or if you use a mesh part to create a display body, Abaqus/CAE retains your mesh when the input file is generated; however, it is used only to render the display in the Visualization module and is not included in the numerical analysis.

Displaying display bodies in the Visualization module

You can use the **Display Body Options** in the Visualization module to customize the appearance of display bodies. The options available for display bodies are similar to the common and superimposed plot options in Abaqus/CAE: render style; edge visibility, color, and style; fill color; scaling; and translucency can all be customized. Similarly, display body options are plot state–independent; i.e., the options applied to display bodies are reflected in all plot states. For more information, see *Customizing the appearance of display bodies*.

Display body options apply to all display bodies in your model; to customize the appearance of individual display bodies, you can create display groups based on part instances (see *Creating or editing a display group*). Individual item coloring can also be applied to individual display body part instances to override the edge and/or fill colors specified as general display body options; see *Customizing the display color of individual objects* for more information.

Eulerian analyses

This section explains how to create Eulerian models in Abaqus/CAE.

In this section:

- *Overview of Eulerian analyses*
- *Assembling coupled Eulerian-Lagrangian models in Abaqus/CAE*
- *Defining contact in Eulerian-Lagrangian models*
- *Assigning materials to Eulerian part instances*
- *Using the volume fraction tool*
- *Eulerian mesh motion*
- *Viewing output from Eulerian analyses*

Overview of Eulerian analyses

Pure Eulerian analysis is a finite element technique in which materials are allowed to flow across element boundaries in a rigid mesh.
In the more traditional finite element formulation (also known as the Lagrangian technique), materials are closely associated with an element, and the materials move only with the deformation of the mesh. Because the element quality issues associated with a deformable mesh are not present in Eulerian analyses, the Eulerian technique can be very effective at treating problems involving very large deformations, material damage, or fluid materials. Eulerian analysis can be performed only in **Dynamic, Explicit** steps. For a detailed discussion of Eulerian capabilities and applications, refer to *Eulerian Analysis*.

The techniques for modeling pure Eulerian analyses in Abaqus/CAE are very different than those used to model pure Lagrangian analyses. Most notably, instead of defining several parts and assembling them into a complete model, Eulerian models typically consist of a single Eulerian part. This part, which can be arbitrary in shape, represents the domain within which Eulerian materials can flow. The geometry of bodies in the model is not necessarily defined by the Eulerian part; instead, materials are assigned to different regions within the Eulerian part instance to define the body geometries.

Figure 1 compares the same model created using Lagrangian and Eulerian techniques. In the Lagrangian model, two parts are modeled, and each part is assigned a unique section referencing a material. In the Eulerian model, a single Eulerian part is modeled and assigned an Eulerian section. The Eulerian section defines the materials that can be present within the part. Materials are then assigned to different regions within the instanced part; any region without a material assignment is treated as a void with no material properties.

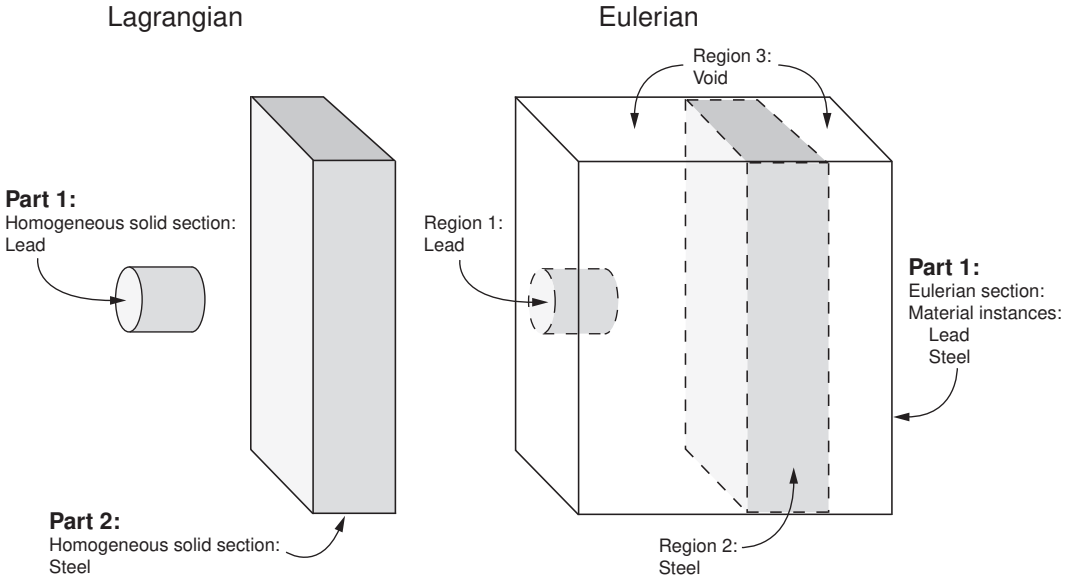


Figure 1: Two bodies modeled using Lagrangian and Eulerian techniques.

The Eulerian analysis technique can be coupled with traditional Lagrangian techniques to extend the simulation functionality in Abaqus:

- Arbitrary Lagrangian-Eulerian (ALE) adaptive meshing is a technique that combines features of Lagrangian and Eulerian analysis within the same part mesh. ALE adaptive meshing is typically used to control element distortion in Lagrangian parts undergoing large deformations, such as in a forming analysis. Most ALE adaptive meshing analyses can also be performed as pure Eulerian analyses, but some of the features associated with a Lagrangian mesh will be lost; for a more detailed comparison, see *About ALE Adaptive Meshing*.
- Coupled Eulerian-Lagrangian (CEL) analysis allows Eulerian and Lagrangian bodies within the same model to interact. Coupled Eulerian-Lagrangian analysis is typically used to model the interactions between a solid body

and a yielding or fluid material, such as a Lagrangian drill traveling through Eulerian soil or an Eulerian gas inflating a Lagrangian airbag. Eulerian-Lagrangian analysis is discussed in *Assembling coupled Eulerian-Lagrangian models in Abaqus/CAE*.

Viewing the results of Eulerian analyses requires some special techniques in the Visualization module. These techniques are discussed in detail in *Viewing output from Eulerian analyses*.

The procedure for creating Eulerian models in Abaqus/CAE involves the following general steps:

1. In the Part module, create an Eulerian-type part that defines the geometric region within which Eulerian materials can flow. For more information, see *Choosing the type of a new part*.
2. In the Part module, you may want to create partitions that represent the initial boundaries between different materials in the part. The partitions will affect the mesh of the part, and they are necessary only if you are assigning materials uniformly across a region. For more information about assigning materials in an Eulerian part, see *Assigning materials to Eulerian part instances*, and *Defining a material assignment field*.
3. In the Property module, define the materials in the model.
4. In the Property module, define and assign an Eulerian section for the model. The Eulerian section determines which materials can be present in the Eulerian part. The topology of the materials within the part will be defined in the Load module, as discussed in Step 7. For more information, see *Creating Eulerian sections*.
5. In the Assembly module, create an instance of the Eulerian part.
6. In the Step module, create a field output request for output variable EVF. This output is necessary to view the deformation of materials in an Eulerian model. For more detailed information, see *Viewing output from Eulerian analyses*.
7. In the Load module, create a material assignment predefined field that defines the topology of materials in the initial configuration of the Eulerian part instance. For more information, see *Assigning materials to Eulerian part instances*, and *Defining a material assignment field*.
8. In the Load module, define any loads or boundary conditions acting on the model. Because the mesh in an Eulerian part is rigid, traditional Lagrangian prescribed conditions do not move with the material deformations; loads and boundary conditions are imposed on any material that occupies (or moves into) the region to which the condition is prescribed. Zero-displacement boundary conditions can be used along the sides of an Eulerian part instance to prevent Eulerian material from entering or exiting the part. Boundary conditions and constraints based on nonzero nodal displacements are ignored in an Eulerian part instance; typically, velocity boundary conditions or predefined fields are used to prescribe initial motion in an Eulerian model. Specialized Eulerian boundary conditions can also be defined to control the flow of material across the boundaries of the Eulerian part (see *Defining an Eulerian boundary condition*). For more information about applying loads and boundary conditions to Eulerian models, see *Boundary Conditions*.
9. In the Mesh module, create a hexagonal mesh for the Eulerian part. EC3D8R elements are assigned to the mesh by default. After creating a regular mesh, you can trim any elements that are not expected to experience material flow to reduce the model size and improve performance of the analysis.

An example of a basic Eulerian model created in Abaqus/CAE is provided in *Eulerian analysis of a collapsing water column*. More complicated Eulerian modeling procedures, including complex material assignments and coupled Eulerian-Lagrangian interactions, are illustrated in *Impact of a water-filled bottle*.

Assembling coupled Eulerian-Lagrangian models in Abaqus/CAE

Abaqus allows you to create models that include both Eulerian part instances and Lagrangian part instances. During a coupled Eulerian-Lagrangian analysis, the Lagrangian mesh interacts with the materials in the Eulerian part. Coupled Eulerian-Lagrangian analyses typically offer better interpretation of contact conditions than pure Eulerian analyses, particularly for interactions between fluid and solid materials. During postprocessing, a solid Lagrangian body in a coupled Eulerian-Lagrangian analysis appears to maintain its shape better than a similar body in a pure Eulerian analysis.

Figure 1 compares a pure Eulerian analysis (top) and a coupled Eulerian-Lagrangian analysis (bottom) of a steel brick passing through a column of water.

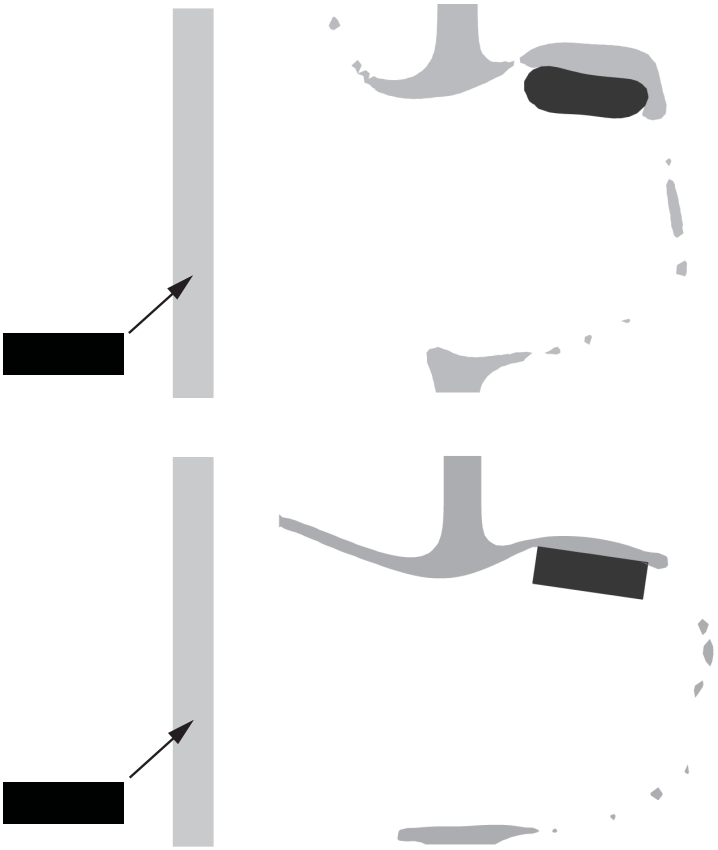


Figure 1: Comparison of a pure Eulerian analysis (top) and a coupled Eulerian-Lagrangian analysis (bottom).

In the Eulerian analysis the water tends to adhere to the sides of the brick, and the brick appears deformed in the final configuration. This apparent deformation is a result of the material volume fraction calculations used for Eulerian materials, as discussed in *Material Interfaces*. On the other hand, the Lagrangian brick maintains its shape as it passes through the Eulerian water, and the water flows around the brick.

Coupled Eulerian-Lagrangian analyses also allow you to capitalize on the strengths of both analysis techniques; for example, you can use the loads on a Lagrangian body moving through an Eulerian material to drive a detailed submodel of the Lagrangian body.

To assemble a coupled Eulerian-Lagrangian model in Abaqus/CAE, simply instance both Eulerian and Lagrangian parts in the same assembly. Coupled Eulerian-Lagrangian analyses can be performed only in **Dynamic, Explicit** steps. You must create a general contact definition to enable contact between Lagrangian and Eulerian parts. The general contact definition allows interactions between Lagrangian surfaces and Eulerian material instances in the model (see

Defining contact in Eulerian-Lagrangian models for more information). Other interactions, loads, boundary conditions, and predefined fields are applied to the Lagrangian and Eulerian parts in the usual manner.

In most cases the Lagrangian part is assembled inside of the Eulerian part instance. While Lagrangian and Eulerian elements and nodes can overlap, three-dimensional Lagrangian elements cannot occupy the same space as an Eulerian material instance. Therefore, Lagrangian parts must be instanced in an area of void within the Eulerian part instance (i.e., a region with no material assignment). To model a three-dimensional Lagrangian part instance that is completely surrounded by Eulerian material, use the volume fraction tool to create an Eulerian material assignment field that includes a void region corresponding to the Lagrangian part instance (see *Using the volume fraction tool* for details).

Defining contact in Eulerian-Lagrangian models

Contact in a coupled Eulerian-Lagrangian analysis must be defined using general contact. The general contact definition enables interactions between Lagrangian part surfaces and the surfaces of Eulerian material instances; contact between different Lagrangian parts in the model is also enabled.

In Abaqus/CAE you can use the global **All* with self** contact domain to enforce contact between all Lagrangian parts and all Eulerian material instances in the model. Alternatively, you can include or exclude contact between a Lagrangian surface and a particular Eulerian material instance. Eulerian material instances appear in the list of surfaces in the **Include Pairs** and **Exclude Pairs** dialog boxes, as shown in *Figure 1*. You can also assign unique contact properties between particular Lagrangian surfaces and particular Eulerian material instances in the **Individual Contact Property Assignments** dialog boxes.

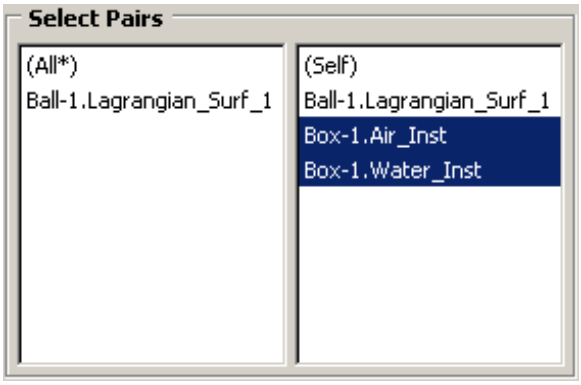


Figure 1: Eulerian material instances in the Edit Excluded Pairs dialog box.

True contact cannot be defined between different Eulerian material instances. A rudimentary contact condition is enforced because material instances cannot penetrate each other. However, material instances do not separate once they come into contact, which prevents the modeling of sliding or rebounding behavior. Contact output is not available for Eulerian material instances. For a detailed discussion of Eulerian analysis contact formulations, refer to *Interactions*. If the contact conditions between two materials are significant to your analysis, at least one of the materials should be modeled as a Lagrangian part.

Assigning materials to Eulerian part instances

In a pure Lagrangian analysis a section definition includes a reference to a single material. When you assign a section to a region or element in the Lagrangian part, that region or element is completely filled with the referenced material. The geometry of the region or element, therefore, defines the geometry of the material.

In a pure Eulerian analysis the relationship between the section definition and material is fundamentally different. An Eulerian section definition can reference a list of materials. When you assign the Eulerian section to an Eulerian part, you are defining which materials may be present in the part over the course of the analysis. The part, however, is initially empty of material. To introduce material to the initial state of an Eulerian part, you must use a material assignment predefined field.

Material assignment predefined fields rely on the concept of material volume fractions. During an Eulerian analysis, Abaqus tracks the material present in each element in terms of a volume fraction assigned to each material instance; the volume fraction represents the percentage of the element's volume that is occupied by a given material instance. For elements that are partially filled or filled with multiple materials, the exact geometric composition of the material within the element is not known; Abaqus interpolates the material volume fractions from adjacent elements to estimate the material boundaries within the element. These calculations are discussed in more detail in *Material Interfaces*.

In Abaqus/CAE the initial material volume fractions in an Eulerian part are specified by creating a material assignment predefined field in the Load module. The predefined field associates each region in an Eulerian part instance with a volume fraction for each material instance. The regions to which volume fractions are assigned can be cells (in geometry), mesh elements, or groups of elements. If you select a cell or a group of elements, the volume fraction values are propagated to each of the underlying Eulerian elements in the cell or group.

Volume fractions in a material assignment predefined field are expressed as a number between zero and one; a volume fraction of one indicates that the region is completely filled with the specified material. A volume fraction of less than one indicates that the region is only partially filled with the specified material; for example, a volume fraction of 0.25 means that the specified material instance occupies 25% of the region. As mentioned previously, Abaqus determines the material boundaries for partially filled elements based on the volume fractions in adjacent elements; to achieve greater control over the material boundaries within a region, you must refine the part mesh or redefine the region boundaries.

If material volume fractions are not defined for a region of an Eulerian part instance, that region is assigned a void. Similarly, if the volume fractions for all materials in a region do not sum to one, the remainder of the volume fraction in that region is assigned a void. Void regions do not have material properties, but other materials can flow into and through a void region during an analysis.

The material assignment predefined field effectively defines the topology of materials in the initial configuration of your model. The Eulerian part is typically arbitrary in shape; the material assignment predefined field adds to the part the Eulerian materials that will interact during the analysis. For example, consider the cross-section of the coupled Eulerian-Lagrangian model in *Figure 1*. The Eulerian part is simply an empty cube. Four regions defined on the part determine the slope of the earth and the amount of water in the tank, and material is assigned to these regions accordingly.

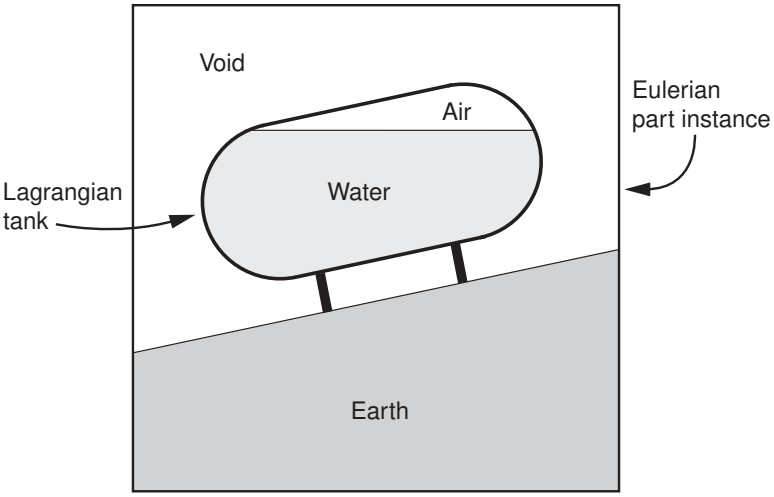


Figure 1: Material assignments in an Eulerian-Lagrangian model.

Material assignment predefined fields can be created only in the initial step of an Eulerian analysis. In subsequent steps the materials deform from their initial configuration and flow across the Eulerian mesh according to the forces present in the model.

Abaqus/CAE offers two techniques for defining material assignment predefined fields:

Uniform definitions

Uniform material assignment field definitions are created by selecting regions from an Eulerian part instance and directly specifying the volume fraction of each material instance within those regions. Geometry must be partitioned into separate cells representing the material regions. In part instances that include orphan elements you can select individual elements to act as regions.

Figure 2 illustrates a material assignment field created using uniform definitions. The Eulerian part is partitioned into three regions, and material volume fractions are defined in each region; each region is completely filled with a single material instance. Volume fractions are not defined for the void region, since void is the default material assignment.

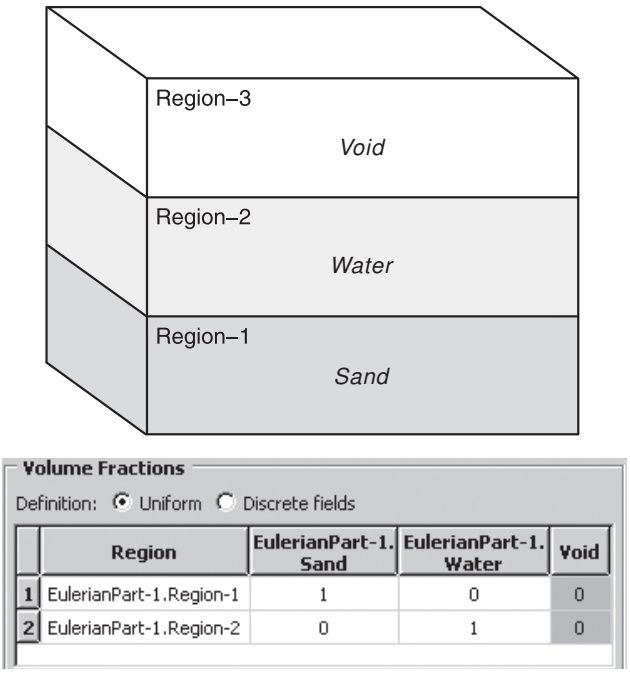


Figure 2: A uniform material assignment field.

The uniform material assignment definition should be used only for relatively simple regions that are uniformly filled with material. The partitions needed to create complex regions can negatively impact the quality of the Eulerian mesh, and partially filled regions are difficult to define and interpret, particularly when working with geometry.

For details on creating a uniform material assignment field in Abaqus/CAE, see *Defining a material assignment field*. An example of a uniform material assignment definition is illustrated in the Python script provided in *Deflection of an elastic dam under water pressure*.

Discrete field definitions

Material assignments for meshed geometry and orphan meshes can be defined using a scalar discrete field. For each material instance in the part you create a discrete field that associates individual elements with a volume fraction for that material instance. For more information on creating discrete fields, see *The Discrete Field toolset*.

When you are assigning materials using a discrete field, you still must select the region of the part instance to which the discrete field applies. If the discrete field includes data for elements outside of the selected region, these data are ignored. The default value associated with the discrete field is assigned to any elements within the selected region that are not explicitly listed in the discrete field.

Figure 3 shows a very simple example of a material assignment defined using discrete fields. The Eulerian part consists of four elements and two material instances. Two discrete fields, as defined in *Table 1*, are used to specify the material composition within the elements. The boundary between the water and the sand is an estimation based on interpolation of the material volume fractions in adjacent elements.

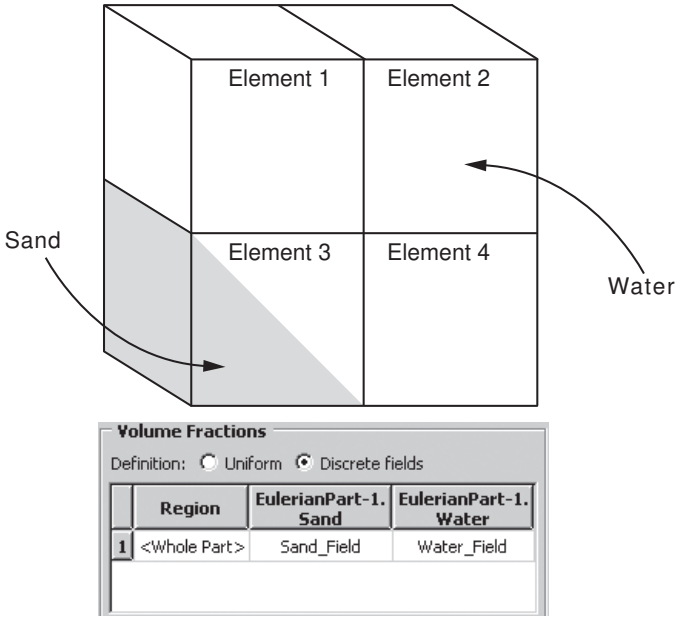


Figure 3: A discrete field material assignment.

Table 1: Volume fractions defined in discrete fields.

Discrete Field	Element 3	Default
Water_Field	0.5	1
Sand_Field	0.5	0

Note:

In any given element the sum of all material volume fractions should not be greater than one. Abaqus/CAE assigns volume fractions incrementally by reading the discrete fields in the **Volume Fractions** table from right to left; once the volume fraction for an element reaches one, additional volume fractions assigned to that element are ignored.

Since the discrete field can assign unique volume fractions to each individual element, it allows more complicated material boundaries than the uniform definition method without the need for excessive partitioning. The volume fraction tool in Abaqus/CAE creates discrete fields specifically for use in material assignment predefined fields. Through this tool, you can define complex Eulerian material regions using the part modeling techniques available in Abaqus/CAE. For more information, see *Using the volume fraction tool*.

For details on creating a discrete field material assignment in Abaqus/CAE, see *Defining a material assignment field*. An example of a discrete field material assignment definition (including the use of the volume fraction tool) is illustrated in the Python script provided in *Rivet forming*.

Using the volume fraction tool

The volume fraction tool creates a scalar discrete field by performing a Boolean comparison between an Eulerian part instance and a second part instance (the reference part instance) that intersects the Eulerian instance.
The comparison determines where the two part instances overlap, then assigns each element in the Eulerian instance a volume fraction based on the percentage of the element that is also occupied by the reference instance. The volume fraction is specified as a decimal between zero and one.

The discrete field that is created by the volume fraction tool can be used to assign material instances to the Eulerian part instance (see *Assigning materials to Eulerian part instances*). The topology of the assigned Eulerian material instance corresponds to the shape of the reference part instance within the Eulerian part instance.

Figure 1 and the following procedure summarize the process of using the volume fraction tool:

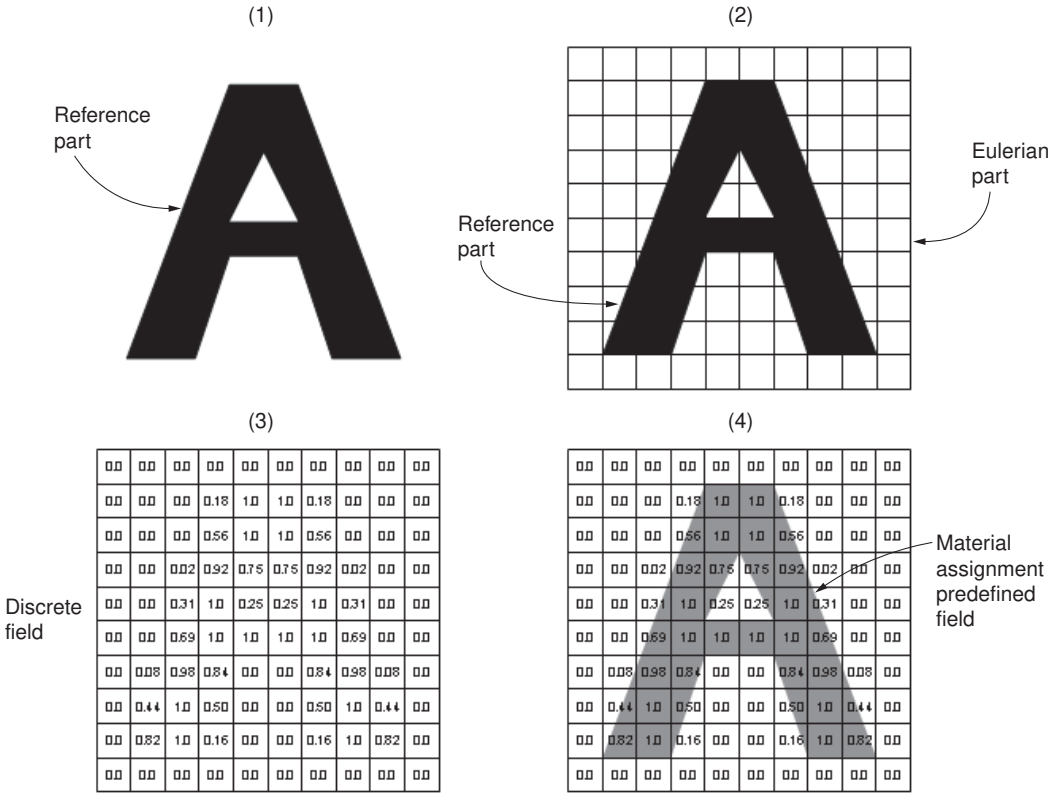


Figure 1: Procedure for using the volume fraction tool.

1. Using any of the modeling tools and techniques in Abaqus/CAE, create a reference part that corresponds to the geometry of the desired Eulerian material region.
2. Instance the reference part within the Eulerian part instance. The reference part instance should spatially correspond to the desired Eulerian material region.
3. Use the volume fraction tool to create a discrete field based on a comparison of the reference part instance and the Eulerian part instance.
4. Define a material assignment predefined field for the Eulerian part instance using the discrete field created by the volume fraction tool.

An option for the volume fraction tool controls whether the calculated discrete field represents the space inside the reference instance (the volume fraction is nonzero in elements that overlap the reference instance) or the space outside

the reference instance (the volume fraction is nonzero in elements that do not overlap or partially overlap the reference instance), as depicted in *Figure 2*.

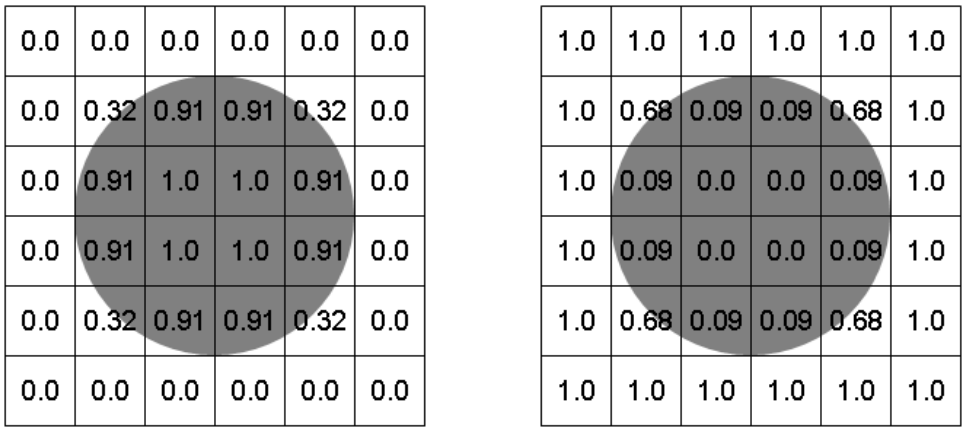


Figure 2: Discrete fields representing volume fractions inside (left) and outside (right) of the shaded reference region.

Typically, calculating the volume fraction outside of a reference instance is used to create an Eulerian material assignment around a Lagrangian part instance in a coupled Eulerian-Lagrangian analysis. Calculating the volume fraction inside of a reference instance is usually used to model complex material assignment fields within a pure Eulerian part instance; in this situation, the reference instance is suppressed after the Eulerian material is assigned. You can also calculate volume fractions inside of a reference instance to create an Eulerian material assignment on the inside of an enclosed Lagrangian shell in a coupled Eulerian-Lagrangian analysis.

To use the volume fraction tool, select **Tools->Discrete Field->Volume Fraction Tool** from the main menu bar. For step-by-step instructions for using the tool, refer to *Creating discrete fields for material volume fractions*.

The Eulerian part instance that is used by the volume fraction tool must be the same instance to which materials are assigned. The Eulerian part must be meshed before using the volume fraction tool. You should not edit the part mesh after creating the discrete field, as the element numbering in the discrete field may not conform to the elements in the updated mesh.

The reference part instance used by the volume fraction tool can include unmeshed geometry, a native mesh, or orphan elements; deformable, Eulerian, and discrete rigid parts are all allowed. The volume fraction tool always uses the mesh representation (if available) of the reference part instance when calculating the discrete field; if the reference instance is partially meshed, only the meshed portion of the instance is considered in the volume fraction calculation. The mesh on the reference instance should be fine enough to capture all important geometric details in the subsequent material assignment.

The reference part instance must be either a three-dimensional solid or a fully enclosed three-dimensional shell. The faces of a shell enclosure must define a single, continuous surface; features that create T-intersections with the surface faces (such as ribs or interior dividing panels) are not allowed (see *Figure 3*). An Eulerian element is considered to be inside a shell reference instance if it lies within the volume enclosed by the shell surface.

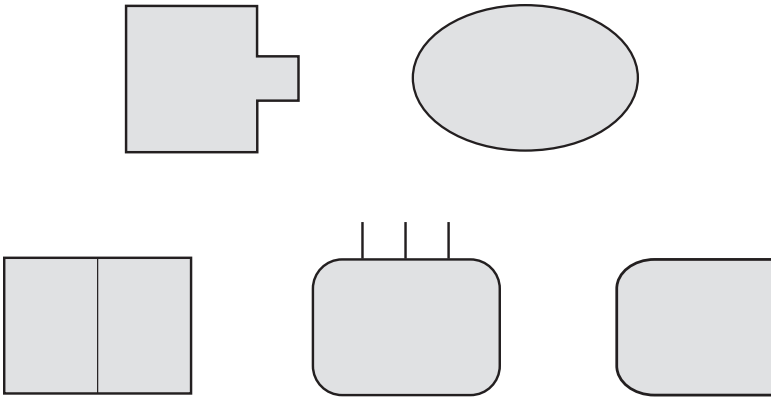


Figure 3: Cross-sections of acceptable reference shell parts (top) and unacceptable reference shell parts (bottom).

Eulerian mesh motion

Eulerian mesh motion is a technique that allows you to reduce the size of an Eulerian mesh for certain models, thereby improving the performance of the analysis. In some cases the Eulerian domain at the beginning of an analysis does not sufficiently capture the deformations in the model by the end of the analysis. Some typical examples include:

- An airbag where the inflator gas is modeled as an Eulerian material: the gas initially occupies a small region, but the region quickly expands as the airbag inflates.
- A projectile impact analysis where the projectile is modeled as an Eulerian material: the projectile initially occupies a region that is far from its ultimate destination.

In such cases it is possible to adjust the size and location of the Eulerian domain during the analysis so that it always captures a part or material of interest. By default, an Eulerian mesh is rigid and fixed in place; but enabling Eulerian mesh motion allows the Eulerian elements to scale and translate during the analysis. Eulerian mesh motion can follow the deformation of an Eulerian material instance (as shown in *Figure 1*) or a Lagrangian surface (as shown in *Figure 2*). The behavior of Eulerian mesh motion is described in detail in *Eulerian Mesh Motion*.

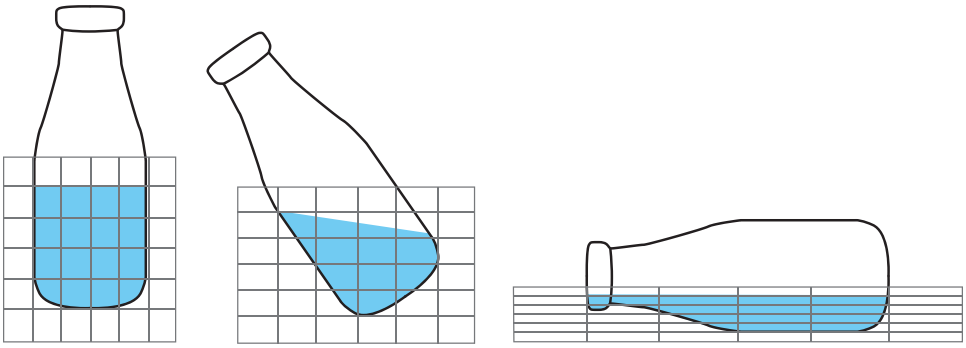


Figure 1: Eulerian mesh motion tracking a material instance.

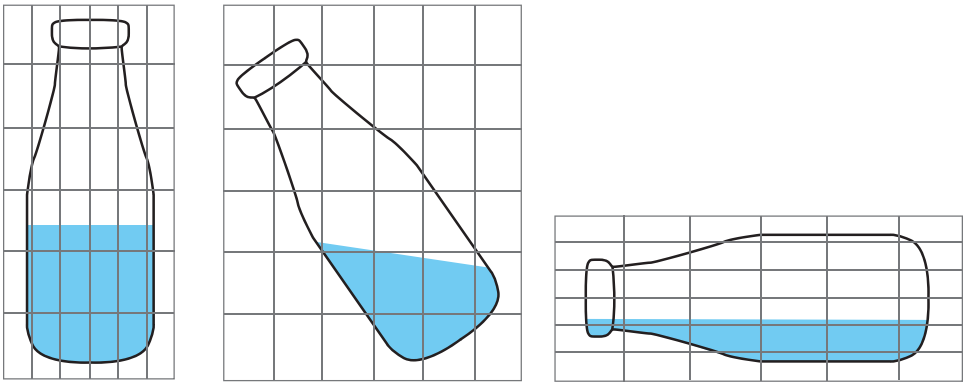


Figure 2: Eulerian mesh motion tracking a Lagrangian surface.

In Abaqus/CAE Eulerian mesh motion is defined as a boundary condition in the Load module. Restrictions can be imposed on the scaling and translation of the Eulerian mesh as part of the boundary condition definition. For more information about creating Eulerian mesh motion boundary conditions, see *Defining an Eulerian mesh motion boundary condition*.

Viewing output from Eulerian analyses

The results of an Eulerian analysis must be interpreted differently than those from a Lagrangian analysis. In particular, any results based on nodal displacements are meaningless in an Eulerian model because the Eulerian part is fixed and rigid. Details concerning how Eulerian results are written to the output database are available in *Output*.

Special steps must be taken in the Visualization module of Abaqus/CAE to view material instances within an Eulerian part. By default, Abaqus/CAE displays the full Eulerian part mesh in both undeformed and deformed plot states with no indication of the material instance boundaries within the mesh.

Visualization of material instances is based on output variable EVF, the Eulerian material volume fraction. Output variable EVF measures the amount of a particular material instance within an element as a relative fraction. An EVF value of one indicates that the element is completely filled with the specified material instance; an EVF value of zero indicates that the element is completely devoid of the specified material instance.

For elements that are partially filled or filled with multiple materials, Abaqus estimates a simple boundary between materials by interpolating the EVF values in adjacent elements. These simple boundaries may be slightly discontinuous across elements. To improve display of Eulerian materials, you should instruct Abaqus/CAE either to use a results averaging threshold of 100% or to compute scalars after averaging results; Abaqus/CAE remaps the material boundaries so they appear smooth and continuous across elements. For more information about results averaging, see *Controlling result averaging*; for a more detailed discussion of how Abaqus calculates Eulerian material boundaries, see *Material Interfaces*.

Output variable EVF is written to the output database if you request the **Preselected defaults** in the field output request editor (see *Modifying field output requests*). When you request output for EVF, Abaqus creates a separate material volume fraction output variable for each material instance in the model; for example, `EVF_WATER` is the volume fraction for the material instance named `Water`. An output variable named `EVF_VOID` is created to measure the volume fraction of empty regions in an Eulerian part.

The following techniques can be used in the Visualization module to view the initial and deformed states of material in an Eulerian part:

Contour plots

A contour plot of output variable EVF for a particular material instance allows you to visualize which areas of the model are occupied by the material during the analysis. Areas occupied by the material (EVF equal to one) appear as a uniform color from the top of the contour spectrum, while areas unoccupied by the material appear as a different color from the bottom of the contour spectrum; depending on your contour plot settings, the boundary of the material instance appears in a range of colors as EVF transitions from one to zero (see *Figure 1*).

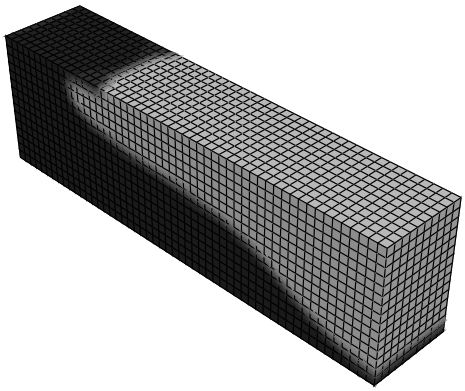


Figure 1: Contour plot of an Eulerian material instance.

Contour plots are of limited usefulness when visualizing Eulerian materials because the contours appear on the faces of the Eulerian parts. You cannot effectively visualize material volume fraction contours on the interior of Eulerian parts.

For further details on using contour plots in the Visualization module, see *Contouring analysis results*.

View cuts

To visualize the behavior of a material on the interior of an Eulerian part, activate a view cut along an isosurface of the EVF variable associated with that material instance. Abaqus/CAE automatically creates these isosurface view cuts for each material instance in the model, but you must activate them in the **View Cut Manager**. Using the view cut options, you can eliminate portions of the part that do not include a selected material by rendering them unfilled, rendering them translucent, or removing them from the display (see *Figure 2*).

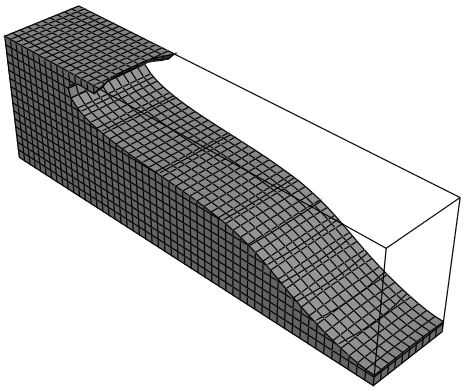


Figure 2: View cut along an Eulerian material instance isosurface.

If your Eulerian part includes regions without a material assignment, it may be helpful to activate an isosurface view cut based on the `EVF_VOID` output variable. By cutting away all regions in which `EVF_VOID` is greater than `0.5`, you are able to see the shape of materials within the part.

After activating an isosurface view cut based on the EVF variable, you can change the primary field output variable without affecting the view cut. This enables you to visualize results contours along the boundaries of material instances instead of on Eulerian part faces (see *Figure 3*).

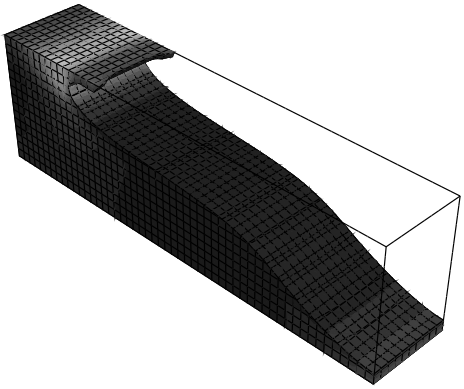


Figure 3: Contour plot of stresses in a cut Eulerian part.

Isosurface view cuts based on output variable EVF do not affect Lagrangian part instances in a coupled Eulerian-Lagrangian model. The Lagrangian parts remain visible when the cut is active. Therefore, this technique is useful for visualizing the interaction between a Lagrangian part and an Eulerian material instance.

For further details on using view cuts in the Visualization module, see *Cutting through a model*.

Combining view cuts and contour plots

In an Eulerian part that includes three material instances, you can use a combination of view cuts and contour plots to distinguish material instances in the undeformed and deformed model states. First, use an isosurface view cut to remove one of the material instances from the display, as discussed above. Then, create a contour plot of output variable EVF for one of the remaining material instances. The resulting colors in the model distinguish one material from the other. To produce a more defined boundary between the materials, you can reduce the number of contour intervals to two.

For example, *Figure 4* depicts an Eulerian model of a lead projectile impacting a brass plate. The void regions of the Eulerian part are cut away. A two-interval contour plot is applied, rendering the brass in one color and the lead (i.e., not brass) in another color. The resulting plot offers a useful generalization of the deformed shape of the projectile and plate.

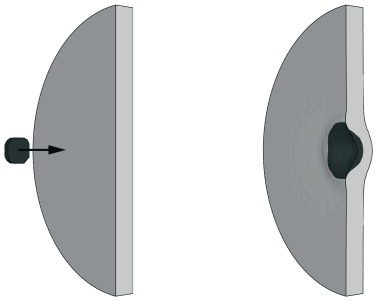


Figure 4: A contour plot is used to distinguished two Eulerian materials in a projectile impact analysis.

Currently there is no way to visually distinguish more than three Eulerian material instances simultaneously using Abaqus/CAE.

Color coding

Color coding cannot be used to visualize material behavior in Eulerian parts. The color coding tool in Abaqus/CAE does not recognize Eulerian section or material assignments. Color coding based on element sets is also ineffective for deformed shape plots because the Eulerian elements do not deform with the material.

However, in coupled Eulerian-Lagrangian models, color coding can distinguish between Eulerian and Lagrangian part instances or Eulerian and Lagrangian element types. When used with the visualization techniques discussed above, color coding can be helpful in distinguishing Lagrangian bodies from Eulerian materials in a model.

For further details on applying color coding to a model, see *Color coding geometry and mesh elements*.

Display groups

Certain types of coupled Eulerian-Lagrangian models involve a single Eulerian material instance throughout the Eulerian part; for example, a Lagrangian penetrator moving through a uniform Eulerian material. In these analyses the deformation of the Eulerian material is not as important as the interaction between the Eulerian material and the Lagrangian body. You can use a display group to remove the Eulerian elements from the display and visualize results (such as contact pressure or stress) on only the Lagrangian body.

For further details on using display groups, see *Using display groups to display subsets of your model*.

Fasteners

This section describes how to model fasteners.

In this section:

- *About fasteners*
- *Managing fasteners*
- *Creating point-based fasteners*
- *Creating discrete fasteners*
- *Creating assembled fasteners*

About fasteners

You use fasteners to model a point-to-point connection between two or more faces, such as a spot weld, a bolt, or a rivet.
You can use the Attachment toolset to create attachments that help you define fasteners; for more information, see *Understanding attachment points and lines*. You can create point-based fasteners, discrete fasteners, or assembled fasteners.

Both point-based and discrete fasteners can be modeled using connectors or beam MPCs. If you define connectors that use basic, assembled, or complex connection types, you can request output from your point-based or discrete fasteners through connector output variables. However, if you use beam MPCs, no output is available from your point-based or discrete fasteners.

In this section:

- *About point-based fasteners*
- *About discrete fasteners*
- *About assembled fasteners*

About point-based fasteners

Point-based fasteners make use of positioning points to create mesh-independent fasteners in Abaqus/Standard and Abaqus/Explicit, as described in *Mesh-Independent Fasteners*. A positioning point can be an attachment point, a reference point, or a node from an orphan mesh. Point-based fasteners modeling spot welds around the edge of a bracket are shown in *Figure 1*. The user first created attachment points at equally spaced intervals around the edge of the bracket. The attachment points were used to define the location of the fastener's positioning points.

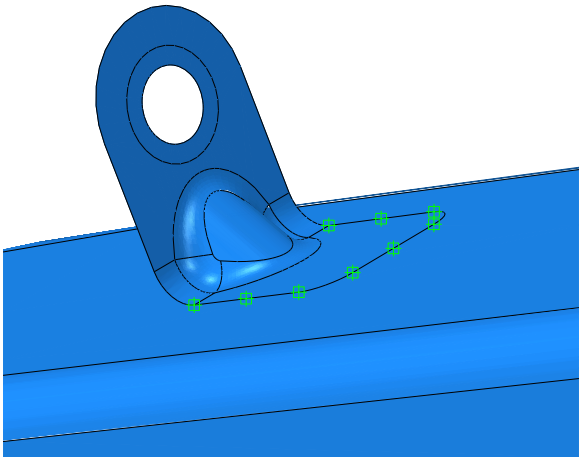


Figure 1: Point-based fasteners modeling spot welds.

A point-based fastener can connect selected faces with either connectors or rigid (beam) multi-point constraints. If you want to model a rigid connection, you can use rigid connectors or rigid multi-point constraints.

Connectors

If you use connectors to connect the faces, you can model either rigid, elastic, or inelastic connections with failure by using the generality of connector behavior definitions. You can use a rigid connector to model a rigid connection. However, if you are using rigid connectors and Abaqus detects two adjacent point-based fasteners that are sharing nodes, you must avoid overconstraining your model by defining some elasticity in the connector behavior to reduce the stiffness.

You can request output from connectors.

Rigid (beam) multi-point constraints

Rigid multi-point constraints are computationally cheaper than connectors and are less likely to result in an overconstrained model when two adjacent fasteners are sharing nodes. When Abaqus detects two adjacent fasteners that are sharing nodes and using rigid multi-point constraints, it uses a penalty distributing coupling formulation that relaxes, to a small degree, the constraint between the motion of the fastening point and its coupling nodes to avoid the overconstraint.

You cannot request output from multi-point constraints.

A point-based fastener uses distributing coupling constraints to connect the faces regardless of how you mesh the faces. When you submit a job for analysis, Abaqus uses your fastener definition to connect the faces with couplings and connectors. When you open the output database file in the Visualization module, you can display the couplings and connectors; however, outside the Visualization module, symbols are displayed only for the positioning points of the point-based fasteners.

If your model contains many fasteners (more than a thousand), point-based fasteners offer better performance than discrete fasteners. In addition, you may want to use point-based fasteners if you have a file from a CAD system that defines the coordinates of each positioning point. Point-based fasteners are available only for three-dimensional models. Point-based fasteners are used to model mesh-independent fasteners as described in *Mesh-Independent Fasteners*.

You can use the following two methods to create a point-based fastener, as shown in *Figure 2*:

- Select a positioning point and allow Abaqus/CAE to project the point to the closest face along a normal to the surface. The first fastening point is created where the normal intersects the closest face.
- Select a positioning point and specify the direction vector along which the point is projected. The first fastening point is created where the vector intersects the closest face.

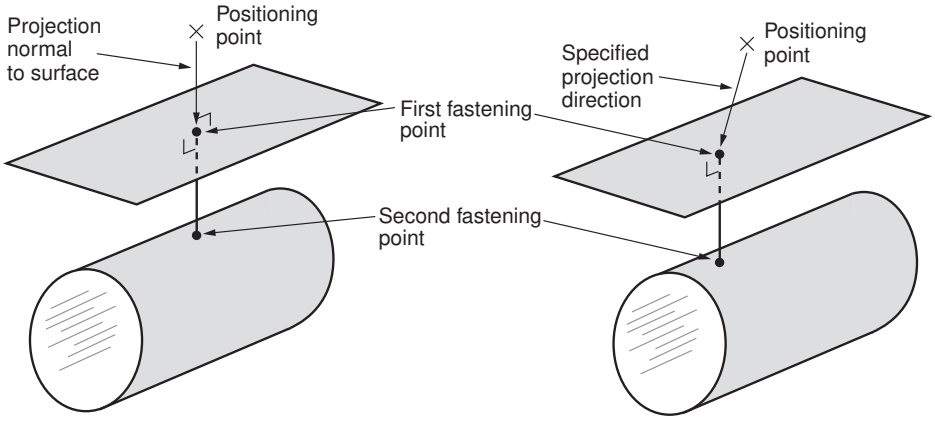


Figure 2: Creating point-based fasteners.

Abaqus creates the second (and subsequent) fastening points when you submit the job for analysis by projecting the first fastening point onto the other surfaces to be connected along the normal to the closest face.

About discrete fasteners

Discrete fasteners make use of attachment lines to create connectors and couplings between selected faces in Abaqus/Standard and Abaqus/Explicit. *Figure 1* shows a discrete fastener that uses attachment lines and connectors (**Beam** connection type) to model a spot weld across three surfaces. The user first created attachment lines at the locations of the spot welds. The attachment lines were used to define the location of the discrete fastener.

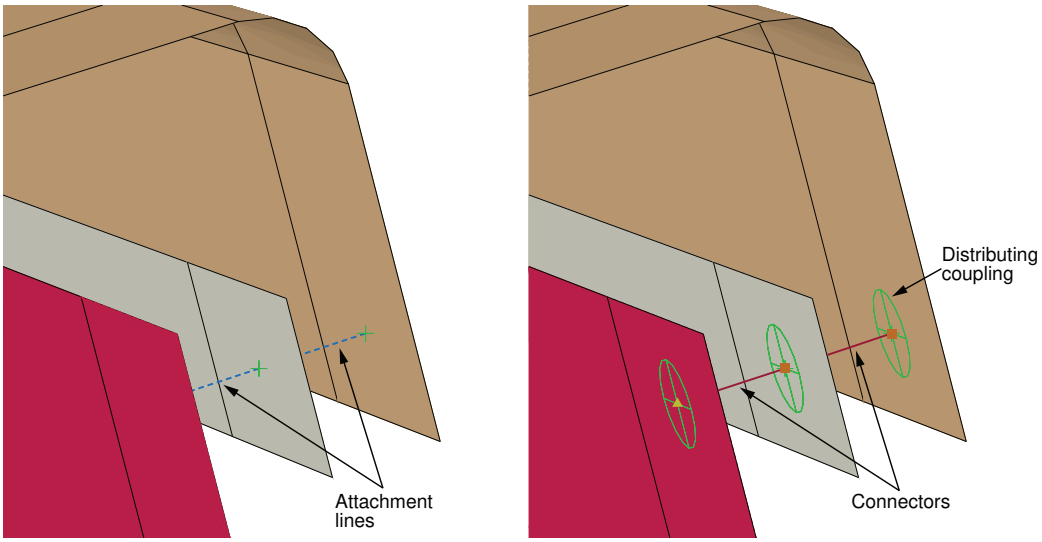


Figure 1: A spot weld across three surfaces modeled with a discrete fastener.

The process of creating an attachment line is similar to the process of creating point-based fasteners. You select a starting point and specify which of the two methods Abaqus/CAE will use to project the point onto the closest face. *Figure 2* shows the two methods for creating an attachment line that is used to define a discrete fastener.

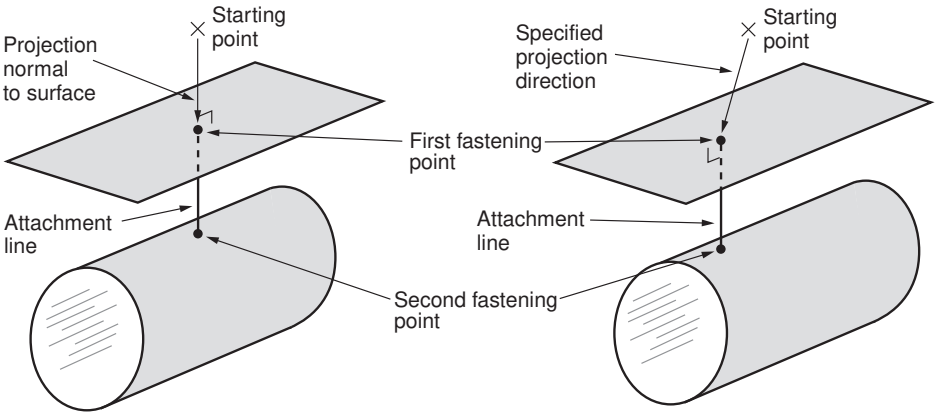


Figure 2: Creating an attachment line that is used to define a discrete fastener.

Abaqus/CAE projects the attachment line along a normal to the closest face. You can use the following two methods to determine the number of faces that are connected by the attachment line:

- Specify the number of projections or layers.
- Specify the maximum length of the projection.

For each attachment line, Abaqus/CAE determines the fastening points and applies a distributing coupling between each fastening point and its corresponding surface. After you assign a connector section to the attachment line, Abaqus/CAE creates a connector, and the discrete fastener is considered to be fully defined. You can control the

accuracy of the attachment line position on a faceted representation of a part instance by specifying the level of curve refinement in the Part module and the Property module. For more information, see *Controlling curve refinement*.

If your model is complex, Abaqus/CAE can create a chain of attachment lines connecting multiple surfaces that would be time consuming to create manually. In contrast with point-based fasteners, you can view attachment lines and discrete fasteners and their connectors and couplings outside the Visualization module.

If two surfaces are used by two attachment lines and share a common face, Abaqus/CAE merges the two faces into a single face when you submit the model for analysis. This results in better performance by Abaqus/Standard or Abaqus/Explicit, especially when the fasteners connect nodes across faces of a refined orphan mesh.

When you submit a job that contains a model with discrete fasteners for analysis, Abaqus/CAE writes special comment lines to the input file. These special comment lines, which are ignored by the Abaqus solvers, allow Abaqus/CAE to recreate the fully defined discrete fasteners upon import into Abaqus/CAE. For more information, see *Importing interactions, constraints, and fasteners*.

An example of creating discrete fasteners is illustrated in the Python script included in *Buckling of a column with spot welds*.

About assembled fasteners

Assembled fasteners let you efficiently assign complex fastener behaviors in a large number of model locations.

In this section:

- *The assembled fastener technique*
- *Using property generation scripts for assembled fasteners*
- *Output requests for assembled fasteners*
- *Assembled fasteners vs. point-based fasteners*

The assembled fastener technique

Using the assembled fastener technique, you can read in connector and constraint behaviors from a template model and assign these attributes in multiple locations in your main model.
For a large system such as an airframe or automobile, assembled fasteners allow you to define the fastener template once and then assign it many times in the main model. With the appropriate use of coupling constraints and adjust points constraints, secondary nodes in the template model can be automatically resized in the main model to accommodate the actual surface spacing.

The overall process for building assembled fasteners is as follows:

1. Build the template model containing your fastener-like construct: connector section assignments, tie constraints, coupling constraints, and solid or beam section assignments. You must assign names to all surfaces involved in constraints. You must also define a single-point set as the control point, which will be used to locate the template model copies in the main model.
2. Develop your main model, placing attachment points at the locations where you want the template fastener to be replicated. The template model control point will be mapped onto the locations of the attachment points in the main model (see *Figure 1*).
3. Working in your main model, use the **Create Fasteners** and **Edit Fasteners** dialog boxes to define how the template model will be read in, assigned, and oriented.
4. Optionally, use one or more property generation scripts to modify the properties copied into the main model from the template model. Multiple property generation scripts can be used with the same template model to achieve different results in separate assembled fastener objects. For example, you could use two scripts to apply different materials to the same fastener template. You can use the Abaqus Scripting Interface to write your property generation scripts; see *Creating and running your own scripts*, and the *Abaqus Scripting User's Guide*.

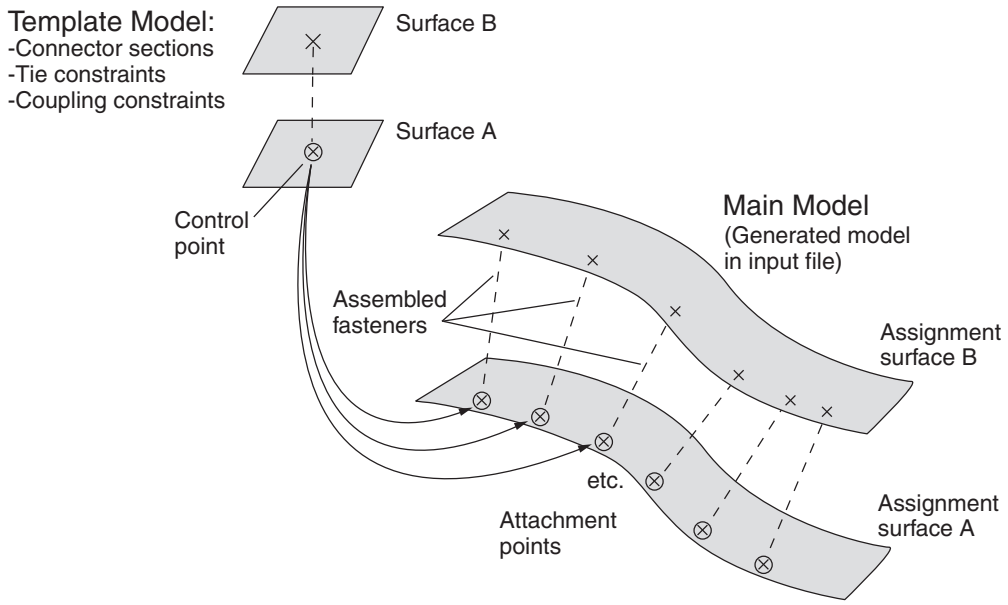


Figure 1: Replicating the template model in the main model.

The template model uses the global coordinate system, and the positive *Z*-axis direction of the fastener construct will be aligned with the selected normal direction in the main model.

Assembled fasteners are different from point-based (mesh-independent) and discrete fasteners in Abaqus/CAE. Assembled fasteners do not create individual fastener objects like point-based and discrete fasteners, but instead they allow you replicate fastener-like behavior in many places. You cannot view or manipulate the individual fasteners (at

each attachment point) in the main model while working in the Abaqus/CAEGUI; they are produced only in the input file generated by Abaqus/CAE. The template model sets are aggregated in the input file generated by Abaqus/CAE to help you manage models containing large numbers of assembled fasteners.

Assembled fastener template models are intended only to model fastener-like constructs and do not provide a generic subassembly capability. Only a few Abaqus/CAE features are supported in template models, such as connectors and mass inertia. No other Abaqus/CAE features are allowed in template models.

Table 1 lists the features that are supported and read from the template model into the main model. The actual model data are not transferred, only the constraints and behaviors.

Table 1: Features supported in assembled fastener template models.

Beam and solid parts (beam, solid, and cohesive elements)
Beam and solid section assignments (but not distributions or composite layups)
Connector section assignments
Tie constraints and coupling constraints (except for tie constraints generated due to incompatible meshes)
Adjust points constraints
Mass inertia

In particular, the Abaqus/CAE features and attributes listed in *Table 2* are not supported by assembled fasteners and are not read in from the template model.

Table 2: Features that are not supported in assembled fastener template models.

Point-based (mesh-independent) and discrete fasteners
Attachment points
Analytical rigid surfaces
Orphan mesh surfaces
Tie constraints generated internally due to incompatible meshes
Any constraint type other than tie, coupling, or adjust points

In addition, the following restrictions exist:

- Solid parts are allowed in the template model but material orientations are not.
- Beam parts are allowed in the template model, but beam orientations must be assigned using the same region as the beam section assignment.
- Solid and beam parts are allowed, but complex parts/assemblies may not work.
- Unmeshed constraint surfaces in the template model are allowed.
- Shell parts are not allowed in the template model, except as reference surfaces for constraints. These constraint surfaces will effectively be replaced with main model surfaces when the input file is generated by Abaqus/CAE. All constraint surfaces should be substituted with main model surfaces for correct behavior of the assembled fastener.
- Template model surfaces are not copied into the main model when the input file is generated by Abaqus/CAE. The only surfaces allowed in the template model are those that will have main model surfaces substituted for them.
- Constraint surfaces cannot be orphan mesh surfaces.
- Reference points are allowed in template models, but attachment points are not allowed.
- The modeling space of the template models must match the modeling space of the main model (see *Part modeling space*).

- Coupling constraints in the template model must constrain all degrees of freedom and must not specify a local coordinate system.
- Tie constraints in the template model must use a generic main surface and a secondary node region, not a secondary surface.
- Point mass and rotary inertia features are allowed in template models. However, for rotary inertia the local coordinate system will be ignored; thus, the only useful scenario is $I_{11} = I_{22} = I_{33}$, and $I_{12} = I_{13} = I_{23} = 0$. Essentially, the rotary inertia feature must have rotational symmetry.

The control point must be a predefined set containing a single vertex or node in your template model. You must create this set in the template model before creating the assembled fasteners in your main model. When the template model is read in, the control point will be placed at the locations of your attachment points in the main model.

The constraint surfaces in the template model must be mapped to corresponding surfaces in the main model to enable the constraint behavior in the main model. When you define the assembled fasteners in the main model, the **Edit Fastener** dialog box will automatically prepopulate the surface assignment table with any template model surfaces involved in tie constraints, coupling constraints, or adjust points constraints. The template surfaces are initially listed in ascending *Z*-axis order using the template model global coordinate system. You can change the order to coincide with your template fastener design; however, the ordering of the surfaces is not significant beyond the selection of the first surface, because the corresponding assignment surface normal is used to orient the assembled fastener. In the main model you must assign names to the surfaces that will be involved in the assembled fastener constraints.

At each attachment point in the main model, the template model is positioned and translated so that the template model control point coincides with the attachment point. The template model is rotated into the main model and oriented such that the positive *Z*-axis of the global coordinate system of the template model aligns with the coordinate system you specify (on the **Orientations** tabbed page of the **Edit Fasteners** dialog box). The default is to orient the template model copies according to the normal vector of the first surface in the main model. This default orientation from the first surface normal creates a *Z*-axis alignment at every attachment point. The *X*-axis is then computed by projecting the global *X*-axis onto the surface.

For any sets that you create in the template model, the main model will aggregate all of the sets' objects into a single set per assembled fastener. When the input file is generated by Abaqus/CAE, template model sets will be aggregated across all attachment points of the assembled fastener. For example, consider a template model that contains a connector section assignment for a wire set named *Wire-1-Set-1*, and that set contains a single wire. If the assembled fastener is then placed at 10 attachment points in the main model, the main model will have a set named *TM-1_Wire-1-Set-1* that contains 10 wires. However, Abaqus/CAE generates these aggregated sets only when it creates the input file. The aggregated sets are not directly visible in Abaqus/CAE.



Note:

The **Adjust control point to lie on surface** option can be useful in coupling constraints you create in assembled fastener template models; see *Defining coupling constraints*. The more general-purpose adjust points constraint can also be useful in assembled fastener template models; see *Defining adjust points constraints*.

The adjust points capability should not be used in assembled fastener template models when the main model attachment points are located at bolt hole center points. Any point along the bolt hole centerline will be moved (incorrectly) to a random location along the perimeter of the hole instead of being projected along the surface normal to the center of the hole.



Note: The size and shape of the template model surfaces do not matter, except with regard to the display of those surfaces during template model rendering from the **Edit Fasteners** dialog box. The recommended best practice is to make your template model surfaces square or rectangular in shape to facilitate rendering speed and accuracy in the main model. Circular template model surfaces or surfaces with any curvature will be rendered in a crudely approximate fashion in the main model.

Using property generation scripts for assembled fasteners

This section discusses how to use property generation scripts to modify template model properties.

In this section:

- *Property generation scripts*
- *Example property generation script*
- *Registering the property generation script*

Property generation scripts

Each assembled fastener can optionally reference a script that will modify the template model properties; for example, to use different materials in the assembled fasteners. These property generation scripts can be used to calibrate or adjust the fastener properties.
In Abaqus/CAE property definitions include material, profile, section, and connector section definitions.

You can use the Abaqus Scripting Interface to write your property generation scripts; see *Creating and running your own scripts*, and the *Abaqus Scripting User's Guide*.

This feature lets you use the same template model multiple times with different property generation scripts. The property definitions from the template model are copied into the main model and given names based on the original names plus a prefix. The default prefix is *TM-1* for the first assembled fastener you create, and *TM-2*, *TM-3*, etc., for subsequent ones. For example, a connector section named *BoltSection* in the template model will be named *TM-1_BoltSection* in the main model. You can change the prefix on the **Properties** tabbed page of the **Edit Fasteners** dialog box. Property prefix strings must be unique for all assembled fasteners you create.

Figure 1 shows an example in which two template models are used with three different property generation scripts.

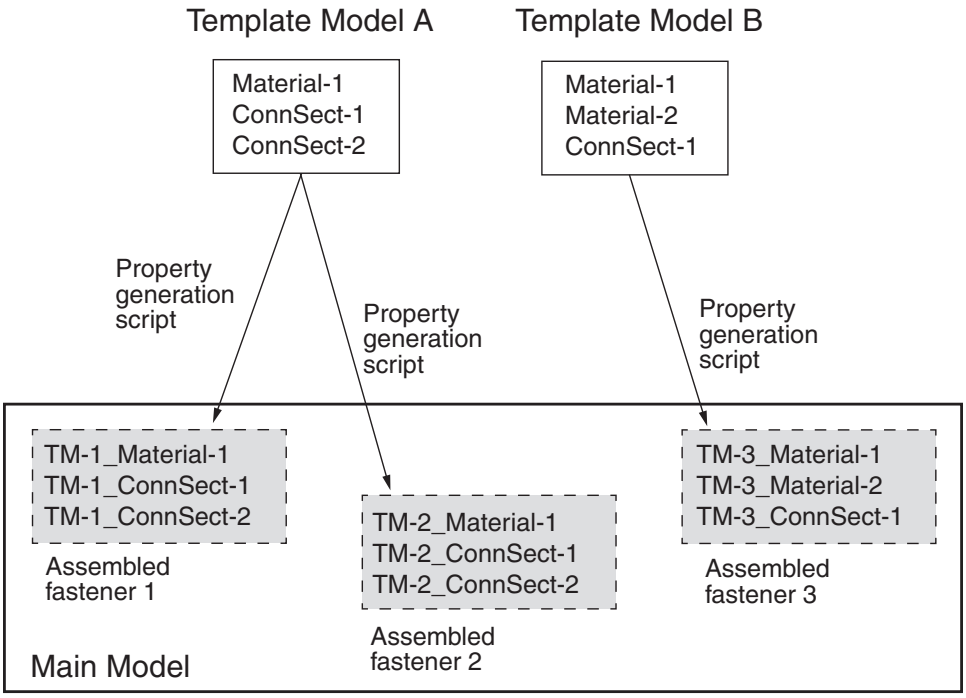


Figure 1: Using multiple property generation scripts.

Example property generation script

A complete example of a property generation script is provided in the blog post “Property generation scripts for assembled fasteners in Abaqus/CAE” in the *https://www.3ds.com/products-services/simulia/simulia-academic-program/learning-community/*. The script uses a method named `getSurfaceSections` that is useful for assembled fastener property scripts. This method accepts a surface name and returns a list of all section names found on the geometry underlying the named surface region; see *Assembly object*. The `getSurfaceSections` method can be used to obtain section information such as the material name and thickness, as shown in the code fragment below.

```
    def __init__(self, scriptName, modelName, fastenerName):
        self.scriptName   = scriptName
        self.modelName    = modelName
        self.fastenerName = fastenerName
        print 'Running script "%s" for "%s"' % (scriptName,
          fastenerName)
        assy = mdb.models[modelName].rootAssembly
        eo = assy.engineeringFeatures.fasteners[fastenerName]
        print eo.assignedSurfaces
        diameter = getInput('Enter %s bolt diameter:' % scriptName)
        print '  Bolt diameter: %s' % diameter
        for aSurf in eo.assignedSurfaces:
            sectNames = assy.getSurfaceSections(aSurf)
            print '  %s: %s' % (aSurf, sectNames)
            # No section assigned
            if (len(sectNames) > 0 and sectNames[0] == ''):
                continue
            for section in sectNames:
                sectObj = mdb.models[modelName].sections[section]
                if (type(sectObj) == HomogeneousShellSectionType):
                    print '    %s:  mat=%s, thk=%s' % \
                        (section, sectObj.material, sectObj.thickness)
```

Registering the property generation script

You provide the file name of the property generation script on the **Properties** tabbed page of the **Edit Fasteners** dialog box. The script is executed only when you click **OK** in the **Edit Fasteners** dialog box, not when Abaqus/CAE creates the input file. The property generation script is run only in the main model on the assembled fasteners, not in the template model.

To make your property generation scripts available from the **Edit Fasteners** dialog box, you must register each script similar to the way in which you register an Abaqus/CAE plug-in. The script must be registered before you startAbaqus/CAE. Any scripts that have been registered will be available from a pull-down list in the **Edit Fasteners** dialog box.

To register a property generation script, you must write a small registration script containing the following lines:

```
from abaqusGui import *
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerAsmbdFastenerScript('filename')
```

This registration script file must be named `registerAsmbFstnrScripts_plugin.py`. In the code replace *filename* with the name of your property generation script. The registration script informs Abaqus/CAE that you have a property generation script named *filename*`.py` that you wish to use with assembled fasteners.

Your property generation scripts and the registration script must be placed in your `abaqus_plugins` directory; see *Where are plug-in files stored?*, for the possible locations of the `abaqus_plugins` directory. For more details about registration scripts, see *What are the kernel and GUI registration commands?*.

Output requests for assembled fasteners

You can obtain both field output and history output from assembled fasteners in your main model. The output must be requested on named sets that you have defined in the template model.

1. In the Step module, working in your main model (not the template model), display the **Edit History Output Request** dialog box or the **Edit Field Output Request** dialog box; see the instructions in *Creating and modifying output requests*.
2. From the **Domain** list, choose **Assembled fastener set**.
3. From the **Fastener** list, choose the name of the assembled fastener for which you want output.
4. From the **Set** list, choose any set from the template model. The assembled fastener in your main model references the sets defined in the template model.
5. Continue selecting output variables and other options, and click **OK** to save your request.

Additional information

- *Creating assembled fasteners*
- *Understanding output requests*

Assembled fasteners vs. point-based fasteners

Assembled fasteners and point-based (mesh-independent) fasteners produce slightly different analysis results when used for the same fastener model. Assembled fasteners and point-based fasteners will generate different coupling nodes and coupling weights, leading to different solutions. The differences between point-based and assembled fasteners are described below and in *Figure 1* and *Figure 2*.

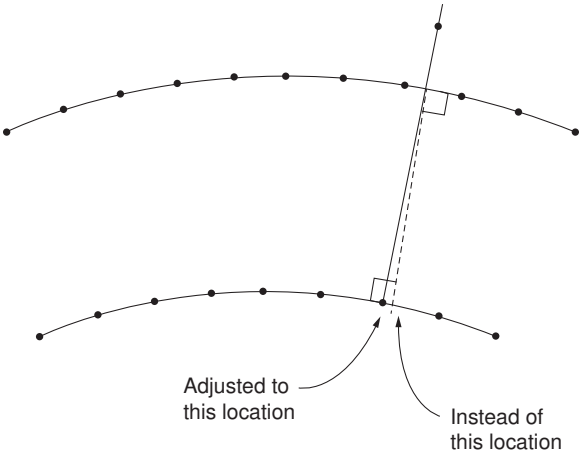


Figure 1: Adjustments for assembled fasteners vs. point-based fasteners.

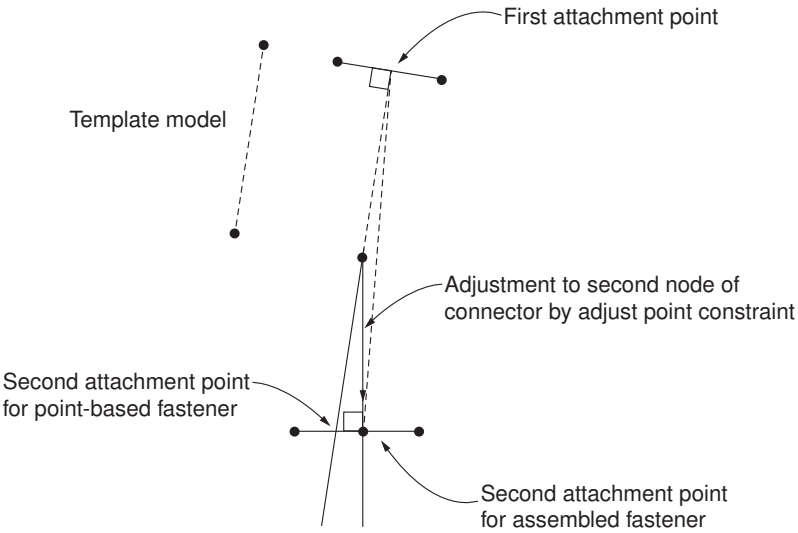


Figure 2: Location of attachment point for assembled fasteners vs. point-based fasteners.

Point-based fasteners

Abaqus looks at the first node of the connector and projects it onto the first surface (given the data line). Abaqus then moves normal to the facet, where it finds the first projection point and finds the subsequent projection points on the second surface.

Assembled fasteners

When the assembled fastener is instanced in the main model, Abaqus places the first node of the connector where the first projection point of the point-based fastener model is located. Both fastener types accomplish this

placement with normal projections. However, for assembled fasteners an adjust point constraint is performed on the second node of the connector; i.e., a normal projection onto the second surface. This action is not the same as moving normal to the first projection point facet unless the two normals align perfectly. Therefore, the second (subsequent) attachment points can be different for assembled fasteners versus point-based fasteners, which will lead to differences in the analysis solution.

In an ideal case where the two surfaces are placed exactly the same distance apart as the template model connector and are parallel to each other, Abaqus will get the same projection points for assembled fasteners and point-based fasteners. However, this situation will not be the case in general.

Managing fasteners

The **Fasteners Manager** allows you to create and manage fasteners. The manager includes a list of the names and types of fasteners that you have defined. The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the manager allow you to create new fasteners or to edit, copy, rename, and delete existing ones. The icons in the column along the left side of the manager allow you to suppress and resume existing fasteners. You can also initiate these procedures using the **Special->Fasteners** menu from the main menu bar in the Interaction module. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

Additional information

- *Suppressing and resuming objects*
- *Creating point-based fasteners*
- *Creating discrete fasteners*

Creating point-based fasteners

You can create point-based fasteners in the Interaction module by picking each positioning point from the viewport or by reading the coordinates of the positioning points from a file.
You can allow Abaqus/CAE to project the positioning point to the closest face along a normal to the surface, or you can specify the direction vector along which the point is projected to the closest face. For more information, see *About fasteners*.

In this section:

- *Defining the surfaces where fasteners will be connected*
- *Defining the fastener attachment zone*
- *Defining the fastener properties*
- *Defining the fastener formulation*
- *Adjusting the fastener definition*

Defining the surfaces where fasteners will be connected

You use the **Domain** tabbed page to edit the positioning points and to select the target surfaces onto which Abaqus/CAE projects the points and the vector along which the points are projected.

1. From the **Edit Fasteners** dialog box, display the **Domain** tabbed page.
2. Choose the direction vector along which the positioning points will be projected onto the target surface.
   - By default, Abaqus/CAE creates the direction vector along which the positioning points will be projected along the normal to the closest face.
   - To change the direction vector, toggle on **Specify**, click , and select the points representing the start and end of the direction vector. Abaqus/CAE displays the direction vector.
3. Select the approach for the target surfaces.
   - Choose **Whole model** to use all of the surfaces in the model as the target surfaces.
   - Choose **Fasten specified surfaces by proximity** to specify the surfaces to be connected. The fastening point connectivity is defined by the relative positions of the surfaces along the direction vector.
   - Choose **Fasten in specified order** to specify a particular order of the target surfaces. The fastening point connectivity is determined by the direction vector.
4. Double click the first row in the target surfaces table, and select surfaces in the viewport. Click **Done** in the prompt area to indicate you have finished the surface selection.

   You can select multiple surfaces from the viewport to define a single surface region.
5. Click mouse button 3 anywhere on the target surfaces table to edit the selected surface, add a new surface, or delete a surface.

Additional information

• *Mesh-Independent Fasteners*
• *About fasteners*
• *Understanding attachment points and lines*

Defining the fastener attachment zone

You use the **Criteria** tabbed page to specify parameters that Abaqus uses to determine where to project the fasteners when you analyze the model.

1. From the **Edit Fasteners** dialog box, display the **Criteria** tabbed page.
2. Choose the **Attachment method**. In general, a positioning point should be located as close to the surfaces being connected as possible. Abaqus determines the actual fastening points where the fasteners are attached to the surfaces being connected by first projecting the positioning point onto the closest surface. You can choose from the following projection methods:

   - **Face-to-Face**
   - **Face-to-Edge**
   - **Edge-to-Face**
   - **Edge-to-Edge**

   The method that you should choose depends on how the surfaces are oriented relative to each other. For more information, see *Specifying the Positioning Points, Projection Method, and Fastening Points*.
3. Choose the method for specifying the distance from the positioning points within which the fastening points must lie.

   - Choose **Facet-based default** to allow Abaqus to compute a default search radius based on the facet thickness (for shell elements) or characteristic facet length (for non-shell elements) in the vicinity of each positioning point.
   - Choose **Specify**, and enter the maximum distance from the positioning point within which the fastening points will lie.

   For more information, see *Forming Fasteners on Surfaces inside a User-Specified Search Zone*.
4. Choose the maximum number of layers through which the fastening points will be projected.

   - Choose **All** to allow Abaqus to use the maximum possible number of layers.
   - Choose **Specify**, and enter the number of layers.

Additional information

• *Mesh-Independent Fasteners*
• *About fasteners*
• *Understanding attachment points and lines*

Defining the fastener properties

You use the **Property** tabbed page to define the radius and the mass of the fastener. You can also specify the connector section and the orientation of the local directions at each end of the connector.
Depending on the connection type, a local orientation for the first node of the connector is required, optional, or not applicable. Depending on the connection type, a local orientation for the second node of the connector is optional or not applicable. For the local orientation requirements for each connection type, see the summary tables in *Connection-Type Library*.

For a description of how you can combine connector orientations and fastener orientations, see *Defining the Fastener Orientation*.

1. From the **Edit Fasteners** dialog box, display the **Property** tabbed page.
2. Enter the **Physical radius** of the fastener to define the radius of its circular projection on the connected surfaces. Abaqus uses this information to define the geometric section properties of the fastener.
3. Display the **Section** tabbed page, and choose the method that you will use to model each fastener.

   - Choose **Connector section** and select a connector section to model the point-to-point connection with connectors. Like other uses of connectors in Abaqus/CAE, the connection can be fully rigid or allow for unconstrained relative motion in the local connector component directions. In addition, you can specify deformable behavior using a connector behavior definition that includes the effects of elasticity, damping, plasticity, damage, and friction.
   - Choose **Rigid MPC** to model a perfectly rigid fastener with a rigid, or beam, multi-point constraint (MPC).

   If you choose a connector section that uses a basic, assembled, or complex connection type to model the fasteners, you can request output of connector element output variables. However, if you use a beam MPC to model the fasteners, no output is available from the fasteners.
4. Display the **Connector Orientation 1** tabbed page to define the local directions for the first node of the connector. The **Connector Orientation 1** tabbed page is available only if you use a connector section to model the fasteners.
   a. In most cases Abaqus/CAE orients the nodes at both ends of the connector with the global coordinate system; however, some connection types that are referred to by the connection section require local directions. Click  to select a datum coordinate system that specifies local directions at the first node of the connector. Click  to create the datum coordinate system if it does not already exist.
   b. Do either of the following:

      - Choose **No modifications to CSYS** to use the selected datum coordinate system.
      - If desired, choose **Additional rotation angle** and specify an additional rotation angle and the axis about which Abaqus/CAE will apply the additional rotation. The additional rotation (in degrees) is applied to both directions orthogonal to the selected axis.
5. Display the **Connector Orientation 2** tabbed page to define the local directions for the second node of the connector. The **Connector Orientation 2** tabbed page is available only if you select a datum coordinate system that specifies local directions for the first node of the connector.

   a. Click  to select a datum coordinate system that specifies local directions for the second node of the connector. Click  to create the datum coordinate system if it does not already exist.
   b. Do one of the following:

- Choose **Use orientation 1** to define the same local directions that you specified on the **Connector Orientation 1** tabbed page.
- Choose **No modifications to CSYS** to use the selected datum coordinate system.
- If desired, choose **Additional rotation angle** and specify an additional rotation angle and the axis about which Abaqus/CAE will apply the additional rotation. The additional rotation (in degrees) is applied to both directions orthogonal to the selected axis.

Additional information

• *Connection Types*
• *Mesh-Independent Fasteners*
• *About fasteners*
• *Understanding attachment points and lines*

Defining the fastener formulation

You use the **Formulation** tabbed page to select the method that Abaqus/CAE uses to couple the motion of the fastening points to the selected surfaces. You can also constrain selected rotational degrees of freedom and choose the weighting method.

1. From the **Edit Fasteners** dialog box, display the **Formulation** tabbed page.
2. Choose the method to couple the motion of the fastening points to the average motion of the nodes on the surfaces that fall inside the influence radius.

   - Choose **Continuum distributing** (default) to use a general-purpose distributed coupling scheme.
   - Choose **Structural distributing** to augment the distributed coupling scheme to account for rotations of surface nodes such that a fastening point remains approximately on a shell surface on bending.

   For more information, see *Distributing Coupling Constraints*.
3. If desired, toggle off a rotational degree of freedom to remove its constraint. Abaqus/CAE always constrains all translational degrees of freedom in a fastener.
4. Choose the weighting method that Abaqus uses to modify the default weight distribution at the coupling nodes within the radius of influence.

   - Choose **Uniform** (default) to select a weight distribution that is uniform and equal to 1.0.
   - Choose **Linear** to select a weight distribution that decreases linearly with distance from the fastening point.
   - Choose **Quadratic** to select a weight distribution that decreases by a quadratic polynomial function with distance from the fastening point.
   - Choose **Cubic** to select a weight distribution that decreases by a cubic polynomial monotonic function with distance from the fastening point.

   For more information, see *Weighting Methods*.

Additional information

• *Mesh-Independent Fasteners*
• *About fasteners*
• *Understanding attachment points and lines*

Adjusting the fastener definition

You use the **Adjust** tabbed page to specify the orientation of the fastener. For a description of how you can combine connector orientations and fastener orientations, see *Defining the Fastener Orientation*. You can also specify how Abaqus will compute the radius of influence, which determines the group of nodes that will be associated with each fastening point.

1. From the **Edit Fasteners** dialog box, display the **Adjust** tabbed page.
2. By default, Abaqus determines the orientation of each fastener from the default local directions of the surface that is closest to the fastening point. Click  to select a datum coordinate system (**CSYS**) that specifies the orientation of the fastener. Click  to create the datum coordinate system if it does not already exist. The z-axis of the **CSYS** should be approximately normal to the surfaces that are being connected, and the local x- and y-axes should be approximately tangent to the surfaces that are being connected.
3. By default, Abaqus adjusts the orientation that you specified such that the local z-axis for each fastener is normal to the surface that is closest to the fastening point. Toggle off **Adjust CSYS to make Z-axis normal to closest surface** to use the selected coordinate system to define the local directions precisely.
4. Each fastening point is associated with a group of nodes on the surface in the immediate neighborhood of the fastening point called a region of influence. The motion of the fastening point is then coupled in a weighted sense to the motion of the nodes in this region by a distributed coupling constraint. Each region of influence contains a minimum of three nodes. Choose either of the following to define the region of influence:
   - Choose **Facet-based default** to allow Abaqus to compute an internal radius of influence based on the geometric properties of the fastener, the characteristic length of the connected facets, and the weighting method you selected. The default radius of influence is always chosen to be the largest of the internally computed radius of influence, the physical fastener radius, and the distance from the projection point to the closest node.
   - Choose **Specify**, and enter the radius of influence. If you specify a radius of influence that is smaller than the computed default radius of influence, Abaqus uses the computed value.

   For more information, see *Defining the Radius of Influence*.

Additional information

• *Mesh-Independent Fasteners*
• *About fasteners*
• *Understanding attachment points and lines*

Creating discrete fasteners

You can create discrete fasteners in the Interaction module by selecting the attachment lines that connect the desired surfaces.
For each attachment line, Abaqus/CAE determines the fastening points and applies a distributing coupling between the fastening points and their corresponding surfaces.

You must assign a connector section to each attachment line that you select. You can use any of the Abaqus connection types to model complex behavior of your fasteners, such as deformable connectors that include the effects of elasticity, damage, plasticity, and friction. In addition, you can assign different connector sections to different attachment lines.

1. From the main menu in the Interaction module, select **Special->Fasteners->Create**.

   Abaqus/CAE displays the **Create Fasteners** dialog box.

    **Tip:** You can also create discrete fasteners using the  tool in the Interaction module toolbox.

2. From the **Create Fasteners** dialog box that appears, select **Discrete** fasteners and click **Continue**.
3. Select the attachment lines that represent the location of the discrete fasteners, and click **Done**. For more information, see *Creating attachment lines by projecting points*.

   Abaqus/CAE displays the **Edit Fasteners** dialog box.

4. Choose the method to couple the motion of the fastening points to the average motion of the nodes on the surfaces that fall inside the influence radius.

   - Choose the **Coupling type** option **Continuum distributing** (default) to use a general-purpose distributed coupling scheme or **Structural distributing** to augment the distributed coupling scheme to account for rotations of surface nodes such that a fastening point remains approximately on a shell surface on bending.
   - Choose the **Rotational Coupling type** option **Rotational Continuum distributing** or **Rotational Structural distributing** (default).

   For more information, see *Distributing Coupling Constraints*.

5. If desired, toggle off a rotational degree of freedom to remove its constraint. Abaqus/CAE always constrains all translational degrees of freedom in a distributing coupling.
6. Choose the weighting method that Abaqus will use to modify the default weight distribution at the coupling nodes:

   - Choose **Uniform** (default) to select a weight distribution that is uniform and equal to 1.0.
   - Choose **Linear** to select a weight distribution that decreases linearly with distance from the fastening point.
   - Choose **Quadratic** to select a weight distribution that decreases by a quadratic polynomial function with distance from the fastening point.
   - Choose **Cubic** to select a weight distribution that decreases by a cubic polynomial monotonic function with distance from the fastening point.

   For more information, see *Weighting Methods*.

7. If desired, select or create a datum coordinate system that specifies the initial orientation of the local system in which the constrained degrees of freedom are defined. By default, the orientation of the constrained degrees of freedom is defined by the global coordinate system.
8. Specify the radius of influence centered about the fastening point. You can enter a value or allow Abaqus to use the entire surface to define the coupling constraint.

9. Click **OK** to close the **Edit Fasteners** dialog box.

   Abaqus/CAE displays the distributing coupling of the discrete fastener between the fastening points and connected faces. The radius of the circle reflects the specified radius of influence. If the coupling constraint definition uses the entire surface, the lines from the fastening point will extend past the circle. For orphan mesh models, you can modify the appearance of the distributing coupling by changing the face and edge density in the assembly display options; for more information, see *Controlling the display of attributes*.

10. Complete the discrete fastener definition by assigning a connector section to the attachment lines, as described in *Creating and modifying connector section assignments*.

    Abaqus/CAE displays the connector symbols for the discrete fastener.

Additional information

- *Coupling Constraints*
- *About fasteners*
- *Understanding attachment points and lines*

Creating assembled fasteners

You can create assembled fasteners in the Interaction module after you have developed your template model. For more information, see *About fasteners*.

In this section:

- *Mapping the template model surfaces to main model surfaces*
- *Orienting the geometry read in from the template model*
- *Modifying fastener properties in the main model*
- *Re-running a property generation script*

Mapping the template model surfaces to main model surfaces

You use the **Surfaces** tabbed page to assign the constraint surfaces from the template model to the corresponding target surfaces that will be fastened in your main model. The constraint surfaces must be mapped into the main model to enable the constraint behaviors in the main model.

All surfaces involved in the template model constraints must have assigned names. The corresponding target surfaces in the main model must also have names.

1. From the **Edit Fasteners** dialog box, display the **Surfaces** tabbed page.
2. In the **Constraint Surface Substitutions** table, assign each **Template Surface** to an **Assignment Surface** in the main model.

   The **Template Surface** column will be prepopulated with the names of any template model surfaces involved in tie constraints, coupling constraints, or adjust points constraints. If the template surfaces are ordered correctly, you can simply choose the main model surface names in the **Assignment Surface** column of the table. If you need to change the order of the template model surfaces that will be assigned, click in each table cell and then click the arrow that appears to display the list of available surface names. The template surfaces are initially listed in ascending *Z*-axis order, but you can change the order to coincide with your template fastener design. However, the ordering of the surfaces is not significant beyond the selection of the first surface, because the corresponding assignment surface normal is used to orient the assembled fastener.

   To choose the main model assignment surface for each template surface, click in the corresponding table cell and click the arrow that appears to choose from the list of available surfaces in the main model. If you have not assigned a name to any of the surfaces you need to use, click  (**Create Assignment Surface**) to define a new surface and add it to the list.
3. You can toggle on **Render template surfaces** to show the surfaces in the main model, if desired.

Orienting the geometry read in from the template model

You use the **Orientations** tabbed page to specify how the template fastener construct is oriented at each attachment point. The template model is rotated into the main model and oriented such that the global coordinate system of the template model aligns with the coordinate system you specify. The default is to orient the template model copies such that the positive *Z*-axis of the global coordinate system of the template model aligns with the normal vector of the first surface at every attachment point. This default orientation from the first surface normal creates a *Z*-axis alignment at every attachment point. The *X*-axis is then computed by projecting the global *X*-axis onto the surface. When the global *X*-axis is parallel to the normal or to the *Z*-axis alignment, then the *X*-axis is computed by projecting the global *Y*-axis onto the surface. Further, the *Y*-axis is computed by taking the cross product of the normal and the computed *X*-axis.

1. From the **Edit Fasteners** dialog box, display the **Orientations** tabbed page.
2. Choose the orientation option that Abaqus/CAE will use to orient the fastener template at each attachment point.

   - Choose **Compute orientation based on first surface normal** (default) to orient the fastener geometry such that the global coordinate system of the template model aligns with the normal vector of the first assignment surface in the main model.

     You can toggle on **Flip normal direction** to reverse the direction, if desired.
   - Choose **Specify uniform CSYS** to orient the fastener geometry with a coordinate system that you choose. Click  (**Edit**) to select from the existing datum coordinate systems in your main model.

     The default is to use the global coordinate system of the main model. Click  (**Create Datum CSYS**) to create a new coordinate system.

3. If desired, toggle on **Display orientation direction at attachment points** to show orientation triads at the attachment point of each assembled fastener. These visualization options are not available until you have selected the control point set and made the first surface assignment. Choose either of the following suboptions:

   - Toggle on **Display only the normal direction** to remove the 1-axis and the 2-axis from the triads, leaving only the surface normal vector.
   - Toggle on **Include template model surfaces every *x* points** to display sketched outlines of the constraint surfaces from the template model. Adjust the number ***x*** up or down to show the surfaces at every attachment point, every other point, every third point, etc. The surfaces will be color-coded according to the colors shown on this tabbed page and in the **Constraint Surface Substitutions** table on the **Surfaces** tabbed page.

     This option is helpful for positioning and orienting the template model correctly. If your template model surfaces are too large, however (relative to the main model dimensions), the resulting display may be too crowded. You may wish to go back and resize the surfaces in your template model if this is the case.

Modifying fastener properties in the main model

You use the **Properties** tabbed page to specify a property generation script to modify the properties copied into the main model from the template model. The script is executed by Abaqus/CAE only when you click **OK** in the **Edit Fasteners** dialog box, not when Abaqus/CAE creates the input file. See *Using property generation scripts for assembled fasteners*, for complete information.

Your property generation scripts must be registered before starting Abaqus/CAE; see *Registering the property generation script*. Any scripts that have been registered will be available for selection on the **Properties** tabbed page.

1. From the **Edit Fasteners** dialog box, display the **Properties** tabbed page.
2. By default, the properties copied into the main model from the template model are named with the prefix *TM-1* for the first assembled fastener you create. For example, a connector section named *BoltSection* in the template model will be named *TM-1_BoltSection* in the main model. Subsequent assembled fasteners will use the prefixes *TM-2*, *TM-3*, etc.

   If desired, you can change the name prefix used in the **Property prefix string** field. The property prefix string must be unique for each assembled fastener.
3. Toggle on **Specify a property generation script**, and select the filename of your script from the list.

Re-running a property generation script

You use the **Re-run** tabbed page to have Abaqus/CAE re-copy properties from the template model and then re-run the property generation script. The script currently specified on the **Properties** tabbed page is run. The action occurs when you click **OK** to close the dialog box.

The **Re-run** tabbed page is available only when you are editing an assembled fastener that has already been created.



Note:

Any time you make a change in the the **Edit Fasteners** dialog box, the re-run action occurs automatically when you click **OK**. You only need to use the **Re-copy template model properties and re-run script** option when no other changes have been made.

1. From the **Edit Fasteners** dialog box, display the **Re-run** tabbed page.
2. Toggle on **Re-copy template model properties and re-run script**.
3. Click **OK** to close the dialog box.

Fracture mechanics

You can do the following to model fracture mechanics with Abaqus/CAE:

- Create a seam crack that defines an edge or a face with overlapping nodes that can separate during an analysis.
- Use contour integral estimates to study the onset of cracking in quasi-static problems; however, a contour integral estimate does not predict how a crack will propagate. You can compute contour integrals for two- or three-dimensional models.
- Use the extended finite element method (XFEM) to study the initiation and propagation of a crack along an arbitrary, solution-dependent path without needing to remesh your model. XFEM is available only for three-dimensional solid and planar models and orphan meshes.
- Use the virtual crack closure technique (VCCT) to study the initiation and propagation of a crack along a known crack surface.
- Use the **Crack Manager** to create and manage your cracks.

In this section:

- *Seam cracks*
- *Using contour integrals to model fracture mechanics*
- *Using the extended finite element method to model fracture mechanics*
- *Using the virtual crack closure technique to model crack propagation*
- *Managing cracks*

Seam cracks

A seam defines an edge or a face with overlapping nodes that can separate during an analysis. You can include a seam crack in your model. Alternatively, you can refer to the seam when creating a contour integral; however, you cannot use a seam crack with the extended finite element method (XFEM).

In this section:

- *What is a seam?*
- *Creating a seam*

What is a seam?

A seam defines an edge or a face in your model that is originally closed but can open during an analysis. Abaqus/CAE places overlapping duplicate nodes along a seam when the mesh is generated. A seam cannot extend along the boundaries of a part and must be embedded within a face of a two-dimensional part or within a cell of a solid part. After you create a seam, you can determine its crack properties using a contour integral analysis. Because a seam modifies the mesh, you cannot create a seam on a dependent part instance.

Figure 1 shows a seam on the face of a planar part and the effect of applying a tensile load to the model. The duplicate nodes along the seam are independent of each other and are free to move.

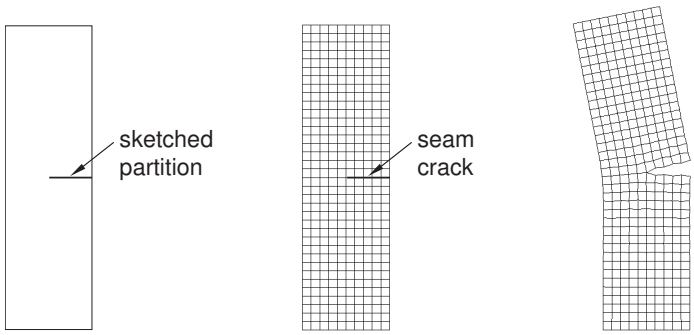


Figure 1: A seam embedded in a face.

Figure 2 shows a similar analysis of a seam embedded in a solid part. The seam was created by partitioning the solid with a sketch drawn on a datum plane.

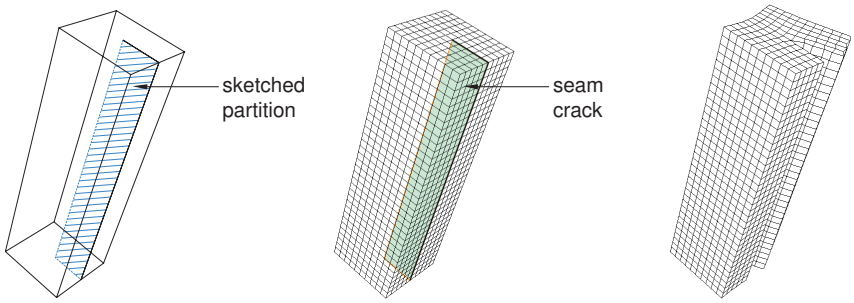


Figure 2: A seam embedded in a cell.

For detailed instructions, see *Creating a seam*, .

Creating a seam

You can use the **Special** menu in the Interaction module to define a seam in your model. A seam defines a region in your model that can open during an analysis. Abaqus/CAE places overlapping duplicate nodes along a seam when the mesh is generated. A seam cannot extend along the boundaries of a part and must be embedded within a face of a two-dimensional part or within a cell of a solid part. You can use the seam when you are selecting the crack front and the crack tip that will be used in a contour integral analysis. For more information, see *What is a seam?*.

1. From the main menu bar in the Interaction module, select **Special->Crack->Assign seam**.
2. From the model in the viewport, select the entities representing the seam. The entities must be embedded edges within a face of a two-dimensional part or embedded faces within a cell of a solid part; you cannot select any entities that lie on the boundary of the part.
3. Click mouse button 2 to indicate that you have finished selecting the seam.

   Abaqus/CAE creates the seam.
4. To include the seam in a contour integral analysis, follow the procedure described in *Creating a contour integral crack*.

Additional information

- *What is a seam?*
- *Fracture Mechanics*

Using contour integrals to model fracture mechanics

You can study the onset of cracking in a quasi-static model by selecting regions from your model that will be used to compute contour integral estimates.
During the analysis Abaqus/Standard writes the values of the contour integrals to the output database as history output. For a detailed description of contour integrals, see *Contour Integral Evaluation*. This section describes how you define a crack in Abaqus/CAE.

In this section:

- *Performing a contour integral analysis*
- *Defining the crack front*
- *Defining the crack tip or crack line*
- *Defining the crack extension direction*
- *Controlling the singularity at the crack tip for a small-strain analysis*
- *Contour integral output*
- *Meshing the crack region and assigning elements*
- *Controlling the singularity at the crack tip*
- *Creating a contour integral crack*
- *Modifying data for contour integrals*
- *Requesting contour integral output*

Performing a contour integral analysis

You can study the onset of cracking in quasi-static problems using contour integral estimates; however, a contour integral estimate does not predict how a crack will propagate.

You can compute contour integrals for two- or three-dimensional models, and you can choose to perform one of the following types of contour integral calculations:

- *J*-integral
- C_t-integral (for creep)
- *T*-stress (for linear materials)
- Stress intensity factors for linear homogeneous materials and for interfacial cracks lying on the interface between two linear homogeneous materials

For more information, see *Contour Integral Evaluation*.

You define cracks in the Interaction module. To perform a contour integral analysis, you must select the crack front, the crack tip or crack line, and the crack extension direction, as described in the following sections. The entities that you can select depend on whether the part is two- or three-dimensional and whether you are defining the part using geometry or using elements and nodes from an orphan mesh. In some cases the crack tip or the crack line is the same as the crack front that you selected, and Abaqus/CAE selects the crack tip or crack line for you.

A crack in a two-dimensional model is a region containing edges that are free to move apart. A crack in a three-dimensional model is a region containing faces that are free to move apart. The simplest approach to performing a contour integral analysis uses a region that already contains edges or faces that are free to move apart as the crack separates. Alternatively, you can model the crack as a line embedded in a face in a two-dimensional model or as a face embedded in a cell in a three-dimensional model. The embedded line or face is called a seam, and you can perform a contour integral analysis using a seam. When you mesh the model, Abaqus/CAE creates duplicate overlapping nodes on the seam; these coincident nodes are free to move apart as the seam separates. Seams are described in more detail in *What is a seam?*. *Figure 1* shows a two-dimensional model with existing edges that are free to move apart and a similar model with an embedded seam that is free to move apart.

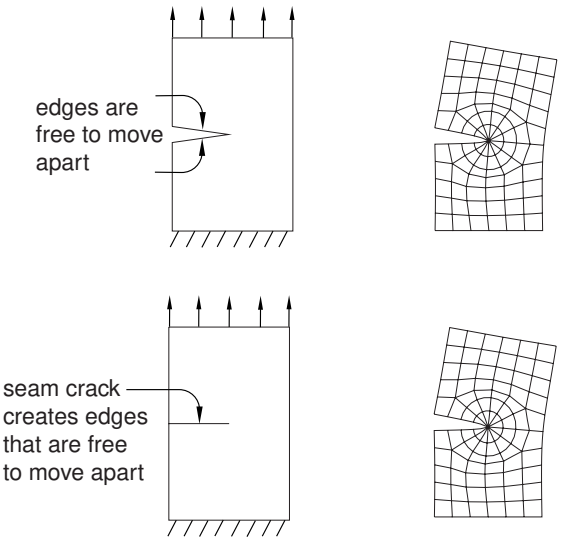


Figure 1: You can define a crack using an existing region or using a seam.

You can specify that the crack front is defined on a symmetry plane, in which case you need to model only half of the structure. Abaqus doubles the values calculated by the contour integral to arrive at the correct values. In most cases you need a refined mesh surrounding the crack. After you create a crack, you must use the **History Output Request**

editor to request that Abaqus write contour integral information to the output database. For more information, see *Contour integral output*.

For detailed instructions, see *Creating a contour integral crack*.

Defining the crack front

The first step in the procedure to configure a contour integral is to define the crack front by selecting entities from the assembly. The crack front is the forward part of the crack. Abaqus uses the crack front to compute the first contour integral using all of the elements inside the crack front and one layer of elements outside the crack front. You can request output for more than one contour integral, in which case Abaqus/CAE adds a single layer of elements to the group of elements that were used to calculate the previous contour integral. *Figure 1* illustrates how Abaqus/CAE computes successive contour integrals for a two-dimensional model by adding layers of elements.

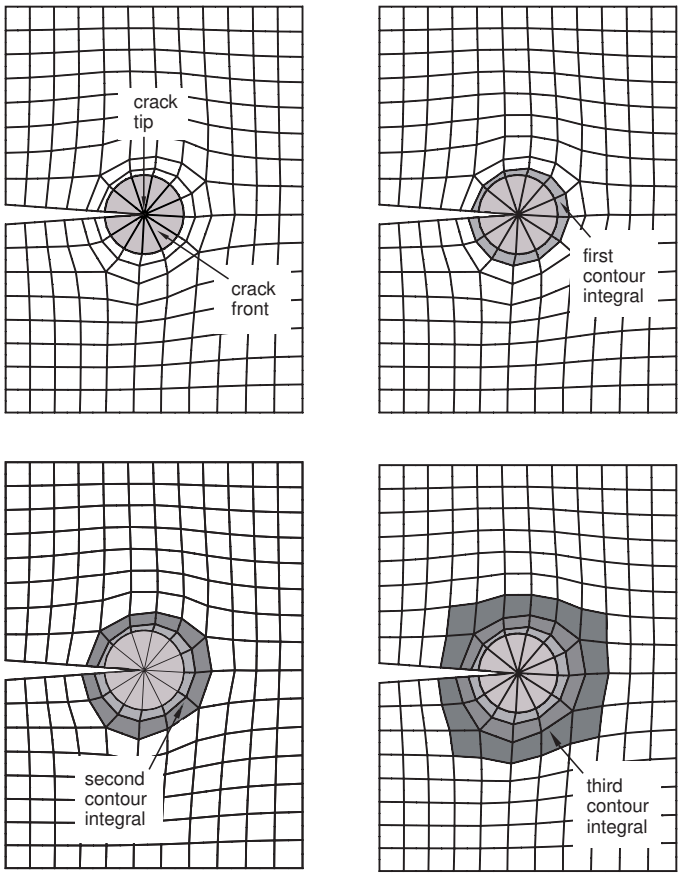


Figure 1: Successive contour integrals are calculated by adding a layer of elements.

If your part is three-dimensional, Abaqus computes contour integrals at each node along the crack line, as shown in *Figure 2*. For more information, see *Defining the Crack Front*.

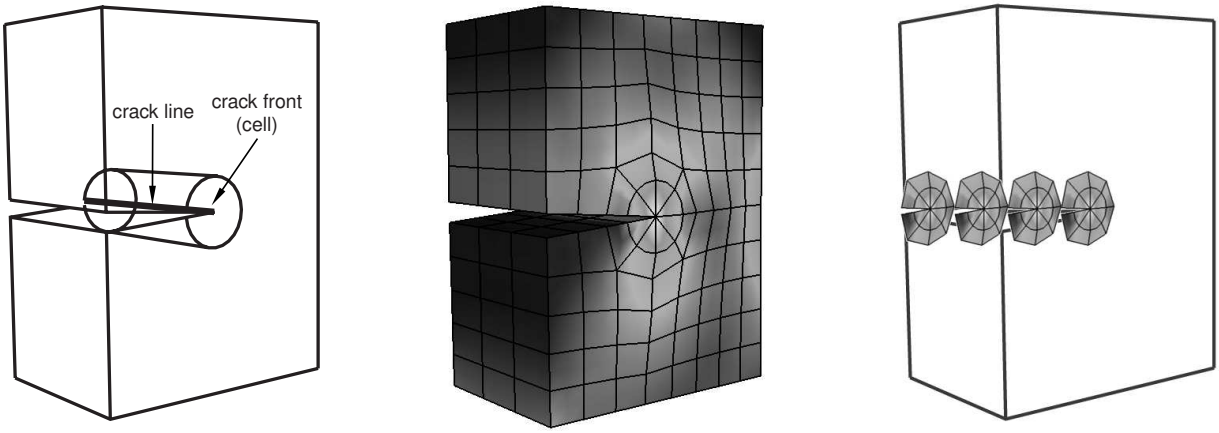


Figure 2: Abaqus computes the contour integral at each node along the crack line.

The entities from which you can select depend on whether the crack front is located in geometry or an orphan mesh and on the modeling space of the part.

Geometry

When you are defining the crack front on geometry, the entities that you can select depend on the modeling space of the part.

Two-dimensional geometry

If you are defining the crack front on two-dimensional geometry, you can select the following:

- A single vertex
- Connected edges
- Connected faces

Figure 3 shows the entities from which you can select when defining a crack front on two-dimensional geometry.

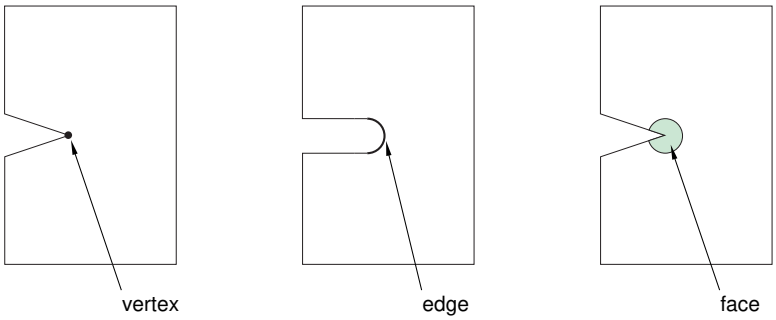


Figure 3: Selecting the crack front from two-dimensional geometry.

Three-dimensional geometry

If you are defining the crack front on three-dimensional geometry, you can select the following:

- Connected edges
- Connected faces
- Connected cells

Figure 4 shows the entities from which you can select when defining a crack front on three-dimensional geometry.

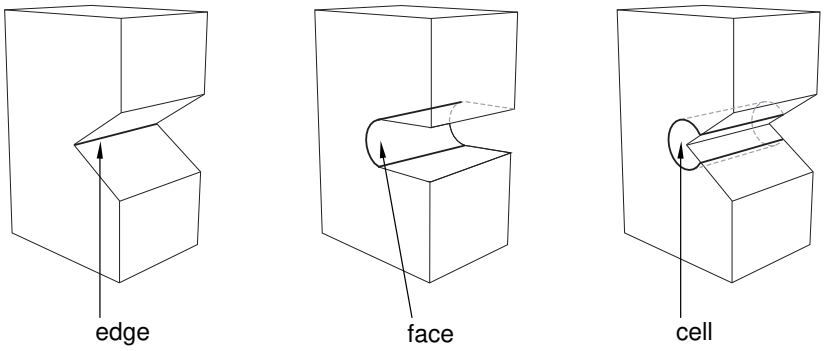


Figure 4: Selecting the crack front from three-dimensional geometry.

Mesh

When you are defining the crack front on an orphan mesh, you can select the elements or element edges or faces that define the crack front. Alternatively, you can select the nodes from the corresponding region. When you are defining the crack front on an orphan mesh, the entities that you can select depend on the modeling space of the part.

Two-dimensional orphan mesh

If you are defining the crack front on a two-dimensional orphan mesh, you can select the following:

- A single node
- Connected element edges
- Connected elements

Figure 5 shows the entities from which you can select when defining a crack front on a two-dimensional orphan mesh.

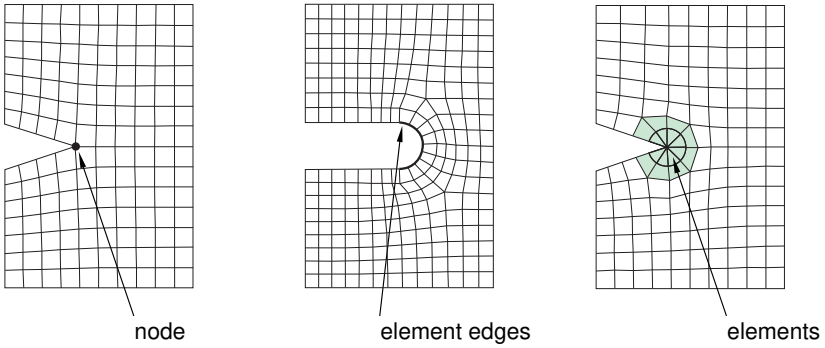


Figure 5: Selecting the crack front from a two-dimensional orphan mesh.

Three-dimensional orphan mesh

If you are defining the crack front on a three-dimensional orphan mesh, you can select the following:

- Connected element edges
- Connected element faces
- Connected elements

Figure 6 shows the entities from which you can select when defining a crack front on a three-dimensional orphan mesh.

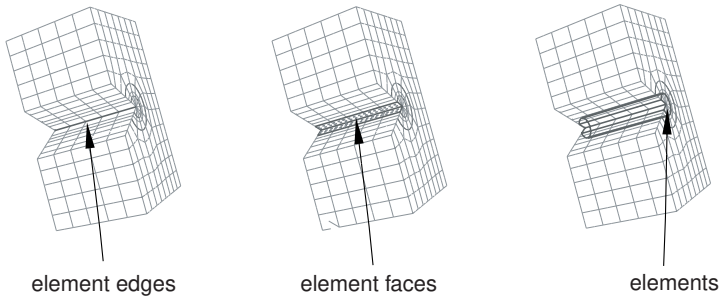


Figure 6: Selecting the crack front from a three-dimensional orphan mesh.

Defining the crack tip or crack line

After you have defined the crack front, the next step in the procedure to configure the contour integral is to define the crack tip or crack line by selecting entities from the assembly. The modeling space of your assembly governs whether you need to define a crack tip or a crack line for a contour integral analysis.

Two-dimensional

If your assembly is two-dimensional, you must define the crack tip by selecting a vertex or a node. The crack tip is the point on the crack front where you define the crack extension direction, $\mathbf{q}$. In some cases the desired vertex or node will not exist, and you must create it by partitioning the crack front.

If you selected a vertex or a node to define the crack front, the same vertex or node defines the crack tip.

Three-dimensional

If your part is three-dimensional, you must define the crack line by selecting edges or element edges that form a continuous line. The crack line is a series of connected edges along the crack front where you define the crack extension direction, $\mathbf{q}$. In some cases the desired edges will not exist, and you must create them by partitioning the crack front.

If you selected edges or element edges to define the crack front, the same edges define the crack line.

Selected edges (from an Abaqus/CAE native part or from an orphan mesh) must be connected, must connect one side of the crack front to the other, and must be included in the crack front.

Defining the crack extension direction

After you have defined the crack front along with the crack tip or crack line, Abaqus/CAE prompts you to specify the crack extension direction at the crack tip or along the crack line. You can specify the normal to the crack plane, **n**; or you can specify the crack extension direction, **q**, directly. For a detailed discussion of how Abaqus uses **n** or **q** to calculate the crack extension direction, see *Specifying the Virtual Crack Extension Direction*.

Normal to crack plane

If you select **Normal to crack plane**, you can define the normal by selecting points from the model that represent the start and the end of the normal to the crack plane. The crack plane contains the **q** vector that Abaqus needs to compute the contour integral. In many cases the crack plane represents the plane of symmetry of the crack.

You should define the normal to the crack plane only if the direction is the same at all points along the crack line. If the direction of the normal to the crack plane varies along the crack line, you cannot select a single normal that defines the crack extension direction at all points along the crack line.

To define the normal to the crack plane, you can select points from geometry (such as vertices, datum points, or midpoints), or you can select orphan mesh nodes. Alternatively, you can enter the coordinates of the points in the prompt area. If you select points from geometry and subsequently modify the part, Abaqus/CAE regenerates the points and updates the normal accordingly. If you are working with orphan mesh nodes, you must select nodes that represent the start and the end of the normal.

Abaqus calculates a crack extension direction, **q**, that is orthogonal to the crack front tangent, **t**, and the normal, **n**. If required, you can flip the **q**-direction vector.

q vectors

If you select **q vectors**, you can define the crack extension direction, **q**, directly by selecting points from the model that represent the start and end of the **q** vector. If you are working with orphan mesh nodes, you must select nodes that represent the start and the end of the **q** vector. Alternatively, you can enter the coordinates of the points in the prompt area.

Figure 1 shows an example of a crack front formed by a circular arc along which the direction of the **q** vector is constantly varying. In contrast, the normal to the crack plane is constant. As a result, for this case you should define the crack extension direction by specifying the normal to the crack plane.

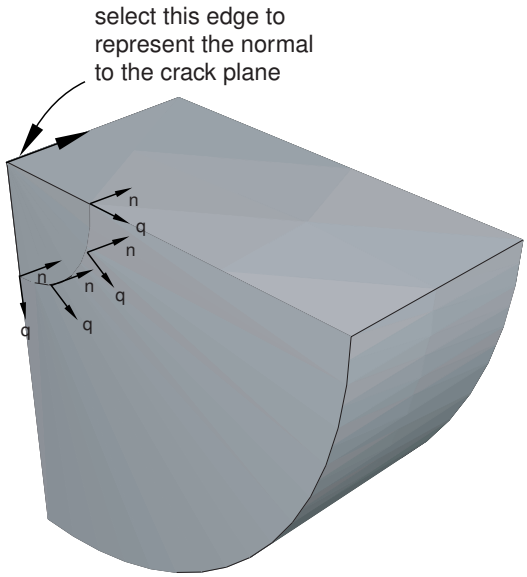


Figure 1: The direction of the q vector varies but the direction of the normal is constant.

Figure 2 shows a crack front formed by the edge of a truncated cone along which the directions of both the **q** vector and the normal to the crack plane are varying.

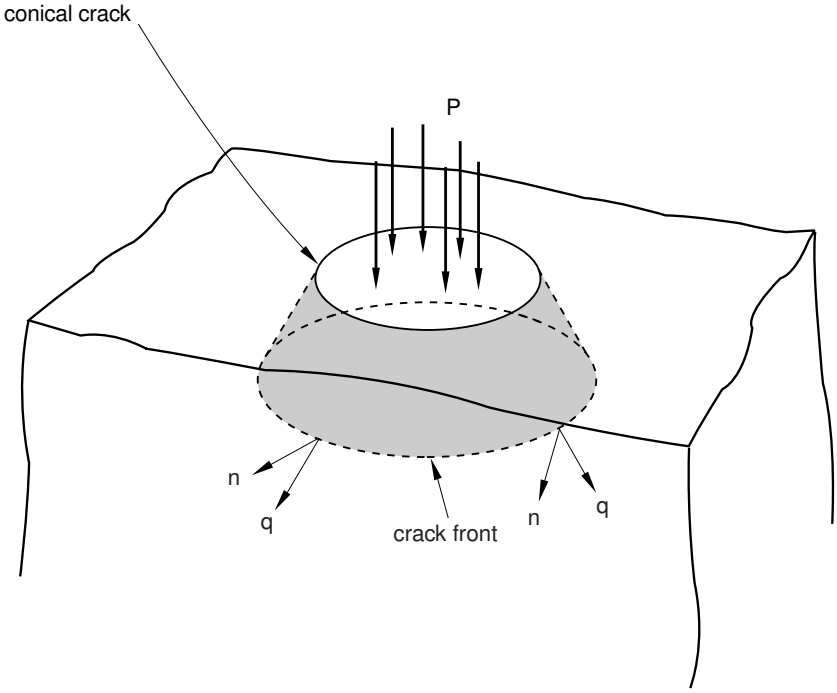


Figure 2: The directions of both the q vector and the normal vary along the crack front.

For this case you should do the following:

1. Define the contour integral, and use a single **q** vector to specify the crack extension direction.
2. Mesh the part, create a job, and write the model to an input file.

3. Import the input file, which will create an orphan mesh representation of the part.
4. Edit the crack that was imported with the model. Each node along the crack front will have the crack extension direction defined by the **q** vector that you specified in Step 1.
5. Use the Query toolset to determine the start and end coordinates of the **q** vector at each node, and edit the data defining the **q** vectors at each node along the crack front.

This technique is shown in *Figure 3*.

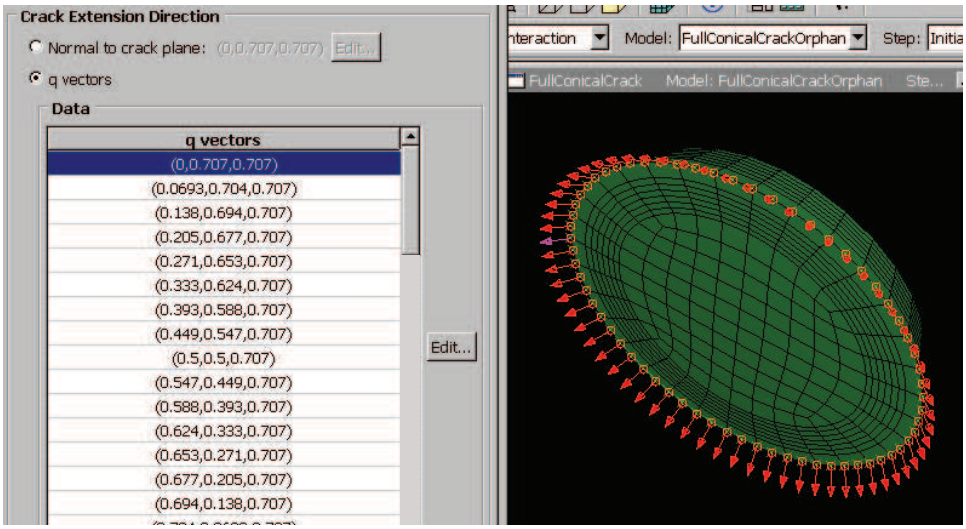


Figure 3: Entering the q vector at each node along the crack front.

Figure 3 is taken from *Contour integrals for a conical crack in a linear elastic infinite half space*. An Abaqus Scripting Interface script is provided with this example that illustrates how you can enter the **q** vector at each node along the crack front.

Controlling the singularity at the crack tip for a small-strain analysis

If the geometry of the crack region defines a sharp crack, the strain field becomes singular at the crack tip, as described in *Constructing a Fracture Mechanics Mesh for Small-Strain Analysis with the Conventional Finite Element Method*. Including the singularity in your model for a small-strain analysis improves the accuracy of the contour integral and the stress and strain calculations.

To include the singularity in your contour integral estimates, click the **Singularity** tab in the **Crack** editor and choose the desired position of the midside node and degenerate element control. For detailed instructions, see *Controlling the singularity at the crack tip*.

In addition, you must do the following in the Mesh module:

- If the assembly or part is two-dimensional, you must model the crack front with a ring of triangles and assign quadrilateral elements to the remainder of the contour integral region.
- If the assembly or part is three-dimensional, you must model the crack front with a ring of wedges and assign hexahedral elements to the remainder of the contour integral region.

When you mesh the crack front, Abaqus does the following:

- Converts the elements in the crack front to collapsed quadrilateral or hexahedral elements.
- If you modeled the crack front with a ring of second-order triangles or wedges, Abaqus moves the midside nodes to the specified position along the element edges that radiate out from the crack tip or crack line. (If you modeled the crack front with a ring of first-order triangles or wedges, Abaqus ignores the position that you specified for the midside nodes.)

A part instance that includes orphan mesh elements is always a dependent instance. As a result, you cannot adjust the midside nodes of these instances in the assembly. For more information, see *What is the difference between editing an orphan mesh, a meshed part, and a meshed part instance in the assembly?*. You must display the original orphan mesh and use the Edit Mesh toolset to adjust the position of the midsize nodes. For more information, see *Adjusting the position of midside nodes*. In addition, you cannot create collapsed elements from triangular and wedge elements of an orphan mesh.

Contour integral output

After you have defined the crack, you must use the **History Output Request** editor in the Step module to include the contour integral data in the output database generated by the analysis. The editor allows you to configure the following:

- The output frequency.
- The type of contour integral calculation to perform (*J*-integral, C_t-integral, stress intensity factors, or *T*-stress).
- The number of contours to evaluate.

Abaqus writes a history output variable to the output database for each contour integral that it calculates. *Figure 1* shows the format of the name of an output variable representing a contour integral.

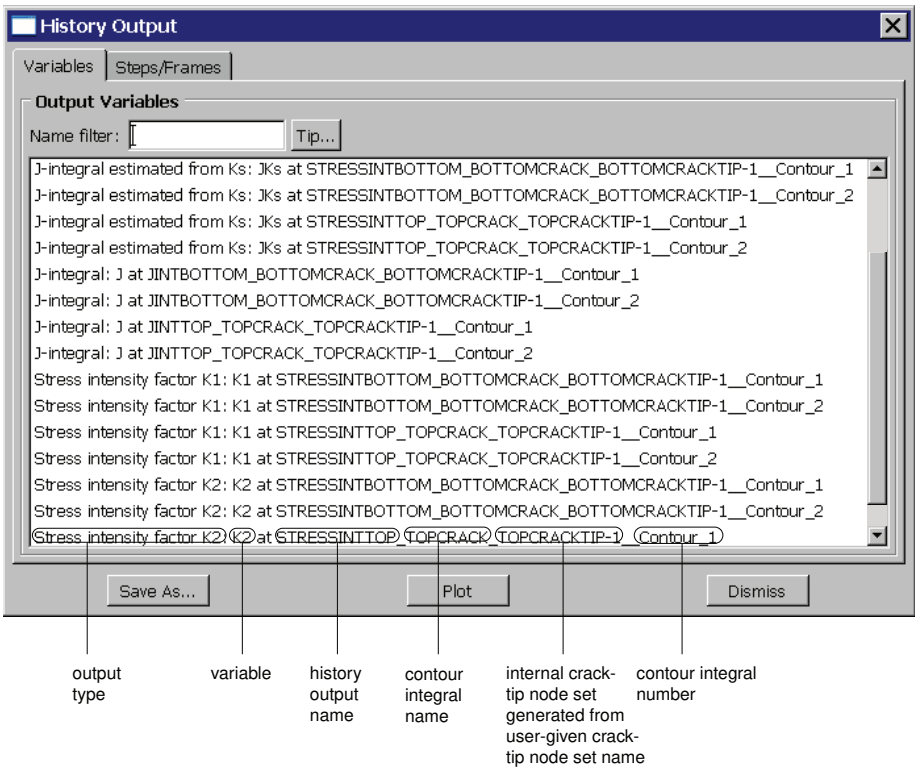


Figure 1: The format of a contour integral name in the output database.

For detailed instructions, see *Requesting contour integral output*. For more information, see *Modifying history output requests*, and *Contour Integral Evaluation*.

Meshing the crack region and assigning elements

Large stress concentrations occur at the crack tip. As a result, you should create a refined mesh around the crack tip to get accurate results for stresses and strains. In contrast, because the *J*-integral is an energy measure, you can obtain accurate *J*-values with a relatively coarse mesh. However, if the material becomes more nonlinear, you must create a finer mesh at the crack tip to maintain the accuracy of the *J*-values. You can control the density of the mesh at the crack tip by partitioning the region and by assigning mesh seeds to the resulting edges. For more information, see *Understanding seeding*.

For a large-strain analysis during which Abaqus will allow for nonlinear geometry, you should mesh the contour integral region with quadrilateral or hexahedral elements. For more information, see *Constructing a Fracture Mechanics Mesh for Finite-Strain Analysis with the Conventional Finite Element Method*.

However, for a small-strain analysis that does not allow for nonlinear geometry, you must allow for the singularity at the crack tip or crack line by meshing the region that defines the crack front with a ring of triangles or wedges. For more information, see *Controlling the singularity at the crack tip for a small-strain analysis*.

You must use the swept meshing technique to create wedge elements; however, there are limitations on the regions that Abaqus/CAE can mesh using the swept meshing technique, as described in *Swept meshing of three-dimensional solids*. As a result, if you cannot use the swept meshing technique, you cannot create wedge elements, and you cannot allow for the singularity at the crack line. In most cases you can ignore the singularity if your mesh is sufficiently refined to model the deformation around the crack tip or crack line and the resulting high strain gradients. You can also ignore the singularity if you are interested in only the contour integral output.

Controlling the singularity at the crack tip

If the geometry of the crack region defines a sharp crack, the strain field becomes singular at the crack tip, as described in *Constructing a Fracture Mechanics Mesh for Small-Strain Analysis with the Conventional Finite Element Method*. Including the singularity in your model for a small-strain analysis improves the accuracy of the contour integral and the stress and strain calculations.

1. Display the **Crack** editor using one of the following methods:
   - To configure a new contour integral, follow the instructions in *Creating a contour integral crack*.
   - To edit an existing contour integral, select **Special->Crack->Edit->*crack name*** from the main menu bar in the Interaction module.

2. To include the strain singularity in your contour integral estimates, click the **Singularity** tab in the **Crack** editor, and do one of the following:
   - To create a $1/\sqrt{r}$ strain singularity for an elastic fracture mechanics application:
     1. From the **Second-order Mesh Options** field, enter a value of 0.25 for the **Midside node parameter** to move the midside nodes to the $1/4$ points.
     2. From the **Degenerate Element Control at Crack Tip/Line** field, choose **Collapsed element side, single node**.
   - To create a $1/r$ strain singularity for a perfect plasticity fracture mechanics application:
     1. From the **Second-order Mesh Options** field, enter a value of 0.5 for the **Midside node parameter** to keep the midside nodes at the midside points.
     2. From the **Degenerate element control at Crack Tip/Line** field, choose **Collapsed element side, duplicate nodes**.
   - To create a combined $1/\sqrt{r}$ and $1/r$ strain singularity for a power law hardening material:
     1. From the **Second-order Mesh Options** field, enter a value of 0.25 for the **Midside node parameter**.
     2. From the **Degenerate element control at Crack Tip/Line** field, choose **Collapsed element side, duplicate nodes**.

3. Click **OK** to include the singularity in the contour integral estimation and to close the editor.



Note:

In most cases you must assign second-order triangle or wedge elements to the crack front region to include the singularity in your contour integral estimates. If you assign first-order elements, Abaqus ignores the value of the midside node setting in the **Singularity** tabbed page. However, if you assign first-order elements and choose **Collapsed element side, duplicate nodes** from the **Degenerate Element Control at Crack Tip/Line** field, you will create a $1/r$ strain singularity.

Creating a contour integral crack

You can use the **Special** menu in the Interaction module to create a contour integral crack. For more information, see *Fracture Mechanics*.

The entities that you select to define the contour integral depend on whether the part is two- or three-dimensional and whether you are defining the part using geometry or using orphan mesh elements and nodes.

1. From the main menu bar in the Interaction module, select **Special->Crack->Create**.
2. From the **Create Crack** dialog box that appears, select **Contour integral**.
3. Enter the name of the crack, and click **Continue** to close the dialog box.
4. From the model in the viewport, select the entities representing the crack front region. For a description of the entities to select, see *Defining the crack front*.
5. Click mouse button 2 to indicate that you have finished selecting the crack front region.
6. From the model in the viewport, select the entities representing the crack-tip region. In some cases, depending on the modeling space of your model and the entities that you selected to define the crack front, Abaqus/CAE selects the crack tip for you and skips to Step 7. For more information, see *Defining the crack tip or crack line*.

   From the prompt line, toggle on **Select mesh entities** to select entities from an orphan mesh.
7. Click mouse button 2 to indicate that you have finished selecting the crack-tip region.
8. From the prompt area, choose the method for defining the crack extension direction.

   ### Normal to crack plane

   Select **Normal to crack plane** to specify the normal to the crack plane, and select points representing the start and end of the normal.

   ### q vectors

   Select **q vectors** to specify the crack extension direction directly, and select points representing the start and end of the vector.

   For more information, see *Defining the crack extension direction*.

   Abaqus/CAE displays a blue arrow indicating the crack extension direction and, if specified, a red arrow indicating the normal to the crack plane and displays the **Edit Crack** dialog box.
9. Use the **Edit Crack** dialog box to configure the parameters that control a contour integral analysis.
   - Click the **General** tab of the dialog box to do the following:
     - Specify that the crack front is defined on a symmetry plane that models only half of the structure.
     - Change the entities that define the crack front or crack tip/line.
     - Change the method for defining the crack extension direction. You can also change the entities that define the crack extension direction and flip the crack extension direction (**q** vector).

     For more information, see *Modifying data for contour integrals*.
   - Click the **Singularity** tab of the dialog box to model a singularity of the strain field at the crack tip. For more information, see *Controlling the singularity at the crack tip*.
10. Click **OK** to configure the contour integral and to close the editor.

    Abaqus displays green crosses on the region to represent the crack front.

You must use the **History Output Request** editor in the Step module to include the contour integral data in the output database generated by the analysis. For more information, see *Contour integral output*, and *Modifying history output requests*.

Additional information

- *Using contour integrals to model fracture mechanics*
- *Fracture Mechanics*

Modifying data for contour integrals

If desired, you can modify the entities that you selected to define the crack front, the crack tip, and the crack extension direction that will be used to define a contour integral. You can also specify that the crack front is defined on a symmetry plane that models only half of the structure and flip the calculated crack extension direction, if desired. For more information, see *Symmetry with the Conventional Finite Element Method*.

1. Display the **Crack** editor using one of the following methods:
   - To configure a new contour integral, follow the instructions in *Creating a contour integral crack*.
   - To edit an existing contour integral, select **Special->Crack->Edit->*crack name*** from the main menu bar in the Interaction module.
2. Toggle on **On symmetry plane (half-crack model)** to specify that the crack front is defined on a symmetry plane that models only half of the structure. Abaqus doubles the change in potential energy calculated from the virtual crack front advance to compute the correct contour integral values.
3. If desired, click  to modify your selection of entities that define the crack front, crack tip, or crack line. If Abaqus/CAE selected the crack tip or crack line region to be the same as the crack front, you cannot edit the selection.
4. If desired, change the method for defining the crack extension direction. You can also click  to modify your selection of entities that define the crack extension direction.
5. If desired, you can click  to confirm or flip the calculated crack extension direction. In the prompt area, click **Yes** to confirm the **q**-vector direction or click **Flip** to reverse the direction.
6. Click **OK** to configure the contour integral and to close the editor.

   Abaqus displays green crosses on the region to represent the crack front.

   You must use the **History Output Request** editor in the Step module to include the contour integral data in the output database generated by the analysis. For more information, see *Contour integral output*, and *Modifying history output requests*.

Requesting contour integral output

You use the **History Output Request** editor in the Step module to request output from the contour integral analysis. You must select **Crack** from the **Domain** field. The editor then allows you to select the output frequency and the type of contour integral analysis to perform. For more information, see *Modifying history output requests*, and *Contour Integral Evaluation*.

You can also request contour integral output for an XFEM fracture analysis. For more information, see *Requesting contour integral output for XFEM*.

1. Enter the Step module.
2. From the main menu bar, select **Output->History Output Request->Create**.

   Abaqus/CAE displays the **Create History** dialog box.
3. From the **Create History** dialog box, enter the name of the output request and the step in which it will be created. Click **Continue** to close the dialog box.

   Abaqus/CAE displays the **History Output Requests** editor.
4. From the **Domain** field, select **Crack**, and select the desired contour integral.
5. From the fields that appear in the **History Output Requests** editor, enter the output frequency.
6. Enter the number of contours to evaluate. Abaqus/CAE computes the next contour integral by adding a single layer of elements to the region defined by the previous domain, as described in *Figure 1*. Each contour generates a value or a group of values for the contour integral.
7. If desired, toggle on **Step for residual stress initialization values**, and select the step from which Abaqus will read the stress data in the last available increment and use the data to calculate the residual stresses. If you select the initial step, the residual stresses are defined by the specified initial conditions. You cannot select a step that occurs after the step in which the history output request is created. For more information, see *Including the Effect of a Residual Stress Field on J-Integral Evaluation*.
8. Select the type of contour integral calculation to perform. You can select from the following:

J-integral

You use the *J*-integral in rate-independent quasi-static fracture analysis to characterize the energy release associated with crack growth. The *J*-integral can be related to the stress-intensity factor if the material response is linear. For more information, see *J-Integral*.

C_t-integral

You use the C_t-integral for time-dependent creep behavior, where it characterizes creep crack deformation under certain creep conditions, including transient crack growth. For more information, see *Ct-Integral*.

T-stress

You use the *T*-stress component to represent a stress parallel to the crack front. For more information, see *T-Stress*.

Stress intensity factors

You use the stress intensity factors K_I, K_{II}, and K_{III} in linear elastic fracture mechanics to characterize the local crack-tip stress and displacement fields. For more information, see *Stress Intensity Factors*.

If you request a contour integral calculation using stress intensity factors, Abaqus also computes the crack propagation direction at initiation. You must choose one of the following to indicate the crack initiation criteria:

- **Maximum tangential stress**
- **Maximum energy release rate**
- $\mathbf{K_{II} = 0}$

For more information, see *Crack Propagation Direction*.

9. Click **OK** to configure the contour integral analysis and to close the editor.

   For examples of how Abaqus names history output variables in the output database for each contour integral that it calculates, see *Contour integral output*.

Using the extended finite element method to model fracture mechanics

You can use the extended finite element method (XFEM) to study the initiation and propagation of a crack along an arbitrary, solution-dependent path without needing to remesh your model.
XFEM is available for three-dimensional solid and two-dimensional planar models; three-dimensional shell models are not supported.

In this section:

- *The extended finite element method (XFEM)*
- *Choosing the type of XFEM analysis*
- *Viewing an XFEM crack*
- *Creating an XFEM crack*
- *Deactivating and activating an XFEM crack growth*
- *Specifying a contact interaction property for XFEM*
- *Requesting contour integral output for XFEM*

The extended finite element method (XFEM)

You can study the onset and propagation of cracking in quasi-static problems using the extended finite element method (XFEM). XFEM allows you to study crack growth along an arbitrary, solution-dependent path without needing to remesh your model.
XFEM is available only for three-dimensional solid and two-dimensional planar models; three-dimensional shell models are not supported. You can use XFEM to study a crack in parts containing geometry, orphan mesh elements, or a combination of the two. You can choose to study a crack that grows arbitrarily through your model or a stationary crack. You define an XFEM crack in the Interaction module. You can specify the initial location of the crack. Alternatively, you can allow Abaqus to determine the location of the crack during the analysis based on the value of the maximum principal stress or strain calculated in the crack domain. For more information, see *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*. Examples of XFEM models created in Abaqus/CAE are provided in *Modeling discontinuities using XFEM*.

To perform an XFEM crack analysis, you must specify the following:

Crack domain

To define the crack domain, you can select one or more cells from three-dimensional parts or one or more faces from two-dimensional planar parts. If you are defining the crack domain on an orphan mesh or a part containing both orphan and native mesh elements, you can select elements. The crack domain includes regions that contain any existing cracks and regions in which a crack might be initiated and into which a crack might propagate.

After you define the initial crack location, you can reduce the size of the crack domain by indicating that you want to use only elements in the vicinity of the crack geometry. You specify the number of element layers to include to prescribe a minimal enrichment zone automatically. The crack domain is defined using the elements intersected by the crack location. These elements are the initial set. The enrichment zone is built of elements in the neighborhood of the initial crack using the number of element layers that you specify. You can highlight the elements in the viewport and, if desired, save the highlighted elements to a set.

Crack growth

You can allow the crack to propagate along an arbitrary, solution-dependent path, or you can specify that the crack is stationary.

Initial crack location

To define the initial crack location, you can select faces from a three-dimensional solid or edges from a two-dimensional planar model. The initial crack location must be contained within the crack domain. A selected face can be a face of the solid, a face created by a partition, or a planar part instance. Similarly, a selected edge can be an edge of the solid, an edge created by a partition, or a wire part instance; you should not select a seam crack. You should not mesh the faces or edges that you selected to define the initial crack location. *Figure 1* shows examples of the crack domain and the crack location for two- and three-dimensional geometry and orphan meshes.

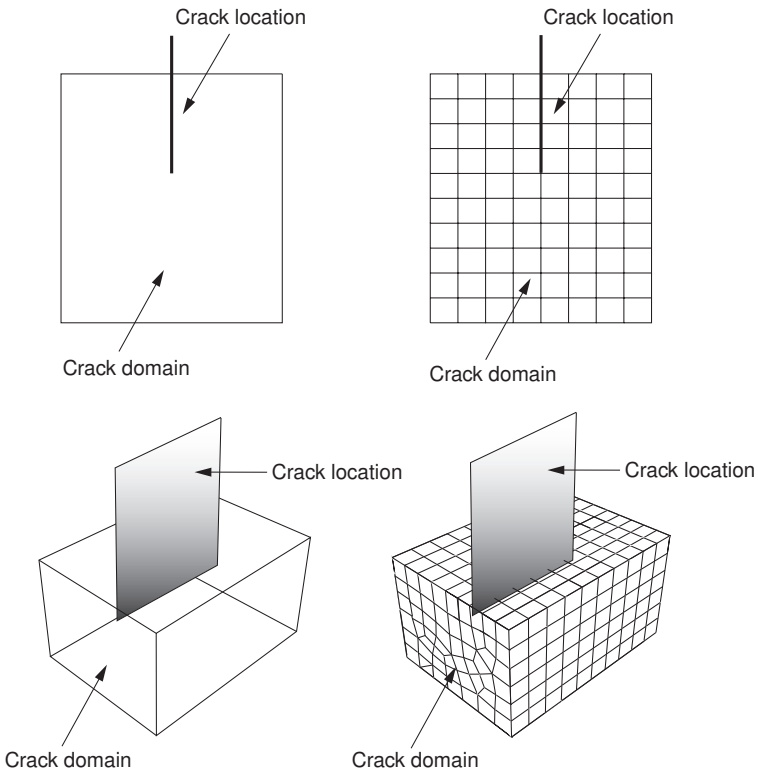


Figure 1: Defining a crack for XFEM.

Alternatively, you can choose not to define the initial crack location. Regardless of whether you define the initial crack location, Abaqus initiates the creation of cracks during the simulation by searching for regions that are experiencing principal stresses and/or strains greater than the maximum damage values specified by the traction-separation laws.

Enrichment radius

The enrichment radius is a small radius from the crack tip within which the elements will be used for calculating crack singularity for a stationary crack. Elements within the enrichment radius must be included in the cells or faces that you chose to represent the crack domain. You can allow Abaqus to calculate the radius (six times the typical element characteristic length in the enriched area), or you can specify its value.

Contact interaction property

You can choose to associate a contact interaction property with the XFEM crack that defines the contact of cracked element surfaces. For detailed information, see *Specifying a contact interaction property for XFEM*.

Damage initiation

You must specify the conditions that will initiate a crack by specifying damage initiation criteria in the material definition. You can specify a criterion based on either maximum principal stress or maximum principal strain. For more information, see *Maximum principal stress or strain damage*.

Analysis procedure

You can include an XFEM crack in a static analysis procedure. Alternatively, you can include an XFEM crack in an implicit dynamic analysis procedure to simulate the fracture and failure in a structure under high-speed impact loading. The XFEM-based crack propagation simulated in an implicit dynamic procedure can also be preceded or followed by a static procedure to model the damage and failure throughout the loading history.

For detailed instructions, see *Creating an XFEM crack*.

Choosing the type of XFEM analysis

Abaqus provides the traction-separation cohesive behavior approach and the linear elastic fracture mechanics (LEFM) approach for studying crack initiation and propagation using XFEM.

Traction-separation cohesive behavior

The traction-separation cohesive behavior approach is described in detail in *Modeling Moving Cracks with the Cohesive Segments Method and Phantom Nodes*. You can specify the material properties that define the evolution of damage leading to eventual failure. You apply the material to the section that is assigned to the crack domain. You can choose to associate a normal behavior contact interaction property with the XFEM crack that defines the contact of cracked element surfaces. For detailed information, see *Specifying a contact interaction property for XFEM*. To assist convergence as the material fails, you can introduce localized damping using the viscous regularization technique. For more information, see *Damage stabilization*.

Linear elastic fracture mechanics (LEFM)

The linear elastic fracture mechanics (LEFM) approach uses the modified Virtual Crack Closure Technique (VCCT) to calculate the strain energy release rate at the crack tip. The approach is more appropriate for brittle fracture problems and is described in detail in *Applying the VCCT Technique to the XFEM-Based LEFM Approach*. To use this approach, you must create a fracture criterion contact interaction property, as described in *Specifying fracture criterion properties for crack propagation*. If you specify localized damping in the contact interaction property, Abaqus will use the viscous regularization technique to assist convergence as the material fails. For more information, see *Specifying fracture criterion properties for crack propagation*.

Viewing an XFEM crack

When you are creating your model, you must request field output for the signed distance function, PHILSM. When you open an output database file containing the PHILSM output variable, Abaqus/CAE creates a view cut along the XFEM crack (where the value of the signed distance function is zero). Therefore, when you display a deformed or contour plot, the cracked elements in the enriched region are visible, as shown in *Figure 1*.

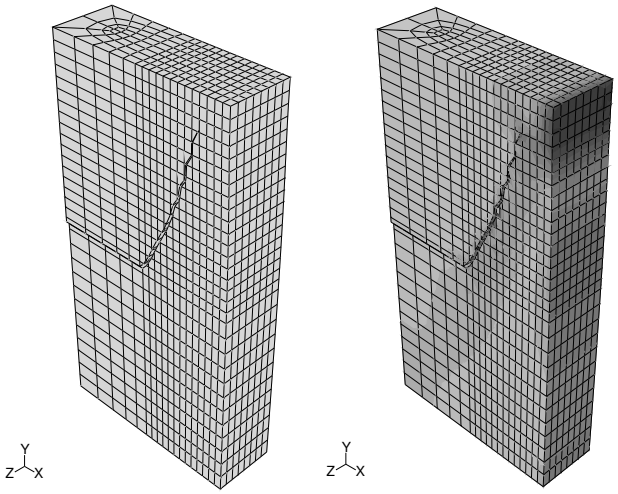


Figure 1: Viewing an XFEM crack.

You can perform a time history animation of a contour plot and view the onset of damage calculated by XFEM. The animation also allows you to view the propagation of the damage through the enriched region as the analysis progresses. Turning on translucency allows you to see the progression of the crack through interior elements. For more information, see *Changing the translucency*.

To view the initial crack front, you can create an isosurface cut shape with PSILSM selected as the current primary field output variable. For more information, see *Creating or editing a view cut*.

If you want to investigate the crack in more detail, you can create a contour plot of the signed distance function (PHILSM) and determine in which elements the signed distance values are negative or positive. The crack surface is situated in the elements where the value of PHILSM transitions from a negative number to a positive number. In addition, you can use the  option in the **View Cut Manager** to view only the surface where the value of the signed distance function is zero, which corresponds with the surface of an XFEM crack. For more information, see *Understanding view cuts*.

Creating an XFEM crack

You can use the **Special** menu in the Interaction module to create an extended finite element method (XFEM) crack.

1. From the main menu bar in the Interaction module, select **Special->Crack->Create**.
2. From the **Create Crack** dialog box that appears, select **XFEM**.
3. Enter the name of the crack, and click **Continue** to close the dialog box.
4. From the model in the viewport, select the entities representing the crack domain. You can select cells from a three-dimensional part instance or faces from a two-dimensional part instance. If you have an orphan mesh or an instance containing both orphan mesh and native mesh elements, you can select elements to represent the crack domain. You should select the entities that contain an existing crack along with any entities into which a crack might propagate.
5. Click mouse button 2 to indicate that you have finished selecting the crack domain.

   The **Edit Crack** dialog box appears.
6. You can reduce the size of the crack domain. This option is available only if you specify the crack location. Do the following:
   a. Toggle on **Shrink crack domain using crack location**.
   b. Specify the number of element layers to include to prescribe a minimal enrichment zone automatically.
   c. Click **Preview** to highlight the elements in the viewport and, if desired, save the highlighted elements to a set.
7. Do either of the following:
   - Toggle on **Allow crack growth** to define a crack that grows along an arbitrary path through your model as the solution progresses.
   - Toggle off **Allow crack growth** to define a stationary crack that cannot grow.
8. If you chose to allow crack growth, you can do either of the following to specify the crack location:
   - Toggle off **Crack location**, indicating that you will allow Abaqus to determine the location of the crack based on the damage initiation criterion that you specified.
   - Toggle on **Crack location** and click  to specify the crack location by selecting interior faces from a three-dimensional model or edges from a two-dimensional planar part. You should not select a seam crack.
9. If you chose to prevent crack growth, do the following:
   a. Click  to specify the crack location by selecting interior faces from a three-dimensional model or edges from a two-dimensional planar part. You should not select a seam crack.
   b. To specify the enrichment radius, do either of the following:
      - Choose **Analysis default**, to allow Abaqus to determine the enrichment radius. The default radius is three times the typical element characteristic length in the enriched area.
      - Choose **Specify** and enter a value. The value should be the radius from the crack tips within which the elements are used to calculate the crack singularity.
10. Choose the type of XFEM analysis:
    - By default, Abaqus will use the traction-separation cohesive behavior approach. You can select or create a contact interaction property that specifies the compressive and frictional behavior of the

cracked faces based on a small-sliding contact formulation. For more information, see *Defining surface-to-surface contact*.

- If you create a fracture criterion contact interaction property, Abaqus will use the linear elastic fracture mechanics (LEFM) approach. For more information, see *Specifying fracture criterion properties for crack propagation*.

11. Click **OK** to configure the XFEM crack and to close the editor.

Abaqus displays green crosses to represent the crack domain and the crack location.

To view the crack growth in the Visualization module, you must use the **Field Output Request** editor in the Step module and request that Abaqus writes the signed distance function PHILSM to the output database during the analysis. For more information, see *Viewing an XFEM crack*, and *Modifying field output requests*.

Additional information

- *Fracture mechanics*
- *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*

Deactivating and activating an XFEM crack growth

By default, XFEM crack growth is active in all steps. However, if an analysis that includes XFEM crack growth fails to converge, you can deactivate the crack in a selected step and allow the analysis to reach equilibrium before reactivating the crack in a subsequent step.

1. Create an interaction in the initial step. For more information, see *Creating interactions*.
2. Select an interaction type of **XFEM crack growth**, and click **Continue**.
3. From the **Edit Interaction** dialog box, select the XFEM crack that you want to activate or deactivate. You can select only cracks that allow crack growth.
4. Click **OK** to create the interaction.
5. Open the **Interaction Manager**. For more information, see *Managing objects in the Interaction module*.
6. Select the XFEM crack growth interaction in the desired step, and click **Edit**.
7. From the **Edit Interaction** dialog box, toggle off (or on) **Allow crack growth** to deactivate (or activate) the XFEM crack growth in the selected step, and click **OK**.

   The **Interaction Manager** displays **Modified** to indicate the status of the XFEM crack growth in that step. Abaqus propagates the state of the XFEM crack growth (activated or deactivated) to subsequent steps.

Specifying a contact interaction property for XFEM

When you are configuring an XFEM analysis, you can select a contact interaction property that defines the compressive behavior of the crack surfaces. This section describes the settings in the contact interaction property that apply to XFEM crack growth. For more information, see *Understanding interaction properties*.

1. From the main menu bar, select **Interaction->Property->Create**.
2. In the **Create Interaction Property** dialog box that appears, do the following:
   - Name the interaction property.
   - Select the **Contact** type of interaction property.
3. Click **Continue** to close the **Create Interaction Property** dialog box.
4. From the menu bar in the contact property editor, select **Mechanical->Normal Behavior**.
5. From the **Constraint enforcement method** field, select **Penalty (Standard)** to enforce contact constraints using the penalty method.
6. Toggle off **Allow separation after contact** if you want to prevent surfaces from separating once they have come into contact.
7. Choose the **Linear** enforcement behavior in the **Behavior** field to use the linear penalty method for the enforcement of the contact constraint.
8. Specify the contact stiffness in the **Stiffness value** field.
   - Choose **Use default** to have Abaqus calculate the penalty contact stiffness automatically.
   - Choose **Specify**, and enter a positive value for the linear penalty stiffness.
9. Specify a factor by which to multiply the chosen penalty stiffness in the **Stiffness scale factor** field.
10. Specify the **Clearance at which contact pressure is zero**. The default value is 0.
11. Click **OK** to create the contact property and to exit the **Edit Contact Property** dialog box.

Requesting contour integral output for XFEM

You use the **History Output Request** editor in the Step module to request contour integral output from the XFEM crack. You must select **Crack** from the **Domain** field. The editor then allows you to select the output frequency and the type of contour integral analysis to perform. For more information, see *Modifying history output requests*, and *Contour Integral Evaluation*.

1. Enter the Step module.
2. From the main menu bar, select **Output->History Output Request->Create**.

   Abaqus/CAE displays the **Create History** dialog box.
3. From the **Create History** dialog box, enter the name of the output request and the step in which it will be created. Click **Continue** to close the dialog box.

   Abaqus/CAE displays the **History Output Requests** editor.
4. From the **Domain** field, select **Crack**, and select the desired XFEM crack.
5. Select the frequency at which the history output will be generated.
6. Enter the number of contours to evaluate. Abaqus/CAE computes the next contour integral by adding a single layer of elements to the region defined by the previous domain. Each contour generates a value or a group of values for the contour integral.
7. If desired, toggle on **Step for residual stress initialization values** and select the step from which Abaqus will use the stress data in the last available increment to extract the residual stresses. If you select the initial step, the residual stresses are defined by the specified initial conditions. You cannot select a step that occurs after the step in which the history output request is created. For more information, see *Including the Effect of a Residual Stress Field on J-Integral Evaluation*.
8. Select the type of contour integral calculation to perform. You can select from the following:

J-integral

You use the *J*-integral in rate-independent quasi-static fracture analysis to characterize the energy release associated with crack growth. The *J*-integral can be related to the stress-intensity factor if the material response is linear. For more information, see *J-Integral*.

C_t-integral

You use the C_t-integral for time-dependent creep behavior, where it characterizes creep crack deformation under certain creep conditions, including transient crack growth. For more information, see *Ct-Integral*.

T-stress

You use the *T*-stress component to represent a stress parallel to the crack front. For more information, see *T-Stress*.

Stress intensity factors

You use the stress intensity factors K_I, K_{II}, and K_{III} in linear elastic fracture mechanics to characterize the local crack-tip stress and displacement fields. For more information, see *Stress Intensity Factors*.

If you request a contour integral calculation using stress intensity factors, Abaqus also computes the crack propagation direction at initiation. You must choose one of the following to indicate the crack initiation criteria:

- **Maximum tangential stress**
- **Maximum energy release rate**
- $\mathbf{K_{II} = 0}$

For more information, see *Crack Propagation Direction*.

9. Click **OK** to configure the contour integral analysis and to close the editor.

   For examples of how Abaqus names history output variables in the output database for each contour integral that it calculates, see *Contour integral output*.

Using the virtual crack closure technique to model crack propagation

You can use the virtual crack closure technique (VCCT) to study the initiation and propagation of a crack along a known crack surface.
Modeling crack propagation using VCCT is available only for Abaqus/Standard models (three-dimensional solid and shell and two-dimensional planar and axisymmetric models).

In this section:

- *The virtual crack closure technique*
- *Creating a VCCT crack for Abaqus/Standard*

The virtual crack closure technique

You can study the onset and propagation of cracking in quasi-static problems using the virtual crack closure technique (VCCT). VCCT uses the principles of linear elastic fracture mechanics (LEFM) , so it is appropriate for problems in which brittle crack propagation occurs along predefined surfaces.
VCCT is based on the assumption that the strain energy released when a crack is extended by a certain amount is the same as the energy required to close the crack by the same amount.

You can include a VCCT crack in a static or quasi-static analysis procedure. Alternatively, you can include a VCCT crack in an implicit dynamic analysis procedure to simulate the fracture and failure in a structure under high-speed impact loading. VCCT is available only for Abaqus/Standard (three-dimensional solid and shell and two-dimensional planar and axisymmetric models). You can use VCCT to study a crack in parts containing geometry, orphan mesh elements, or a combination of the two. You define a VCCT crack in the Interaction module. You can specify the location of the surfaces that are initially bonded. For more information, see *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*.

Creating VCCT or enhanced VCCT crack propagation models that converge to a successful solution requires some understanding of the principles behind VCCT. See *Tips for Using the VCCT or Enhanced VCCT Criterion in Abaqus/Standard*, for information that will help you create models that can be analyzed successfully.

For detailed instructions, see *Creating a VCCT crack for Abaqus/Standard*.

Creating a VCCT crack for Abaqus/Standard

You can create a virtual crack closure technique (VCCT) crack that can be analyzed by Abaqus/Standard by doing the following:

- Create a contact interaction property that specifies the fracture criterion. The fracture criterion interaction property specifies the criterion for crack propagation along initially partially bonded surfaces. You can choose between the following crack propagation criteria:
  - The virtual crack closure technique (**VCCT**) criterion, which uses the principles of linear elastic fracture mechanics. For more information, see *VCCT Criterion*.
  - The enhanced virtual crack closure technique (**Enhanced VCCT**) criterion, in which you can control the onset and growth of a crack using two different critical fracture energy release rates. The enhanced virtual crack closure technique is available only in an Abaqus/Standard analysis. For more information, see *Enhanced VCCT Criterion*.
- Create a surface-to-surface contact interaction that models the potential crack surfaces using main and secondary contact surfaces. The initially bonded region defines a region of the secondary surface that is initially bonded with the main surface. The unbonded portion of the secondary surface behaves as a regular contact surface. For detailed information, see *Defining Initially Bonded Crack Surfaces in Abaqus/Standard*.
- Activate the crack propagation capability to specify that crack propagation can occur between the two surfaces that are initially partially bonded. The crack continues to propagate along the interface between the main and secondary surfaces. The virtual crack closure technique cannot model crack initiation from a surface that is not already cracked; therefore, you must specify a clearance between the secondary and main surfaces that models a preexisting flaw at the beginning of the crack surface.

  For more information, see *Crack Propagation Analysis*.

1. Create a contact interaction property that defines the mechanical fracture criterion, as described in *Specifying fracture criterion properties for crack propagation*.
   a. Choose the crack propagation criteria (**VCCT** or **Enhanced VCCT**).

   Note:

   The **Direction of crack growth relative to 1–direction** option applies only to enriched regions that will be used by the extended finite element method (XFEM) for crack propagation.

   b. Specify the critical energy release rates for crack onset and propagation.
2. Create a surface-to-surface contact interaction that specifies the main and secondary surfaces (the region that contains the potential crack surfaces), as described in *Defining surface-to-surface contact in an Abaqus/Standard analysis*.
   a. Select a discretization method of **Surface to surface**.
   b. From the **Bonding** tabbed page, toggle on **Limit bonding to secondary nodes in subset**, and choose the set that specifies the initially bonded nodes of the secondary surface. The unbonded portion of the secondary surface behaves as a regular contact surface.
   c. Select the contact interaction property that defines the mechanical fracture criterion.
3. Activate the **VCCT** crack propagation capability.
   a. From the Interaction module main menu, select **Special->Crack->Create**.
   b. From the **Create Crack** dialog box that appears, select **Debond using VCCT**.
   c. Enter the name of the crack, and click **Continue** to close the dialog box.

The **Edit Crack** dialog box appears.

d. From the **Edit Crack** dialog box, do the following:

1. Select the step in which the crack propagation is initiated.
2. Select the surface-to-surface contact interaction that specifies the main and secondary surfaces.
3. Specify how the traction between the surfaces defined in the contact pair interaction is released after debonding occurs:
   - Choose **Step** (default) to release the traction during the increment that follows debonding.
   - To avoid a sudden loss of stability, choose **Ramp** to release the traction gradually during succeeding increments that follow debonding.
4. You can specify the rate at which crack propagation information is written to the data (`.dat`) file. If the step is using automatic time incrementation, the value is the suggested time increment to use for the first increment just after debonding starts. If the step is using fixed time incrementation, the value is the time increment after debonding starts if Abaqus/Standard finds it needs a smaller time increment than its current value. The time increment size will be modified as required until debonding is complete.

e. Click **OK** to configure the VCCT crack and to close the editor.

4. If desired, edit the general solutions control and adjust the **VCCT Linear Scaling** to accelerate convergence. For most crack propagation simulations using VCCT or the enhanced VCCT criterion, the deformation can be nearly linear up to the point of the onset of crack growth; past this point the analysis becomes very nonlinear. In this case linear scaling can be used to effectively reduce the solution time to reach the onset of crack growth. For more information, see *Customizing general solution controls*.

Additional information

- *Fracture mechanics*
- *Crack Propagation Analysis*

Managing cracks

The **Crack Manager** allows you to create and manage cracks. The manager includes a list of the names and types of cracks that you have defined. The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the manager allow you to create new cracks or to edit, copy, rename, and delete existing ones. The icons in the column along the left side of the manager allow you to suppress and resume existing cracks. You can also initiate these procedures using the **Special->Crack** menu from the main menu bar in the Interaction module. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

Additional information

- *Suppressing and resuming objects*
- *Fracture mechanics*

Gaskets

This section provides information on how to model gasket behavior.

In this section:

- *Modeling gaskets*
- *Defining materials for gaskets*
- *Assigning gasket elements to a region*

Modeling gaskets

Gaskets are thin sealing components that are positioned between structural components.
(For detailed information on gasket theory, see *Gasket Elements*.) The general procedure for modeling gaskets in three-dimensional space involves the following steps:

1. In the Part module, define the solid geometry. Gasket parts are typically very thin, flat solids.
2. In the Property module, define a gasket material. The material can be a regular material or one that includes special gasket behaviors. See *Defining materials for gaskets*, for more information.
3. In the Property module, define a gasket section that refers to the gasket material. Then assign the gasket section to the gasket region.
4. In the Interaction module, establish appropriate tie constraints or contact interactions between the gasket surfaces and the surfaces of adjacent regions.
5. In the Mesh module, assign the top-down swept meshing technique or the bottom-up meshing technique to the gasket region. Regardless of the meshing technique that you choose, you must sweep, extrude, or revolve the mesh in a direction normal to the gasket plane to produce gasket elements with the correct orientation. See *Specifying the sweep path*, or *Bottom-up meshing*, for more information.
6. In the Mesh module, assign a gasket element type to the gasket region, and mesh the region.

When you model gaskets with solids, you can use one or a combination of the following techniques to define the interaction between the gaskets and surrounding regions:

- You can create a separate gasket part and then use tie constraints or contact interactions to couple the gasket part instance to the other part instances.
- You can create a thin region within a part and then assign a gasket section and element type to that region. This approach is recommended if compatibility between the gasket mesh and the meshes of adjacent regions is important.

Additional steps are necessary if the gasket model is composed of several layers and inserts. For example, *Figure 1* illustrates a gasket modeled as a solid layer with an embedded shell-like insert.

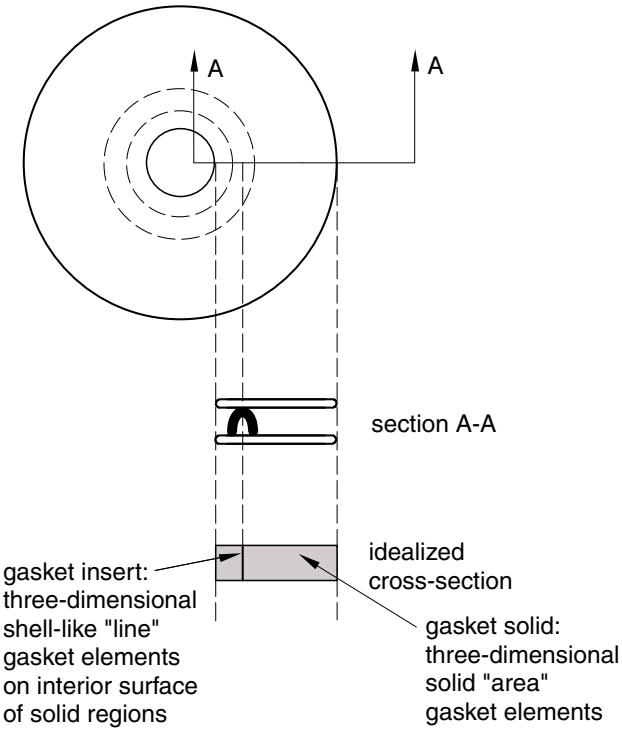


Figure 1: Idealized gasket model.

The Abaqus/CAE representation of the idealized gasket model is shown in *Figure 2*.

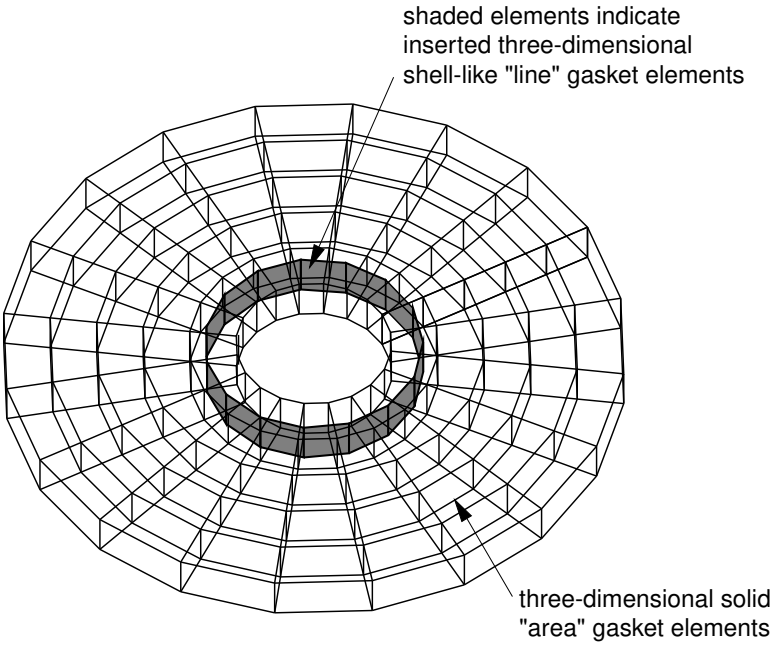


Figure 2: A three-dimensional view of the gasket model and its elements.

If you are working with gaskets composed of several layers and inserts, you must perform the following additional tasks:

1. Use the Partition toolset to partition the solid gasket region so that an internal surface is created at the position of the insert.
2. In the Property module, define a skin reinforcement on the internal surface that represents the insert. (See *Defining skin reinforcements*, for more information.) The gasket sections that you assign to the solid and to the insert are usually different, as are their materials.
3. No meshing is required (or allowed) for the insert skin, but you must assign a three-dimensional “line” gasket element type to the skin in the Mesh module. See *Assigning element types to skin or stringer reinforcements*, for more information.

Defining materials for gaskets

You can create two types of materials to include in gasket section definitions: materials with gasket-specific behaviors and general-use materials. The type of material that you create depends on your requirements for the gasket behavior.

- Create a material using the special gasket behaviors if you want the thickness-direction, transverse shear, and membrane behaviors to be uncoupled. Gasket behavior materials are valid only for gasket sections. For detailed information on this approach to defining gasket behavior, see *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*.
- Create a general-use material if you want to consider only the thickness-direction behavior. General-use materials are valid in gasket sections as well as in other types of sections. For detailed information on this approach to defining gasket behavior, see *Defining the Gasket Behavior Using a Material Model*.

You create a gasket-specific material by entering data for one or more of the behaviors found in the **Other->Gasket** submenu. Data entered for any other behavior in the material editor are ignored, with the following exceptions:

- You can include the **Expansion** behavior (located in the **Mechanical** menu) in a gasket behavior material definition.
- You can include the **Creep** behavior (located in the **Mechanical->Plasticity** menu) in a gasket behavior material definition.
- You can include the **Depvar** and **User Output Variables** behaviors (located in the **General** menu) in a gasket behavior material definition.

You create a general-use material by entering data for any behaviors that are valid for gasket sections except those found in the **Other->Gasket** submenu. (If you enter data for a behavior found in the **Other->Gasket** submenu, you automatically create a gasket behavior material.) For information on which material behaviors are valid for general-use materials included in gasket section definitions, see *Gasket Elements*.

Assigning gasket elements to a region

If the region is a shell, you must assign the swept meshing technique; if the region is a solid, you can assign any meshing technique. For solid regions you can define a stack orientation that is independent of the mesh technique. (For more information, see *Applying a mesh stack orientation*.) The swept meshing technique ensures that the elements are stacked in a manner suitable for gasket modeling. When you sweep mesh shell regions or solid regions that have not been assigned a stack orientation, the axis of each gasket element is coincident with the sweep path direction; therefore, you can control gasket element alignment by specifying an appropriate sweep path.

For example, the part instance shown in *Figure 1* has three possible sweep paths, and each path has two possible sweep directions. If you mesh this part instance using gasket elements, you can choose from six possible gasket axis orientations. For more information, see *Swept meshing*, and *Specifying the sweep path*.

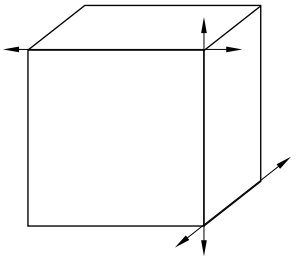


Figure 1: For shell regions, you must choose a sweep path and direction that provides an appropriate gasket axis orientation for your model.

In addition, you can assign gasket elements to orphan mesh elements. You can use the Query toolset to verify that the gasket elements are aligned correctly for regions where the gasket axis has more than one possible orientation. If necessary, you can use the Edit Mesh toolset to change the mesh stack orientation. For more information, see *Orienting the stack direction*. You can use these tools with hexahedral, wedge, and quadrilateral elements because these are the only elements that can be stacked to form a continuum shell or gasket mesh. If you have assigned second-order gasket elements to the region, you must first change the assignment to conventional elements before you reorient the stack direction. You can then convert the region back to a gasket mesh by reassigning second-order gasket elements.

Imperfection

This section provides information on how to model different types of imperfection. You can define imperfection in the Interaction module.

In this section:

- *Defining imperfections*
- *Managing imperfections*
- *Defining file imperfections*
- *Defining input imperfections*
- *Defining data imperfections*
- *Editing the region to which a file imperfection applied*

Defining imperfections

You can define an imperfection for an assembly in the Interaction module.

You can specify the following types of imperfections:

File imperfection

You can define a file imperfection that uses results files from a previous analysis on an assembly. For more information, see *Introducing a Geometric Imperfection into a Model*.

Input imperfection

You can define an input imperfection that uses imperfection data specified in an alternate input file on an assembly. For more information, see *Introducing a Geometric Imperfection into a Model*.

Data imperfection

You can define a data imperfection that uses data lines in the input file specifying components of imperfection for a given node on an assembly. For more information, see *Introducing a Geometric Imperfection into a Model*.

Select **Special->Imperfection->Create**: ***type*** from the Interaction module to define imperfection. Select **Edit** from the same menu to make changes to an existing definition.

Imperfection definitions appear in the Model Tree in the **Engineering Features** container under the assembly.

When you define a file imperfection, you can select objects from the viewport to identify the region on which to apply the imperfection. By default, a node set is created that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.

For detailed instructions on creating this type of engineering feature, see the following sections:

- *Defining file imperfections*
- *Defining input imperfections*
- *Defining data imperfections*

Managing imperfections

The **Imperfection Manager** allows you to create and manage imperfection definitions.

The manager includes a list of the names and types of imperfections that you have defined. The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the manager allow you to create new imperfection definitions or to edit, copy, rename, and delete existing ones.

The icons in the column on the left side of the manager allow you to suppress and resume existing imperfection definitions. You can also initiate these procedures using the **Special->Imperfection** menu from the main menu bar in the Interaction module.

After you select a management operation from the main menu bar, the procedure is the same as if you had clicked the corresponding button inside the manager dialog box.

Additional information

- *Suppressing and resuming objects*
- *Defining file imperfections*
- *Defining input imperfections*
- *Defining data imperfections*
- *Editing the region to which a file imperfection applied*

Defining file imperfections

You can define file imperfections on an assembly.

1. From the main menu bar in the Interaction module, select **Special->Imperfection->Create**.
2. In the **Create Imperfection** dialog box that appears, name the imperfection, select **File Imperfection**, and click **Continue**.
3. Select the points for which you want to define a file imperfection using one of the following methods:
   - Select the points in the viewport. When you have finished selecting, click mouse button 2.
   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit Imperfection** dialog box appears.
4. Select **Result file**, which is an `.odb` file that is to be used in the imperfection definition.
5. Select the **Step** and **Increment** of the results file from which the displacement data are to be read to define the imperfection.
6. In the **Data** table, enter the following:

Mode Number

Enter the mode number.

Scaling Factor

Enter the scaling factor. Repeat this row in the table as often as necessary to specify the imperfection as a linear combination of mode shapes.

7. Click **OK** to save your data and to close the dialog box.

• *Introducing a Geometric Imperfection into a Model*

Defining input imperfections

You can define input imperfection on an assembly.

1. From the main menu bar in the Interaction module, select **Special->Imperfection->Create**.
2. In the **Create Imperfection** dialog box that appears, name the imperfection, select **Input imperfection**, and click **Continue**.
3. Select **System**.
4. Select **Input file**, which is an `.odb` file that is used in the imperfection definition.
5. Click **OK** to save your data and to close the dialog box.

• *Introducing a Geometric Imperfection into a Model*

Defining data imperfections

You can define data imperfection on an assembly.

1. From the main menu bar in the Interaction module, select **Special->Imperfection->Create**.
2. In the **Create Imperfection** dialog box that appears, name the imperfection, select **Data imperfection**, and click **Continue**.
3. Select **System**.
4. In the **Imperfection Data** table, enter the following:

Node Number

Enter the node for which the imperfection is to be specified.

Components

Enter the `comp1`, `comp2`, and `comp3` component values for the imperfection in the selected system.

5. Click **OK** to save your data and to close the dialog box.

• *Introducing a Geometric Imperfection into a Model*

Editing the region to which a file imperfection applied

You can edit the region to which a file imperfection is applied.

1. From the main menu bar in the Interaction module, select **Special->Imperfection->Edit->*file imperfection name*** to display the **Edit Imperfection** dialog box.
2. In the top part of the dialog box, click .
3. Edit the region using one of the following methods:
   - Select and unselect objects in the viewport. When you have finished editing the region, click mouse button 2. For more information, see *Selecting objects within the viewport*.
   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
4. In the dialog box, finish editing the imperfection definition as desired and click **OK**.

Inertia

This section provides information on how to model different types of inertia for regions on a part or on an assembly. You can define inertia in the Property module or in the Interaction module.

In this section:

- *Defining inertia*
- *Managing inertia*
- *Defining point mass and rotary inertia*
- *Defining nonstructural mass*
- *Defining heat capacitance*
- *Editing the region to which inertia is applied*

Defining inertia

Inertia can be defined for a part in the Property module or for an assembly in the Interaction module

You can specify the following types of inertia:

Point mass/inertia

You can define lumped mass and rotary inertia at a point on a part or on an assembly. You can also define mass and inertia proportional damping. In an Abaqus/Standard analysis, you can define composite damping. For more information, see *Point Masses*, and *Rotary Inertia*.

Nonstructural mass

You can define nonstructural mass for a region on a part or on an assembly. For more information, see *Nonstructural Mass Definition*.

Heat capacitance

You can define concentrated heat capacitance at a point on a part or on an assembly. For more information, see *Point Capacitance*.

Select **Special->Inertia->Create**: ***type*** from the main menu bar in the Property module or the Interaction module to define inertia. Select **Edit** from the same menu to make changes to an existing definition.

Inertia definitions appear in the Model Tree in the **Engineering Features** container under the part (if created in the Property module) or under the assembly (if created in the Interaction module).

When you define inertia, you can select objects from the viewport to identify the region on which to apply the inertia. By default, a set or surface is created that contains the selected objects. You can change this behavior by toggling off the option to create a set or surface in the prompt area. A default name is provided in the prompt area, but you can enter a new name.

When you apply inertia to regions of the model, you can choose to display symbols in the viewport that indicate where you have applied the inertia. If you apply inertia to geometry, symbols appear at the vertices. If you apply inertia to an orphan mesh, symbols appear at the nodes. For information about graphical symbol types, see *Symbols used to represent special engineering features*. For information about controlling the visibility of these symbols, see *Controlling the display of attributes*.

For detailed instructions on creating this type of engineering feature, see the following sections:

- *Defining point mass and rotary inertia*
- *Defining nonstructural mass*
- *Defining heat capacitance*

Managing inertia

The **Inertia Manager** allows you to create and manage inertia definitions.

The manager includes a list of the names and the types of inertia that you have defined.

The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the manager allow you to create new inertia definitions or to edit, copy, rename, and delete existing ones.

The icons in the column along the left side of the manager allow you to suppress and resume existing inertia definitions. You can also initiate these procedures using the **Special->Inertia** menu from the main menu bar in the Property module or the Interaction module. After you select a management operation from the main menu bar, the procedure is the same as if you had clicked the corresponding button inside the manager dialog box.

Additional information

- *Suppressing and resuming objects*
- *Defining point mass and rotary inertia*
- *Defining nonstructural mass*
- *Defining heat capacitance*
- *Editing the region to which inertia is applied*

Defining point mass and rotary inertia

You can define lumped mass and rotary inertia at a point on a part or on an assembly.
You can also define proportional damping for point masses and rotary inertia. In an Abaqus/Standard analysis, you can define composite damping. For more information, see *Point Masses* and *Rotary Inertia*.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Inertia->Create**.
2. In the **Create Inertia** dialog box that appears, name the inertia, select **Point mass/inertia**, and click **Continue**.
3. Select the points for which you want to define mass and rotary inertia using one of the following methods:

   - Select the points in the viewport. When you have finished selecting, click mouse button 2.

     If the model contains a combination of orphan mesh elements and geometry, select one of the following from the prompt area:

     - Select **Geometry** to define the point mass and rotary inertia for the geometric portions of a part or assembly or for a reference point.
     - Select **Mesh** to define the point mass and rotary inertia for orphan mesh components.

       You can use the angle method to select a group of nodes from an orphan mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit Inertia** dialog box appears. The region to which you are applying the point mass and rotary inertia is highlighted in the viewport.
4. From the **Mass** portion of the **Magnitude** tabbed page, do the following:

   - For isotropic mass, toggle on **Isotropic** and specify the magnitude of the mass.
   - For anisotropic mass, toggle on **Anisotropic** and specify the mass components M_{11}, M_{22}, and M_{33}.

5. In the **Rotary Inertia** portion of the page, specify the moments of inertia (units ML^2).
   a. Toggle on **Specify off-diagonal terms** if you want to specify products of inertia.
   b. Enter values for the rotary inertia.

I11

Rotary inertia about the local 1-axis, I_{11}.

I22

Rotary inertia about the local 2-axis, I_{22}.

I33

Rotary inertia about the local 3-axis, I_{33}.

c. If applicable, enter values for the products of inertia.

I12

Product of inertia, I_{12}.

I13

Product of inertia, I_{13}.

I23

Product of inertia, I_{23}.

6. If you want to change the coordinate system (**CSYS**) for the rotary inertia, click  and use one of the following methods:

- Select an existing datum coordinate system in the viewport.
- Select an existing datum coordinate system by name.
  1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
  2. Select a name from the list, and click **OK**.
- Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

By default, the global coordinate system is used to define the rotary inertia.

7. Use the **Damping** tabbed page to define proportional damping for point masses or rotary inertia. For an Abaqus/Standard analysis, you can also define composite damping. You can specify values for both proportional damping for point masses and composite damping or for both proportional damping for rotary inertia and composite damping; however, Abaqus uses only the damping that is relevant to the particular dynamic analysis procedure being performed. If different damping values are needed for point masses and rotary inertia, you must create separate inertia definitions.

- In the **Alpha** field, enter the α_R factor to create proportional damping for point masses or rotary inertia in a direct-integration dynamic analysis, in a mode-based analysis that supports nondiagonal damping in Abaqus/Standard, or in an explicit dynamic analysis. This value is ignored in a modal dynamic analysis that does not support nondiagonal damping. The default value is 0.0.
- In the **Composite** field, enter the fraction of critical damping, ξ_α, to be used when calculating composite damping factors for the modes in a modal dynamic analysis. This value is ignored in a direct-integration dynamic analysis and in mode-based analyses that support nondiagonal damping in Abaqus/Standard. The default value is 0.0.

8. Click **OK** to save your data and to close the dialog box.
   Symbols appear in the viewport that represent the point mass and rotary inertia that you just created.

Additional information

- *Controlling the display of attributes*
- *Symbols used to represent special engineering features*

Defining nonstructural mass

You can define nonstructural mass for a region on a part or on an assembly.
For more information, see *Nonstructural Mass Definition*.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Inertia->Create**.
2. In the **Create Inertia** dialog box that appears, name the inertia, select **Nonstructural mass**, and click **Continue**.
3. Select the region for which you want to define nonstructural mass using one of the following methods:

   - Select the region in the viewport. When you have finished selecting, click mouse button 2.

     If the model contains a combination of orphan mesh elements and geometry, select one of the following from the prompt area:

     - Select **Geometry** to define the nonstructural mass for the geometric portions of a part or assembly or for a reference point.
     - Select **Mesh** to define the nonstructural mass for orphan mesh components.

       You can use the angle method to select a group of nodes from an orphan mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit Inertia** dialog box appears. The region to which you are applying the nonstructural mass is highlighted in the viewport.
4. Choose the **Units** that you will use to specify the nonstructural mass magnitude. The available types depend on the type of region that you selected.

   - Select **Total Mass** to specify the magnitude in the form of a total mass value.
   - Select **Mass per Volume** to specify the magnitude in the form of mass per unit volume.
   - Select **Mass per Area** to specify the magnitude in the form of mass per unit area.
   - Select **Mass per Length** to specify the magnitude in the form of mass per unit length.

5. Enter the nonstructural mass magnitude in the **Magnitude** field in the units that you chose in the previous step.
6. If you specify the mass in the form of a total mass value and the region as a set, you can choose the **Distribution** method for distributing the mass over the region.

   - Select **Mass Proportional** to distribute the total mass among the members of the set in proportion to the structural mass. The underlying structural density is scaled uniformly over the region, and the center of mass is not altered.

- Select **Volume Proportional** to distribute the total mass among the members of the set in proportion to the volume in the initial configuration. A uniform value is added to the underlying structural density over the region. If the region has a nonuniform structural density, its center of mass might be altered.

7. Click **OK** to save your data and to close the dialog box.

   Symbols appear in the viewport that represent the nonstructural mass that you just created.

Additional information

- *Controlling the display of attributes*
- *Symbols used to represent special engineering features*

Defining heat capacitance

You can define concentrated heat capacitance at a point on a part or on an assembly.
For more information, see *Point Capacitance*.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Inertia->Create**.
2. In the **Create Inertia** dialog box that appears, name the inertia, select **Heat capacitance**, and click **Continue**.
3. Select the points for which you want to define heat capacitance using one of the following methods:

   - Select the points in the viewport. When you have finished selecting, click mouse button 2.

     If the model contains a combination of orphan mesh elements and geometry, select one of the following from the prompt area:

     - Select **Geometry** to define heat capacitance for the geometric portions of a part or assembly or for a reference point.
     - Select **Mesh** to define heat capacitance for orphan mesh components.

       You can use the angle method to select a group of nodes from an orphan mesh. For more information, see *Using the angle and feature edge method to select multiple objects*.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

     Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit Inertia** dialog box appears. The region to which you are applying the heat capacitance is highlighted in the viewport.
4. **Optional:** Toggle on **Use temperature-dependent data**, and enter the temperature.
5. **Optional:** Specify the **Number of field variables**, and enter the value for the first field variable, the value of the second field variable, etc.
6. In the **Data** table, enter the following:

   ### Capacitance

   Enter the capacitance magnitude, ρcV (density × specific heat × volume).
7. Click **OK** to save your data and to close the dialog box.

   Symbols appear in the viewport that represent the heat capacitance that you just created.

Additional information

- *Controlling the display of attributes*
- *Symbols used to represent special engineering features*

Editing the region to which inertia is applied

You can edit the region to which inertia is applied.

1. From the main menu bar in the Property module or Interaction module, select **Special->Inertia->Edit->*inertia name*** to display the **Edit Inertia** dialog box.
2. In the top part of the dialog box, click .
3. Edit the region using one of the following methods:
   - Select and unselect objects in the viewport. When you have finished editing the region, click mouse button 2. For more information, see *Selecting objects within the viewport*.
   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.



Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. In the dialog box, finish editing the inertia definition as desired and then click **OK**.

   The symbols representing inertia in the viewport change to appear on the newly edited region.

Additional information

- *Symbols used to represent special engineering features*

Load cases

This section provides information on how to use load cases.

In this section:

- *What is a load case?*
- *Managing load cases*
- *Load case editors*
- *Viewing load case output*
- *Defining a load case*

What is a load case?

A load case is a set of loads and boundary conditions used to define a particular loading condition. You can use one or more load cases to study the linear responses of a structure subjected to different loading conditions in the following types of analyses: static perturbation, direct steady-state dynamic, SIM-based steady-state dynamic (mode-based and subspace-based), and substructure generation. A load case analysis is generally much more efficient than the equivalent multiple step analysis since it takes advantage of the principal of linear superposition.

You can define load cases directly in terms of loads and boundary conditions or in terms of combinations of previously defined load cases. You use the Load module to define load cases directly in terms of loads and boundary conditions. You use the Visualization module to define load case combinations of previously defined load cases during postprocessing. Load case output is stored in separate frames in the output database (`.odb`), and you can create results for load case combinations by combining the results from multiple frames.

You use the **Load Case** menu in the Load module to create load cases that include previously defined loads and boundary conditions. You can use a nonzero scale factor to scale the magnitude of individual loads and boundary conditions within a load case. You may include a load or boundary condition only once within each load case. If a step contains load cases, you must include each load and boundary condition in the step in one or more load cases. By default, Abaqus/CAE includes all boundary conditions propagated or modified from the base state in each load case you create but allows you to modify this behavior for individual load cases. You can use the boundary condition manager to deactivate unused propagated boundary conditions in a step containing load cases.

In steps containing load cases Abaqus supports only field output requests. The field output requests created in the Step module apply to all load cases in the step. You can use the Visualization module to view and manipulate load case results (see *Viewing load case output*).

For more information on load cases, see *Multiple Load Case Analysis*. For more information on creating loads and boundary conditions in the Load module, see *Creating and modifying prescribed conditions*.

Additional information

• *Suppressing and resuming objects*

Managing load cases

The **Load Case Manager** allows you to organize and manipulate load cases associated with a given model. You access the manager in the Load module by selecting **Load Case->Manager** from the main menu bar. The **Load Case Manager** is shown in *Figure 1*.

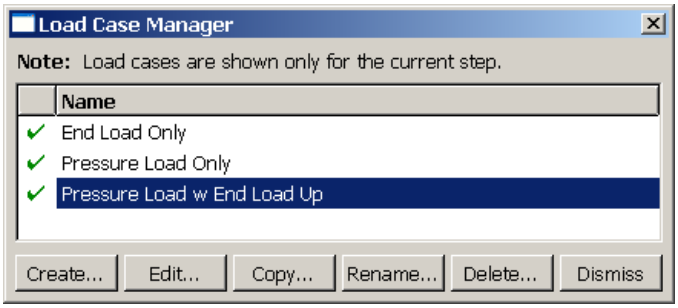


Figure 1: The Load Case Manager.

The icons in the column along the left side of the manager allow you to suppress and resume existing load cases. The **Create**, **Edit**, **Copy**, **Rename**, or **Delete** buttons in the manager allow you to create new load cases or to edit, copy, rename, and delete existing ones. You can also initiate the create, edit, copy, rename, and delete procedures by using the **Load Case** menu in the main menu bar. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

For detailed instructions on creating and editing load cases, see *Defining a load case*.

Additional information

- *Managing objects*
- *Suppressing and resuming objects*

Load case editors

After you enter the Load module and create loads and boundary conditions, you can select **Load Case->Create** from the main menu bar to create a load case. A **Create Load Case** dialog box appears in which you can provide a name for the load case and select the step in which the load case will be created. When you click **Continue** in the **Create Load Case** dialog box, the load case editor appears.

The top panel of the load case editor displays the name of the load case and the current analysis step. The load case editor contains two tabbed pages with tables that allow you to select the loads and boundary conditions to define the load case. By default, Abaqus/CAE includes all boundary conditions propagated or modified from the base state in each load case that you create. You can toggle off this behavior on the **Boundary Condition** tabbed page.

Click  at the bottom of each tabbed page to select loads and boundary conditions from a list. You can enter a scale factor to scale the magnitude of individual loads and boundary conditions. If you enter a negative scale factor for a load, the load will be applied in the opposite direction. You can also enter the names of the loads and boundary conditions and the scale factors directly in the tables. During load case creation, Abaqus/CAE sorts the names of the loads and boundary conditions alphabetically and lists the names in the load case editor. You can highlight selected loads and boundary conditions in the viewport.

For example, you can define a set of load cases to analyze the response of a cantilever beam subjected to an end load and a uniform pressure load. The cantilever beam model is shown in *Figure 1*.

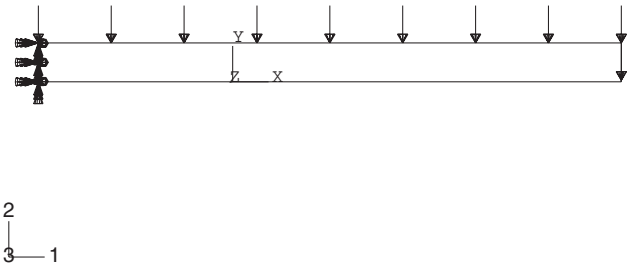


Figure 1: Cantilever beam model with end load, pressure load, and encastre boundary condition.

An encastre boundary condition is applied in the initial step, and the end load and pressure load are applied in a static, linear perturbation step.

You can create a load case that contains only the end load and another that contains only the pressure load. In addition, you can create a load case that contains both the pressure load and the end load with the end load applied in the opposite direction (scale factor value of −1), as shown in *Figure 2*.

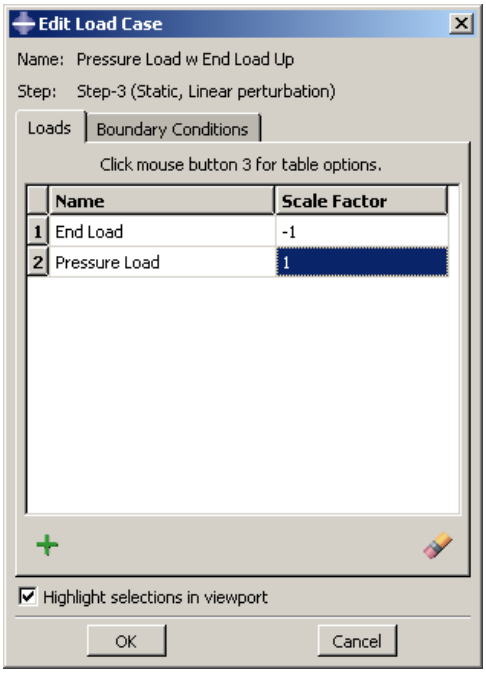


Figure 2: Load case definition that includes loads for the cantilever beam model.

By default, each load case contains the encastre boundary condition propagated from the base state, as shown in *Figure 3*.

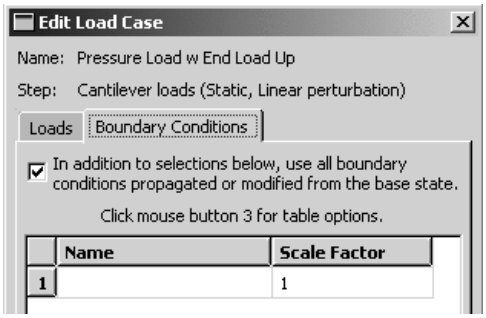


Figure 3: Load case definition that includes boundary conditions propagated from the base state for the cantilever beam model.

The **Load Case Manager** in *Figure 1* shows the three load cases created for the cantilever beam model.

Additional information

- *What is a load case?*

Viewing load case output

Load case output is stored in separate frames in the output database (`.odb`). You use the Visualization module to view and manipulate load case output. The load case name is displayed in the state block for each frame containing load case output. For example, *Figure 1* shows the deformed plot for the `Pressure Load w End Load Up` load case of the cantilever beam model (see *Load case editors*).

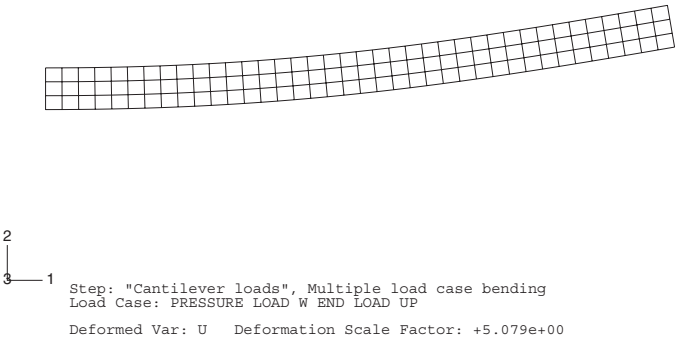


Figure 1: Results from a load case analysis.

In the Visualization module you can linearly combine output from several load cases to represent an actual loading environment. You can also obtain the minimum or maximum value of a selected field variable over some or all load cases.

For example, you can combine the output from the `End Load Only` and `Pressure Load Only` load cases for the cantilever beam model to create new field output with a load case name of `Both loads`, as shown in *Figure 2*.

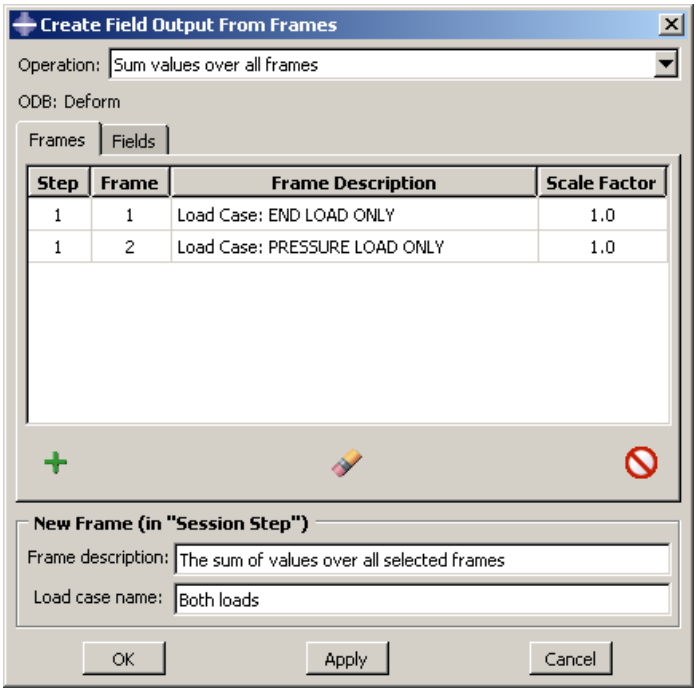


Figure 2: Creating new field output by combining the output from two load cases (frames).

The new field output is contained in a frame of the session step and is available from the **Field Output** dialog box. *Figure 3* shows a plot of the combined deformation results.

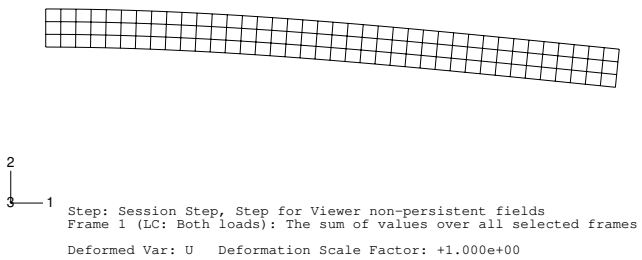


Figure 3: Combined deformation results from both load cases.

For detailed information on manipulating load case output, see *Combining results from several frames*.

Additional information

- *Creating and saving new field output*

Defining a load case

You use the **Load Case** menu in the Load module to define a load case. You can create load cases in static perturbation, direct steady-state dynamic, SIM-based steady-state dynamic (mode-based and subspace-based), and substructure generation analyses. You must create the individual loads and boundary conditions in the Load module prior to creating a load case. For more information, see *Creating loads*, and *Creating boundary conditions*.

1. From the main menu bar in the Load module, select **Load Case->Create**.

   Tip: You can also create a load case using the tool in the Load module toolbox.

2. In the **Create Load Case** dialog box that appears, do the following:
   a. Name the load case. For more information about naming objects, see *Using basic dialog box components*.
   b. Select the step. You can define a load case only during the following steps: static, linear perturbation; steady-state dynamic, direct; SIM-based steady-state dynamic (modal and subspace); and substructure generation.
   c. Click **Continue**.

   The load case editor appears.

3. If you want to include loads in the load case, click the **Loads** tab. Specify the loads using one of the following methods:

   - Click at the bottom of the page to display a list of the loads available for the selected step.
     1. From the **Load Selection** dialog box, select the loads from the list (see *Selecting multiple items from lists and tables*, for more information). The selected loads are highlighted in the viewport.
     2. In the **Scale factor** field, enter a nonzero value to scale the magnitude of the selected loads. Enter a negative value to apply the selected loads in the opposite direction. The default value of the scale factor for each load is 1.0.
     3. Click **OK** to save your selections and to close the **Load Selection** dialog box.
   - Enter data in the table using the keyboard.
     1. Enter the name of a previously defined load.
     2. Enter a nonzero value to scale the magnitude of the load. Enter a negative value to apply the load in the opposite direction. The default value of the scale factor is 1.0.
     3. Repeat the previous steps to specify all the loads to include in the load case.

   The load case editor displays the loads and scale factors included in the load case. To highlight loads in the viewport, click the table row headings to select the loads in the editor (see *Selecting multiple items from lists and tables*, for more information) and toggle on **Highlight selections in viewport**. Suppressed loads are not highlighted in the viewport. To remove loads from the table, click the table row headings to select the loads and click .

4. If you want to include boundary conditions in the load case, click the **Boundary Conditions** tab to display the boundary conditions page. By default, Abaqus/CAE will include all boundary conditions propagated or modified from the base state in each load case you create. You can toggle off this behavior at the top of the boundary condition page. Specify the boundary conditions using one of the following methods:

- Click  at the bottom of the page to display a list of the boundary conditions available for the selected step.

  1. From the **Boundary Condition Selection** dialog box, select the boundary conditions from the list (see *Selecting multiple items from lists and tables*, for more information). The selected boundary conditions are highlighted in the viewport.
  2. In the **Scale factor** field, enter a nonzero value to scale the magnitude of the selected boundary conditions. The default value of the scale factor for each boundary condition is 1.0.
  3. Click **OK** to save your selections and to close the **Boundary Condition Selection** dialog box.

- Enter data in the table using the keyboard.

  1. Enter the name of a previously defined boundary condition.
  2. Enter a nonzero value to scale the magnitude of the boundary condition. The default value of the scale factor is 1.0.
  3. Repeat the previous steps to specify all the boundary conditions to include in the load case.

The load case editor displays the boundary conditions and scale factors included in the load case. The scale factor is relevant only if the boundary condition has a nonzero value. To highlight boundary conditions in the viewport, click the table row headings to select the boundary conditions in the editor (see *Selecting multiple items from lists and tables*, for more information) and toggle on **Highlight selections in viewport**. Suppressed boundary conditions are not highlighted in the viewport. To remove boundary conditions from the table, click the table row headings to select the boundary conditions and click .

5. Click **OK** to save your load case definition and to close the editor.

Additional information

- *Suppressing and resuming objects*
- *Load case editors*
- *Multiple Load Case Analysis*

Midsurface modeling

This section explains midsurface modeling and how you can use it to simplify thin solid models for analysis.

In this section:

- *Understanding midsurface modeling*
- *Understanding the reference representation*
- *Creating shell faces for the midsurface model*
- *Examples of creating a midsurface model*
- *Creating and editing midsurface models*

Understanding midsurface modeling

Midsurface modeling is a technique for creating a simplified shell representation of a solid model. Midsurface modeling can reduce the computational expense of analyzing a complete solid model when a shell model with a defined thickness and fewer details is suitable for the required analysis.

The midsurface modeling process relies on an accurate solid model as the starting point. Midsurface modeling is best suited to thin solids or thin-walled solids where wall thickness is constant or where reasonable approximations of the wall thickness at each point can be made easily. You can apply midsurface modeling to any solid cell within a model; you need not apply it to the entire model. If you apply midsurface modeling to only a portion of a solid model, Abaqus/CAE automatically creates shell-to-solid coupling constraints to couple the motion of the midsurface shell edges to that of the remaining solid model faces. Shell-to-solid coupling constraints will not be created if the angle between the shell surface and the solid face deviates significantly from 90°.

Besides reduced expense, you might use a midsurface model in place of a solid model to better account for bending response in thin sections of the model. Shell elements are designed to manage bending loads within the thickness of a single element, whereas a single solid element will have little or no resistance to bending.

Creating a midsurface model in Abaqus/CAE is a manual process. The procedure presented below provides an overview of the general steps involved. The tasks within each step may vary depending on the complexity and completeness of the original solid model and the analysis requirements. Some steps, such as Step 4 below, may not be required or may be completed in a different order from what is presented here.

1. Assign midsurface regions on the solid part. Use the **Assign Midsurface Region** tool to select one or more cells from the solid part to begin creating a midsurface model. For more information, see *Assigning a midsurface region*.

   Abaqus/CAE removes the selected cells from the active representation and creates a reference representation containing the selected cells.

2. Create new shell faces to replace the solid geometry that you moved to the reference representation.

   You can use tools in the Geometry Edit toolset or the shell feature creation tools to add shell features to the midsurface model. For more information, see *The Geometry Edit toolset* and *Adding a shell feature*, respectively. The offset faces tool in the Geometry Edit toolset is the most common face creation method for midsurface models. The offset process creates new faces based on selected geometry and automatically assigns a shell thickness based on the offset parameters.

3. Assign thicknesses to the new shell features.

   The **Assign Thickness and Offset** tool allows you to assign shell thicknesses to the new midsurface geometry (for more information, see *Assigning thicknesses and offsets*). The tool allows you to assign thicknesses to individual faces, and you can also use it to edit thicknesses that Abaqus/CAE assigned automatically for faces created using the offset process. You must still assign a section to the new faces in the Property module. During the section assignment process you can make a final decision whether to use the thickness defined with the section properties or to use the definition from the geometry. You can toggle the shell thickness display in all part and assembly-based modules to see the thickness in the viewport (for more information, see *Visualizing shell thicknesses*).

4. Revise any analysis attributes that were associated with the solid geometry that was used to create the reference representation.

   Analysis attributes such as loads, boundary conditions, or interactions cannot be applied to geometry in the reference representation. Any attributes that were associated with the solid geometry before it was moved to the reference representation will be associated with an empty region if the geometry is no longer part of the active representation. For more information, see *Understanding the reference representation*.

Understanding the reference representation

A reference representation is an alternative representation of a part or a subset (one or more cells) of a part that is not used in the analysis. The reference representation retains the original solid model geometry, and, like reference geometry in the Sketcher, you can use entities from the reference representation as tools to construct a simpler version of the part. The reference representation is most commonly used to construct a midsurface model. When used with the offset face tool to create faces for the midsurface model, the offset calculations between the reference representation and the new faces also allow Abaqus/CAE to automatically compute shell thicknesses and element offsets in the midsurface model.

Abaqus/CAE creates a reference representation when you assign a midsurface property to one or more cells in a solid part. The selected cells are removed from the active representation of the part and added to the reference representation. The reference representation appears by default in the Part module and is colored translucent brown. You cannot change the color of the reference representation, but you can toggle it on or off in all modules using the **Show Reference Representation** tool  located with the visible object tools in the main toolbar. You can also toggle translucency of the reference representation on or off using the display options (for more information, see *Controlling edge visibility*.

You can use the reference representation with the shape creation tools or with the tools in the Geometry Edit toolset to help construct new geometric entities in the active representation. For example, you can offset faces in the reference representation or you can select a planar face from the reference representation as the sketch plane to create a new shell face using the extrude method. You can use the geometry in the reference representation to define partitions—select an edge and a vertex from the reference representation as the point and normal to partition a cell—or create datum entities. You can also use reference representation geometry to position entities and define constraints in the Assembly module.

You cannot edit the geometry within the reference representation, and you cannot assign analysis attributes such as loads, boundary conditions, and interactions to it. When you create a reference representation, any analysis attributes that were applied to the solid cells before you created the reference representation will now be associated with an empty region. You can edit the empty regions to associate the analysis attributes with geometry—solid geometry or shell geometry that you create for the midsurface model—that is part of the active representation. Alternatively, you can suppress or delete the analysis attributes. If you do not make any changes to the affected analysis attributes, Abaqus/CAE will display a warning message when you submit the job in the Job module.

Creating shell faces for the midsurface model

Once you have a reference representation, you must manually build a shell model to replace the solid cells that have been removed from the model.
You can use the shell feature tools and the tools in the Geometry Edit toolset (for more information, see *Adding a shell feature* and *The Geometry Edit toolset*, respectively) to create and edit shell faces.

This section discusses some special conditions that may apply when you use these tools to create faces for a midsurface model.

In this section:

- *Using the offset face tool for midsurface modeling*
- *Using the extend faces tool for midsurface modeling*

Using the offset face tool for midsurface modeling

You can use the offset face tool from the Geometry Edit toolset to create midsurface faces. The offset face tool creates new faces at an offset distance from an original set of faces. The offset faces are created using a similar process to that used for offsetting objects in the Sketcher (for more information, see *Offsetting objects*). Use of the offset face tool is discussed in detail in *Offset faces*.

In addition to the basic task of creating new shell faces, the offset face tool offers two functions that are helpful for creating a midsurface model:

- The offset distance is used by Abaqus/CAE to assign a thickness to the new shell faces.
- The auto trim option automatically trims the new faces to the reference representation.

Once you have selected the faces to offset, you can enter a value for the offset distance or select a set of target faces that represents the distance through the desired shell thickness and have Abaqus/CAE calculate an offset. Abaqus/CAE calculates the distance between faces being offset and the target faces. You can offset to half the average distance or a fraction of the distance to the closest or farthest point on the target faces.

The difference in using each of the calculated offset distances is shown in *Figure 1*. The part has three different thicknesses. The thinnest face is at the top, the thickest in the middle, and the vertical face on the right has a thickness between the other two. The three outer faces of the reference representation were selected to be offset, and the three inner faces are the target faces. The view on the left shows the default **Half the average distance** method, and the offset is approximately the full thickness of the top face. The middle view shows the **Fraction distance to closest point on face** method with the fractional distance set to `0.5`. The top target face is closest to the top offset face, so Abaqus/CAE offsets the new faces by `0.5` times this distance. The view on the right shows the **Fraction distance to farthest point on face** method with the fractional distance again set to `0.5`. The middle (angled) target face is farthest from the middle offset face, so Abaqus/CAE offsets the new faces by `0.5` times this distance—the top offset face is cut off where it would extend through the reference representation of the original solid part because the **Auto trim to reference representation** option was used.

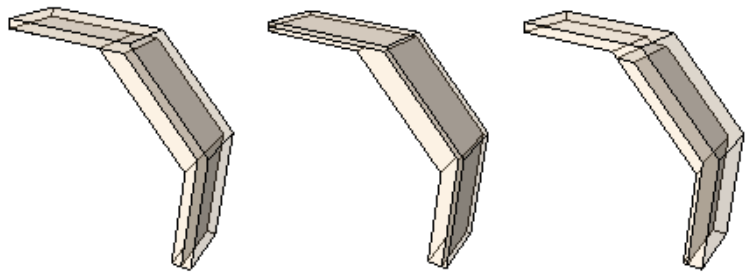


Figure 1: Differences in the offset distance based on the three target face calculation methods.

Abaqus/CAE automatically associates the new offset shell faces with a thickness based on the offset calculations. If you selected target faces to calculate the offset distance, Abaqus/CAE also automatically recalculates the offset for any changes in the geometry of the top and bottom face sets. Use the **Assign Thickness and Offset** tool and the **Render shell thickness** option to edit and view the thicknesses assigned to the shell faces. For more information, see *Assigning thicknesses and offsets*, and *Visualizing shell thicknesses*.

If you use the auto trim option while offsetting faces, Abaqus/CAE offsets the faces and extends the outer edges of the new faces by an amount equal to the offset distance. The new faces are then trimmed where they intersect with the reference representation. *Figure 2* shows a possible effect of using the auto trim option. The reference representation contains a cell, and the larger circular face of the cell was picked for the offset operation. Using the auto trim option, the new face is trimmed to the boundary of the conical face of the reference representation.

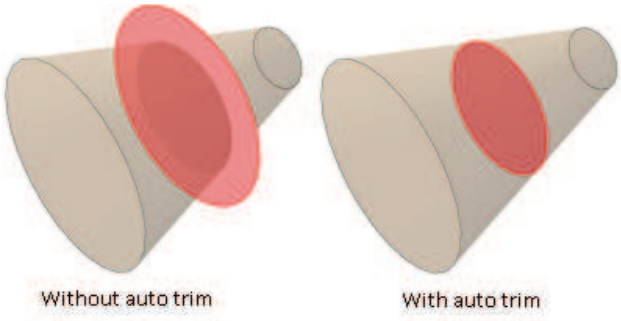


Figure 2: The auto trim option trims to the outer faces of the reference representation.

Additional information

- *Offset faces*
- *Extend faces*

Using the extend faces tool for midsurface modeling

You can use the extend faces tool from the Geometry Edit toolset to create midsurface faces. The extend faces tool extends existing shell faces, so for midsurface modeling it must be used after you have created one or more shell faces. Use of the extend faces tool is discussed in detail in *Extend faces*.

You can extend faces by selecting individual edges to extend the faces in one or more directions, or you can specify faces to extend the faces along all exterior edges. There are also several methods for determining the extension distance; the **Up to reference representation** option is intended specifically for midsurface modeling.

When you use the **Specify edges of faces to extend** option to extend faces along a set of edges, the extend faces operation may fail. If this occurs, try reducing the number of selected edges and then using a second operation to extend the faces along the remaining edges.

Additional information

- *Extend faces*

Examples of creating a midsurface model

This section provides examples of creating midsurface models from solid models. In the first example the solid part has reasonably uniform thickness, and creation of the midsurface model is relatively easy. The second example is more complex and requires multiple steps to create a midsurface model that accurately represents the original solid model.

In this section:

- *Midsurface modeling of a stamped bracket*
- *Midsurface modeling of a complex part*
- *Creating the shell representation of the beam*

Midsurface modeling of a stamped bracket

This example shows how you can use the midsurface modeling tools in Abaqus/CAE to convert a solid model of a bracket to a shell representation of the same part.

The solid model

The model in this example is a 175.0 mm long bracket that is stamped from a 3.0 mm thick plate of mild steel, as shown in *Figure 1*. A default mesh of the solid part results in one or two elements through the thickness of the bracket that will not perform well in bending. Modeling a shell representation of the bracket will provide a more accurate bending response.

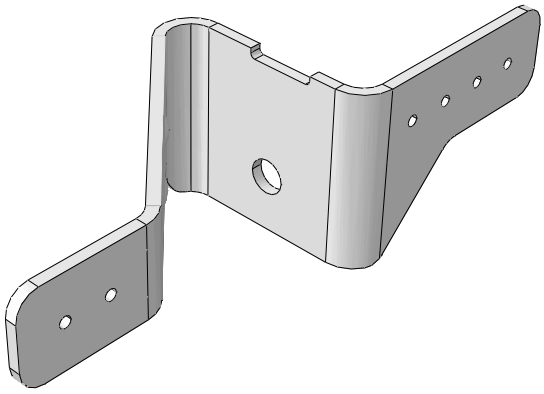


Figure 1: The solid model of the bracket.

Assign the midsurface region

Use the **Assign Midsurface Region** tool in the Part module to remove geometry from the active representation of the model and to create a reference representation of the original solid geometry, as shown in *Figure 2*.

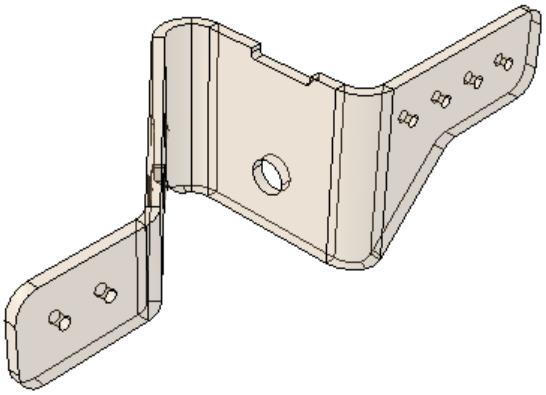


Figure 2: The reference representation of the bracket.

The reference representation is an abstract representation of the original bracket. It retains the original geometry of the bracket, but it cannot be used in the analysis. The reference representation appears by default in the Part module; you can toggle the view off and on using the **Show Reference Representation** tool located with the visible object tools in the main toolbar. For more information, see *Understanding the reference representation*, and *Assigning a midsurface region*.

Create the shell representation

You must create a shell representation of the bracket that can be analyzed by Abaqus. The most commonly used tool for creating a midsurface shell model is the offset face tool, located with the other face tools in the Geometry Edit toolset. The offset face tool allows you to select one or more faces from the reference representation and to create new faces that are offset from the original faces. You can enter a fixed distance, or you can select target faces that Abaqus/CAE uses to calculate the distance. If you select target faces, the resulting shell thickness varies if the distance between the face to offset and the target face is not constant. Abaqus/CAE calculates the nodal thickness and the element offset for each node and element when you analyze the model. The **Offset Faces** dialog box is shown in *Figure 3*.

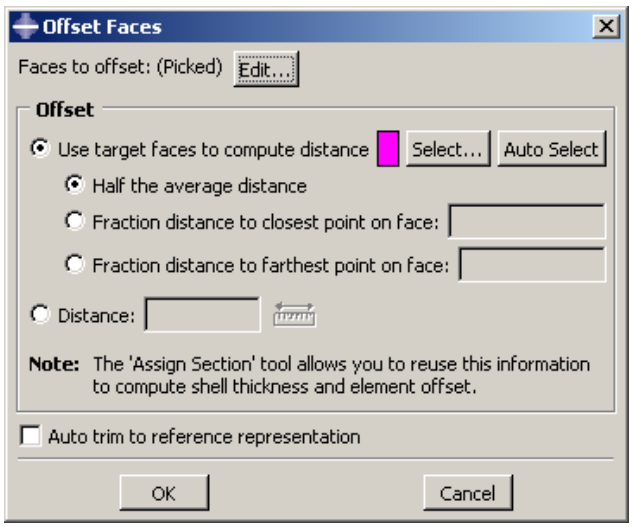


Figure 3: The Offset Faces dialog box.

For more information, see *Methods for editing faces*, and *Adding a shell feature*.

Figure 4 shows the selected faces to offset and the selected target faces. The **by angle** selection method was used to select the inner group of face as the faces to offset and the outer group of faces as the target faces.

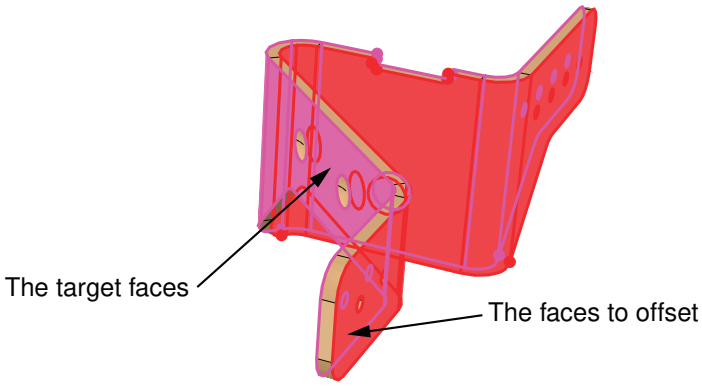


Figure 4: The faces to offset and the target faces.

In this example you select the default option of **Half the average distance** from the **Offset Faces** dialog box, and Abaqus/CAE creates faces that are half the average distance between the faces to offset and the target faces.

By default, **Auto trim to reference representation** is toggled off in the **Offset Faces** dialog box. However, the auto trim option was active for this step, so Abaqus/CAE trimmed the offset faces to align with the reference representation. In some cases the trimming operation fails, which results in a slight misalignment between the new faces and the reference representation and a warning message from Abaqus/CAE. Even when trimming fails, the resulting shell face is often still a reasonable approximation of the reference representation; so you may be able to ignore the warning message. For more information, see *Using the offset face tool for midsurface modeling*.

Assign a shell section

You use the Property module to create a shell section and to assign it to the new face. When you create the shell section, you can enter an arbitrary value for the shell thickness. When you subsequently assign the section to the shell, you specify that the thickness and the shell offset are calculated from the geometry in the **Edit Section Assignment** dialog box, as shown in *Figure 5*. Abaqus/CAE ignores the thickness value that you entered for the shell section.

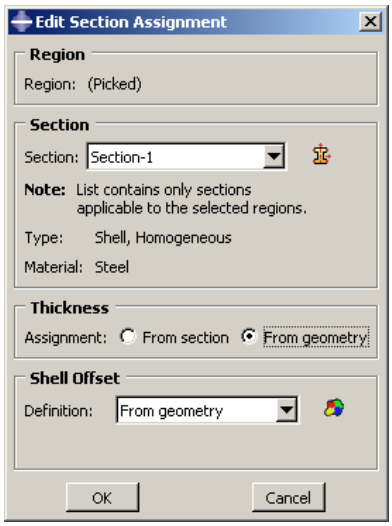


Figure 5: The Edit Section Assignment dialog box.

For more information, see *Assigning a section*.

If desired, you can toggle on **Render shell thickness** from the **Part Display Options** to view the thickness of the shell, as shown in *Figure 6*.

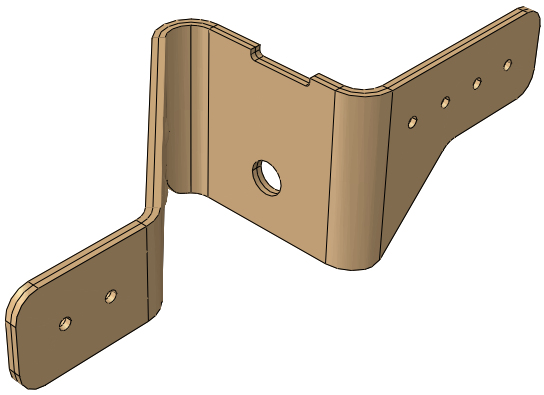


Figure 6: Viewing the shell thickness.

Mesh the part

Abaqus/CAE colors the shell part pink in the Mesh module to indicate it can be meshed using the free meshing technique, as shown in *Figure 7*.

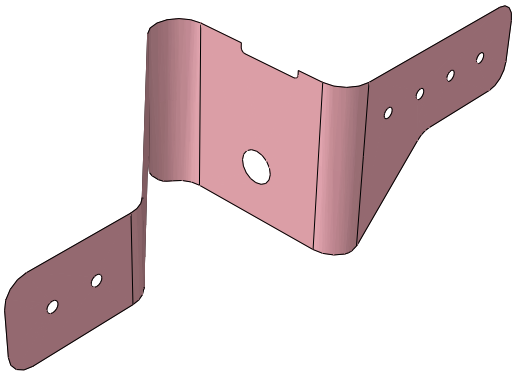


Figure 7: Free meshing can be applied to the part.

You apply automatic virtual topology to the part to remove small details from the original part and to remove details that were created during the face trimming operation, as described in *Creating virtual topology automatically*.

You apply default seeding and mesh controls, and the resulting mesh is shown in *Figure 8*.

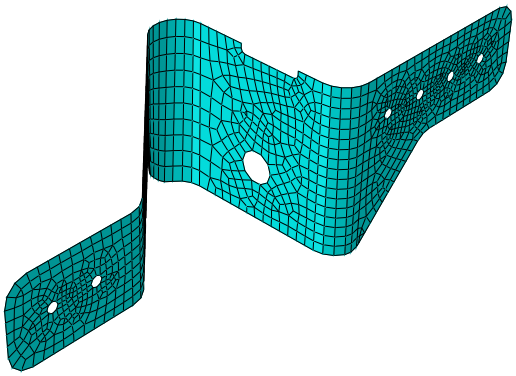


Figure 8: The resulting mesh.

Midsurface modeling of a complex part

This example uses several techniques and tools to create the midsurface model for a reinforced structural component.

The solid model

The model in this example is the structural beam shown in *Figure 1*. The reinforcing ribs, different thicknesses, and asymmetrical shape of the beam do not allow for a simple beam section representation. The complexity of the part combined with its thin cross-sections make it a good candidate for replacement with a midsurface model. As in the previous example, bending performance will be improved by using a shell model for the mesh instead of thin solid sections.

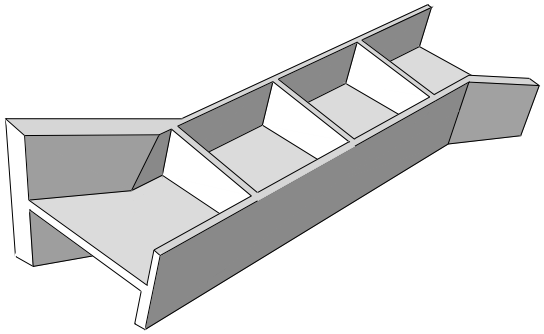


Figure 1: The solid model of the reinforced beam.

Assign the midsurface region

Use the **Assign Midsurface Region** tool in the Part module to remove geometry from the active representation of the beam and to create a reference representation of the original solid geometry, as shown in *Figure 2*. The reference representation is an abstract representation of the original part. It retains the original geometry of the part, but it cannot be used in the analysis. The reference representation appears by default in the Part module; you can toggle it off and on using the **Show Reference Representation** tool  located with the visible object tools in the main toolbar. For more information, see *Understanding the reference representation*, and *Assigning a midsurface region*.

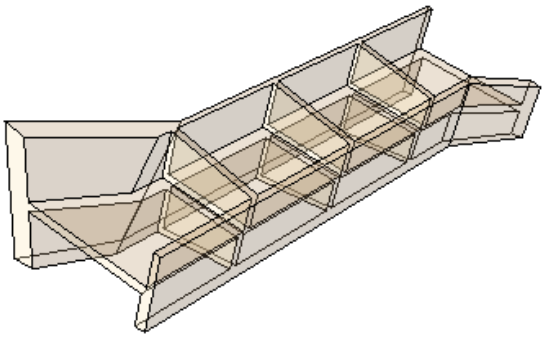


Figure 2: The reference representation of the beam.

Create the shell representation

You must create a shell representation of the beam that can be analyzed by Abaqus. Creating a shell for this part requires multiple steps and tools. There may be several equally valid ways to produce an accurate shell representation for a model. See *Creating the shell representation of the beam*, to use tools from the Geometry Edit toolset to create the new shell faces.

Assign thicknesses

All the original solid geometry has now been replaced with shell geometry. To complete the model, you should verify that the shells have appropriate thickness information. Click the assign thickness and offset tool . Abaqus/CAE highlights any shell faces that do not have thickness data. In this case, since the shell faces were all created using the offset, extend, and blend tools, all of the faces already have thickness data assigned. If there were faces without thickness data, you would select each face and, using the **Compute thickness from opposite faces** method in the **Assign thickness and Offset** dialog box, pick appropriate top and bottom faces from the reference representation to create the missing thicknesses.

To view the model with shell thicknesses, you can toggle on **Render shell thickness** in the **Part Display Options** dialog box (for more information, see *Visualizing shell thicknesses*). As shown in *Figure 3*, the resulting view includes the variations in thickness that were in the original solid model.

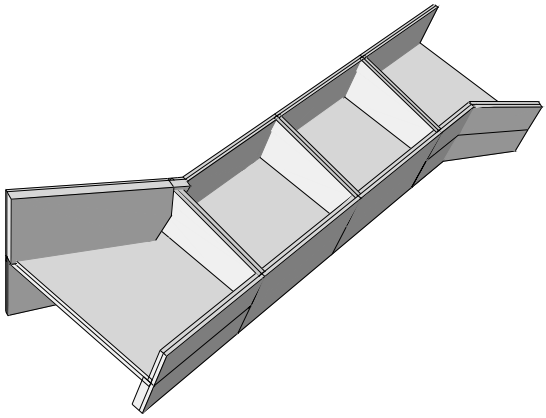


Figure 3: Shell faces with thickness displayed.

Assign a shell section

Use the Property module to create a shell section and assign it to the midsurface model. When you create the shell section, you can enter an arbitrary value for the shell thickness. When you subsequently assign the section to the shell, you specify that the thickness and the shell offset are calculated from the geometry in the **Edit Section Assignment** dialog box. Abaqus/CAE ignores the thickness value that you entered for the shell section and uses the thicknesses assigned to the faces in the Part module. For more information, see *Assigning a section*. *Figure 4* shows the completed midsurface model with section thicknesses after section assignment in the Property module. The geometry is identical to that in *Figure 3*.

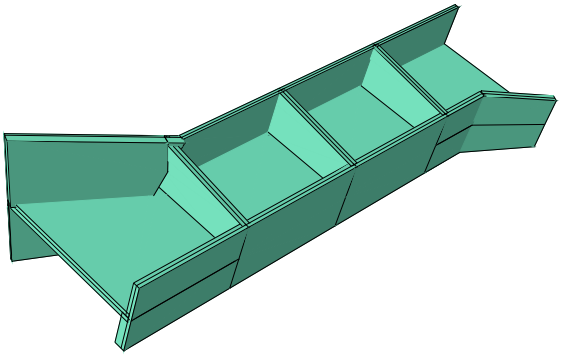


Figure 4: The completed midsurface geometry with shell thicknesses.

Mesh the part

Abaqus/CAE colors the shell part pink in the Mesh module to indicate it can be meshed using the free meshing technique, as shown in *Figure 5*.

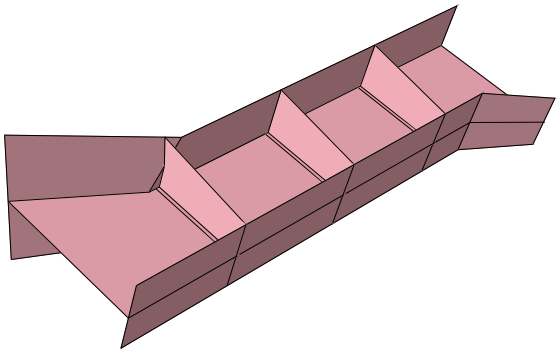


Figure 5: Free meshing can be applied to the part.

Before seeding and meshing the part, you can apply automatic virtual topology to remove small details that are not needed in the mesh (for more information, see *Creating virtual topology automatically*). The default automatic virtual topology settings should remove the blended face edges and other small details that would unnecessarily constrain the part mesh.

Note:

Automatic virtual topology may fail if neighboring faces have inconsistent normals. If this occurs, return to the Part module and use the tool in the Geometry Edit toolset to repair the face normals.

Apply default seeding and mesh controls, and generate the mesh on the part. The resulting mesh is shown in *Figure 6* with the shell thickness displayed.

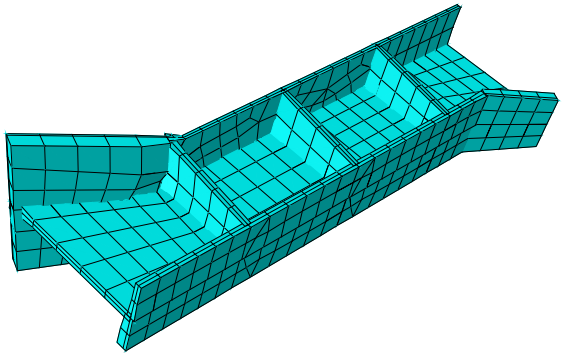


Figure 6: The resulting mesh with shell thickness rendering.

Creating the shell representation of the beam

You must create a shell representation of the beam that can be analyzed by Abaqus. Creating a shell for this part requires multiple steps and tools. There may be several equally valid ways to produce an accurate shell representation for a model. The following steps use tools from the Geometry Edit toolset to create the new shell faces.

1. Use the offset face  tool to create shell faces for the vertical upper left side of the beam.

   Select faces from the reference representation to create the offset shell. Use **Auto Select** to select the target faces. The faces to offset and the target faces are shown in red and magenta, respectively, in *Figure 1*.

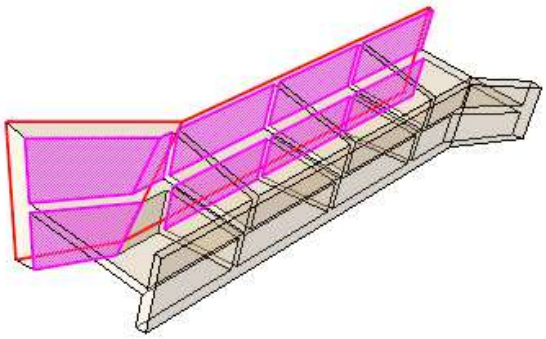
   

   Figure 1: Selected faces to create the offset for the upper left side.

   The faces to offset are the two outer vertical faces toward the rear. The target faces are the nine faces, mostly unconnected, that comprise the inner surface of the same beam wall. The **Fraction distance to closest point on face** method is used with an entry of `0.5` to create the offset with half the thickness of the thinner main portion along the top of the beam. Using this option prevents the thicker sections at the left and along the bottom of the beam from creating an offset larger than the thinner portion. The **Auto trim to reference representation** option is used for this step. This operation extends the faces during the offset process and then trims them along the edges of the reference representation. Auto trim may fail when multiple offset and target faces are selected; if auto trim fails, Abaqus/CAE displays a warning indicating the failure. You can use the extend faces tool to extend the new faces to meet the edges of the reference representation. The resulting offset faces are shown in *Figure 2*.

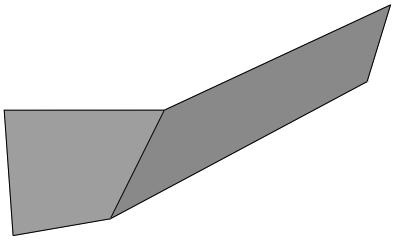
   

   Figure 2: The resulting offset faces.

2. Repeat the process from Step 1 to create offset faces for the opposite side of the beam.

   The resulting shell model now contains the two outer surfaces of the beam, as shown in *Figure 3*.

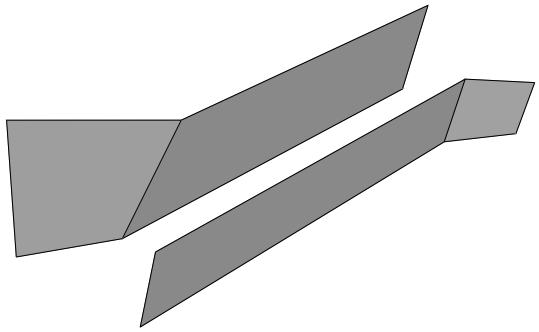


Figure 3: The offset faces from the first and second steps.

3. Create offset faces for the four horizontal sections that connect the two side walls.

   You can create the faces in a single offset operation by selecting the four top faces to offset and using **Auto Select** to select the corresponding bottom faces. In this case **Auto trim to reference representation** is toggled off, so the new faces are not extended and trimmed. The shell model of the horizontal faces is shown in *Figure 4*. (The faces created in Step 2 have been suppressed for clarity.) Notice the gaps between the horizontal faces where they meet the vertical reinforcement ribs in the original part. Similar gaps exist between the horizontal shell faces and the side walls created in the previous two steps.

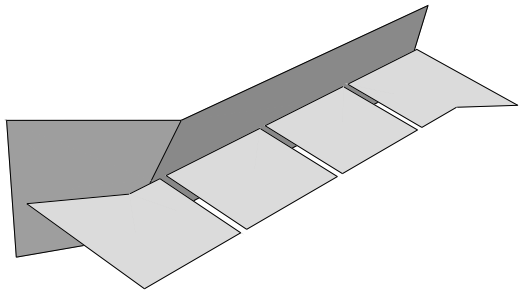


Figure 4: The horizontal offset faces.

4. Repeat the process from Step 3 to create offset faces for the vertical ribs that connect the sides of the beam.

   The midsurface shell model now contains faces representing nearly all of the solid geometry, as shown in *Figure 5*. However, there are gaps between most of the faces due to the thickness of the original solid structure.

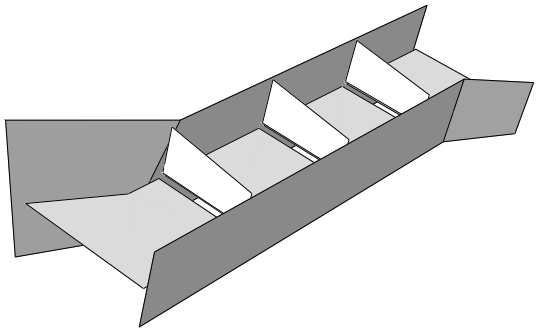


Figure 5: The midsurface model with gaps between the faces.

5. Close the gaps between the vertical ribs and the beam sides.

   Use the extend faces tool to extend the six vertical ribs to intersect with the sides. The selected faces and target faces are shown in magenta and yellow, respectively, in *Figure 6*.

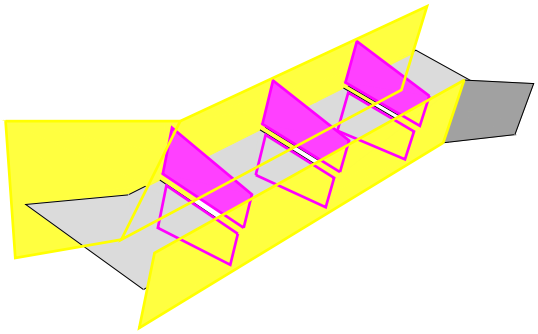


Figure 6: Specifying the faces to extend and the target faces.

When you click **OK**, Abaqus/CAE updates the highlighting to indicate the edges along which the faces will be extended, as shown in *Figure 7*, and displays a warning dialog box with options allowing you to accept the selection, to extend all edges of the selected faces, or to cancel the extend faces procedure.

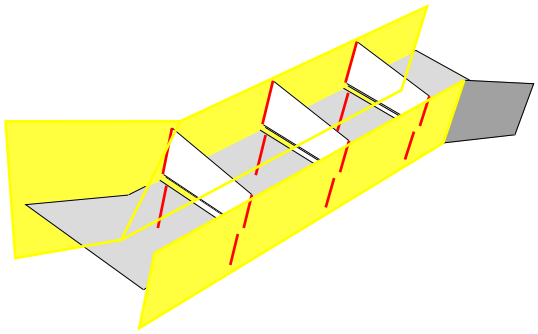


Figure 7: The edges where the vertical faces will be extended.

6. Use the extend faces tool to extend the horizontal faces to intersect with the sides of the beam.

In this case, use the **Specify edges of faces to extend** method with **Up to target faces** to pick the edges and faces, respectively, shown in *Figure 8*. When Abaqus/CAE highlights the edges indicating the areas that will be extended, the four edges indicated by arrows in the figure are removed from the selection because they do not completely match the target faces. Click **No** to use the previous selection of edges.

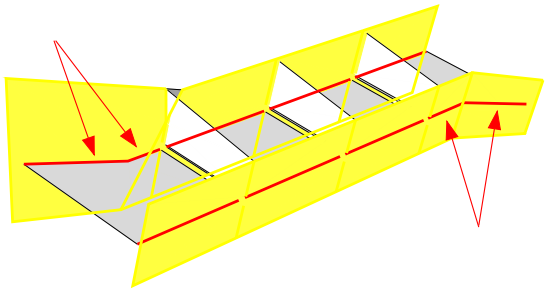


Figure 8: The edges along which the horizontal faces will be the extended.

7. Close the remaining gaps between the horizontal and vertical connecting sections of the beam. You can use either the blend faces tool or the extend faces tool to complete the midsurface model by closing the remaining gaps between the horizontal and vertical connecting sections of the beam.

   a. Use the extend faces tool with the selections shown in *Figure 9*.

   Toggle on **Trim to extended underlying target surfaces** so that Abaqus/CAE will extend and trim the vertical faces to their implied intersections with the selected target face.

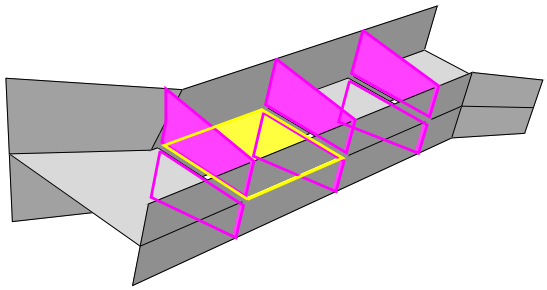


Figure 9: Filling the gaps between the vertical reinforcing ribs.

   b. Use the blend faces tool to fill the remaining gaps.

   The tool must be used three times to fill the three remaining gaps in the model.

Creating and editing midsurface models

The typical process for creating a midsurface model in Abaqus/CAE begins with the solid model. From the solid model, you select cells for which you want to create a midsurface model. Assigning these solid cells as a midsurface region creates a reference representation in the viewport from which you can create shell faces for the midsurface model. You can use tools from the Geometry Edit toolset to create the shell faces, then you can assign thicknesses to the new shells. When you assign thicknesses and offsets to the shell geometry, the default settings typically center the shell representation within the thickness.

In this section:

- *Assigning a midsurface region*
- *Assigning thicknesses and offsets*
- *Assigning thickness to shell faces*
- *Visualizing shell thicknesses*

Assigning a midsurface region

To begin creating a midsurface model, you select the solid model regions that you would like to represent as shells. When you assign a midsurface region to one or more cells of a solid model, Abaqus/CAE creates an **Assign Midsurface** feature and moves the selected cells from the active representation of the part to the reference representation. The reference representation works similarly to reference geometry in a sketch—using the Geometry Edit toolset, you can select points and edges from the reference representation to create points, edges, and faces in the midsurface model. The new midsurface geometry that you create becomes part of the active representation that will be meshed and analyzed.

1. From the main menu bar, select **Tools->Midsurface->Assign**.

   Tip: You can also assign midsurface regions using the  tool, located with the midsurface tools in the Part module toolbox.

2. Select the solid cells from which you want to create the midsurface model.

   You can select individual cells from the viewport or select an existing set of solid cells. For more information on using the selection methods in Abaqus/CAE, see *Selecting objects within the viewport*.

   Abaqus/CAE creates an **Assign midsurface** feature containing the selected cells and displays them as translucent in the viewport.

3. Repeat Step 2 to assign more midsurface regions, or click **Done** to end the procedure.

Additional information

- *Editing techniques*

Assigning thicknesses and offsets

Like all shell faces in Abaqus/CAE, midsurface model faces require a defined thickness. After you have created the shell geometry for the midsurface model, the **Assign Thickness and Offset** tool provides a method to accurately capture the thickness of the solid model that you are replacing. You can also use this tool to edit thicknesses assigned automatically by Abaqus/CAE. For example, you can edit the thickness of faces that were created using the face offset process (for more information, see *Offset faces*). Abaqus/CAE automatically derives the shell thickness for offset faces from the parameters used in the offset operation.

There are two ways to assign a thickness to shell faces in Abaqus/CAE. You can apply a shell thickness definition by using the section properties in the Property module, or you can assign thicknesses by selecting faces and editing their thicknesses with the **Assign Thickness and Offset** tool in the Part module. During the section assignment process you select which method Abaqus/CAE uses to assign thickness to a particular set of geometric faces. (For more information on section assignments, see *Assigning a section*.)



Note:

Thickness data assigned in the Part module—either automatically or using the **Assign Thickness and Offset** tool—appear as **From geometry** in the thickness section of the **Edit Section Assignment** dialog box in the Property module. You cannot use thickness data assigned in the Part module to calculate the offset in an Abaqus/Explicit analysis.

Additional information

- *Editing techniques*

Assigning thickness to shell faces

Defining the thickness in the Part module allows you to define where the midsurface of the solid lies with respect to the shell faces of the midsurface model. The default thickness definition places the midsurface in the center of the thickness. However, you can select surfaces to define a top and bottom thickness, effectively creating an offset between the midsurface and the middle of the shell thickness. Selecting top and bottom faces allows you to easily define different thicknesses at each node and define a different shell offset for each element. This freedom is useful when you use a midsurface model to approximate thin solid models with varying thickness. Using top and bottom faces also allows Abaqus/CAE to automatically update thickness data if you edit the model geometry.

1. From the main menu bar, select **Tools->Midsurface->Assign Thickness and Offset**.

   Tip: You can also assign thicknesses and offsets using the  tool, located with the midsurface tools in the Part module toolbox.

   Shell faces are colored yellow if they are not currently associated with a thickness.

2. If the part contains multiple shell faces, select the faces to which you want to assign thicknesses and offsets (for more information, see *Selecting objects within the viewport*").
3. When you have finished making selections from the viewport, click mouse button 2 to commit your selections.

   Abaqus/CAE opens the **Assign Thickness and Offset** dialog box.
4. If necessary, click the **Edit** button next to **Faces to assign thickness** to change the selections you made in the previous steps.
5. Use one of the following methods to assign a thickness to the selected faces:

Enter or Measure the thickness

Enter a value for thickness, or click  to measure a distance in the viewport and enter it as the thickness.

Compute thickness from opposite faces

Toggle on this option, and click **Select** to select the top and/or bottom faces from the viewport.

Abaqus/CAE highlights the faces in the viewport and calculates the thickness.

If you select only top or only bottom faces, Abaqus/CAE applies the same thickness to both sides of the target shell faces. If you select both top and bottom faces, Abaqus/CAE calculates the thickness and an offset value so that the selected shell faces are not centered within the thickness.

6. Click **OK** to apply the thickness and offset (if used) to the target faces and to close the dialog box.

You can view the thickness of faces in the model by toggling on **Render shell thickness** in the Abaqus/CAE display options. For more information, see *Visualizing shell thicknesses*, and *Controlling shell thickness display*.

Visualizing shell thicknesses

You can view the thickness of your midsurface model in Abaqus/CAE. The thickness representation uses the offsets and thicknesses defined on the part or through the section assignment properties to create the new model view. Visualizing the part thickness along with the reference representation can help you determine whether the midsurface model accurately represents the solid model that you are replacing.

An axisymmetric shell section (smooth curved face) and an extruded shell where faces are joined along a straight edge are shown in *Figure 1*. As shown in the figure, shell thickness is accurately displayed for smooth curved shells; however, where shell faces are joined along edges, the displayed thickness contains gaps and overlaps at the corners between faces.

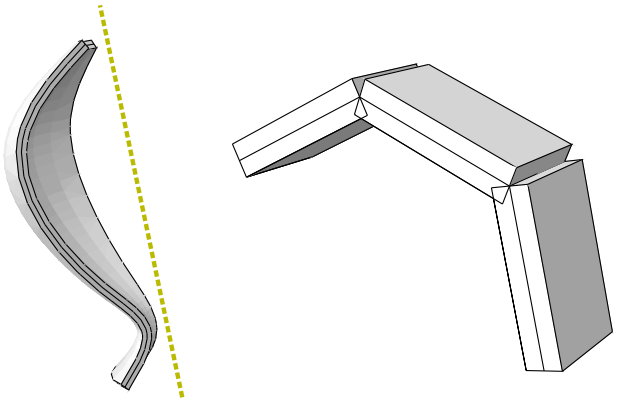


Figure 1: Displaying shell thickness on a smooth curved face and faces joined along edges.

To activate shell thickness display, see *Controlling shell thickness display*.

Additional information

- *Editing techniques*

Skin and stringer reinforcements

This section provides information on how to model skin and stringer reinforcements.

In this section:

- *Defining skin reinforcements*
- *Defining stringer reinforcements*
- *Managing skin and stringer reinforcements*
- *Generating elements on a skin or stringer reinforcement*
- *Assigning element types to skin or stringer reinforcements*
- *Using offset meshes to create skin reinforcements*
- *Assigning surface properties to skins and stringers*
- *Creating and editing skin reinforcements*
- *Creating and editing stringer reinforcements*

Defining skin reinforcements

A skin reinforcement defines a skin that is bonded to the surface of an existing part and specifies its engineering properties. The surface can be a face of a three-dimensional solid part, any edge of an axisymmetric part, or any face of a two-dimensional part. The part can include geometry and orphan mesh elements; however, skin reinforcements must be defined separately for the geometry and the orphan mesh elements. You should think of a skin as a property of a part or region, in the same way a section is a property of a part or region.

The composite beam shown in *Figure 1* is an example of how you might use a skin reinforcement in your model.

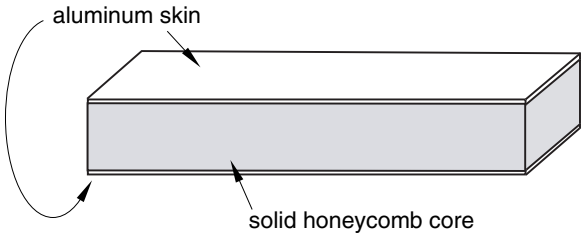


Figure 1: A composite beam modeled by a solid honeycomb core and an aluminum skin.

The beam has a solid honeycomb core and an aluminum skin on the upper and lower faces. You can create a solid part representing the honeycomb and add a skin reinforcement representing the aluminum layers. In the Mesh module you assign solid elements to the honeycomb and shell elements to the skin. The solid and shell elements share the same nodes.

Select **Special->Skin->Create** from the main menu bar in the Property module to define one or more skins. Select **Edit** from the same menu to make changes to an existing definition. All skins you create also appear in the Model Tree in a **Skins** container under the part. Skins are not displayed in the viewport by default, but you can make them visible by color coding them in the viewport. See *Coloring geometry and mesh elements*, for more information.

If you create a skin on a geometry region, Abaqus/CAE updates the skin if you make minor modifications to the underlying geometry. If you edit orphan nodes or elements with a skin, Abaqus/CAE updates the skin if you edit or delete nodes or elements; however, it does not update the skin if you create new nodes or elements.

You may need to select the skin in subsequent modeling operations; for example, to:

- Assign a homogeneous shell section, a composite shell section, a membrane section, a surface section, or a gasket section to a skin. Section assignments are performed in the Property module.
- Assign a material orientation or rebar reference orientation to a skin. Both of these orientation assignments are performed in the Property module.
- Assign a normal direction to a skin in the Property module. Although you cannot assign a normal direction to a skin directly, you can assign a normal direction to a face, which updates the normal direction of all skins defined on that face.
- Prescribe a body force to the skin in the Load module.
- Prescribe a thermal flux on the skin in the Load module. In practice, you perform this modeling operation by applying a thermal flux load to the underlying face or faces. Abaqus/CAE applies the load to all skins on that face during the analysis.
- Assign an element type to the skin in the Mesh module.
- Request field data output or history data output for the skin in the Step module.
- Create a display group to view the stress values on the skin elements in the Visualization module. While you cannot specifically select skins by name in the **Create Display Group** dialog box, you can find them in this dialog box by searching for elements that share the same element type, section assignment, or another property as the skin elements in your model. This process can help you narrow down the list of elements to the skins you want to include.

When you are prompted to select a region for these modeling operations, select **Skins** from the list of object types in the **Selection** toolbar, and select the skin region from the viewport. For more information, see *Filtering your selection based on the type of object*.

When performing contact calculations, Abaqus/CAE considers a geometric surface's skin reinforcements only in certain cases; for example, when you specify general contact for all exterior surfaces, a default all-inclusive surface is defined automatically in Abaqus that will consider skin reinforcements. Skin reinforcements can significantly influence contact calculations, due to skin thickness and potential effects on numerical quantities such as contact penalty stiffness. To explicitly include skin reinforcements in a contact definition, you can create a set on the skin reinforcement and then use the set to define an element-based secondary surface in a contact pair definition in an Abaqus/Standard analysis or use the set to define a node-based secondary surface in a contact definition in an Abaqus/Explicit analysis.

For detailed information on creating a skin, see *Creating and editing skin reinforcements*.

Defining stringer reinforcements

A stringer reinforcement defines a stringer that is bonded to the edge of an existing part and specifies its engineering properties. You can select an edge of a three-dimensional solid part or an edge of a two-dimensional planar part. The part can include geometry and orphan mesh elements; however, stringer reinforcements must be defined separately for the geometry and the orphan elements. You should think of a stringer as a property of a part or region, in the same way a section is a property of a part or region.

The steel-reinforced beam shown in *Figure 1* is an example of how you might use stringer reinforcements in your model.

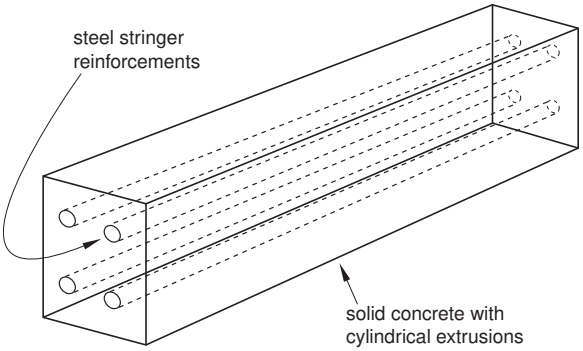


Figure 1: A concrete beam modeled with steel stringer reinforcements.

The beam has a concrete core with four cylindrical extrusions that run the length of the beam. A steel stringer with a circular profile has been created in each extrusion to provide support along the length of the beam. You can create a solid part representing the beam and add four stringer reinforcements representing the steel reinforcements. In the Mesh module you assign solid elements to the concrete and line elements to the stringers. The solid and line elements share the same nodes.

Select **Special->Stringer->Create** from the main menu bar in the Property module to define one or more stringers. Select **Edit** from the same menu to make changes to an existing definition. All stringers you create also appear in the Model Tree in a **Stringers** container under the part. Different stringers can share the same section, and multiple stringers can be placed on an edge of a part. Stringers are not displayed in the viewport by default, but you can make them visible by color coding them in the viewport. See *Coloring geometry and mesh elements*, for more information.

If you create a stringer on a geometry region, Abaqus/CAE updates the stringer if you make minor modifications to the underlying geometry. If you edit orphan nodes or elements with a stringer, Abaqus/CAE updates the stringer if you edit or delete nodes or elements; however, it does not update the stringer if you create new nodes or elements.

You may need to select the stringer in subsequent modeling operations; for example, to:

- Assign a section, beam section orientation, material orientation, or tangent direction to a stringer. All of these activities are performed in the Property module.
- Prescribe a body force or line load to the stringer in the Load module.
- Prescribe a thermal flux on the stringer in the Load module. In practice, you perform this modeling operation by applying a thermal flux load to the underlying edge or edges. Abaqus/CAE applies the load to all stringers on that edge during the analysis.
- Assign an element type to the stringer in the Mesh module.
- Request field data output or history data output for the stringer in the Step module.
- Create a display group to view the stress values on the stringer elements in the Visualization module. While you cannot specifically select stringers by name in the **Create Display Group** dialog box, you can find them in this dialog box by searching for elements that share the same element type, section assignment, or another property as

the stringer elements in your model. This process can help you narrow down the list of elements to the stringers you want to include.

When you are prompted to select a region for these modeling operations, select **Stringers** from the list of object types in the **Selection** toolbar, and select the stringer from the viewport. For more information, see *Filtering your selection based on the type of object*.

When performing contact calculations, Abaqus/CAE considers a geometric surface's stringer reinforcements only in certain cases; for example, when you specify general contact for all exterior surfaces, a default all-inclusive surface is defined automatically in Abaqus that will consider stringer reinforcements. Stringer reinforcements can significantly influence contact calculations, due to stringer thickness and potential effects on numerical quantities such as contact penalty stiffness. To explicitly include stringer reinforcements in a contact definition, you should create a set on the stringer reinforcement and then use the set in a contact definition.

For detailed information on creating a stringer, see *Creating and editing stringer reinforcements*.

Managing skin and stringer reinforcements

The **Skin Manager** and **Stringer Manager** allow you to create and manage skin and stringer reinforcements, respectively. These managers each include a list of the names of skins or stringers that you have defined. The **Create**, **Edit**, **Rename**, and **Delete** buttons in the manager allow you to create new skins and stringers or to edit, rename, and delete existing ones. You can also initiate these procedures using the **Special->Skin** menu and **Special->Stringer** menu from the main menu bar in the Property module. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.



Note:

If you rename a skin or stringer after you have assigned sections, orientations, normals, or other properties to it, the assignments become invalid.

Additional information

- *Creating and editing skin reinforcements*
- *Creating and editing stringer reinforcements*

Generating elements on a skin or stringer reinforcement

You can use the Mesh module to generate two-dimensional elements on a skin reinforcement on a face of a three-dimensional part instance. Similarly, you can generate one-dimensional elements on a stringer reinforcement on an edge of an axisymmetric part instance. Abaqus/CAE generates the skin or stringer elements when the underlying geometry is meshed; you cannot mesh a skin or stringer reinforcement independently.

Figure 1 shows a three-dimensional part instance with a skin reinforcement on the top surface.

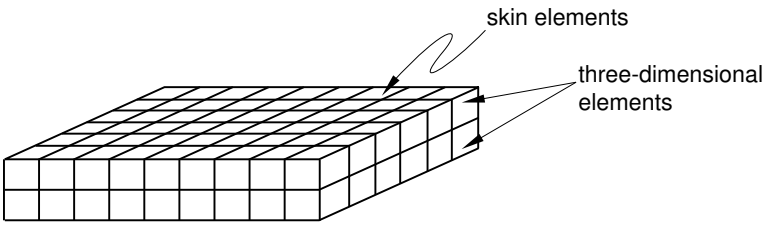


Figure 1: A three-dimensional part instance with a skin reinforcement.

When the part instance is meshed, the skin or stringer elements and the three-dimensional elements share the same nodes and mesh topology. If you assign a different geometric order to the skin elements, you should also change the order of the underlying elements. For more information, see *Assigning element types to skin or stringer reinforcements*.

If you create a skin or stringer on an orphan mesh, the skin or stringer elements and the underlying orphan mesh elements share the same nodes and mesh topology. If you delete an orphan mesh element, Abaqus/CAE automatically regenerates any skins or stringers associated with the deleted element. After regeneration, these skins or stringers have new element ID numbers, so this regeneration renders invalid any sets that include the skins or stringers. Therefore, if you delete elements from an orphan mesh, you should update any set definitions that include skins or stringers formerly associated with the deleted elements.

Assigning element types to skin or stringer reinforcements

You use the Property module to create a skin or stringer reinforcement on a part in your model. You use the Mesh module to assign an element type to a skin or stringer reinforcement. When you are prompted to select the regions to be assigned element types, you must select either **Skins** or **Stringers** from the list of object types in the **Selection** toolbar.

When you create a skin or a stringer for a geometric part, Abaqus/CAE adds an attribute to the geometric entity. This attribute is used when you mesh the part to generate skin or stringer elements, and these elements share nodes with the underlying elements of the part.

You can use stringer reinforcements in a buckling analysis, and you can use skin reinforcements in a coupled structural-acoustic analysis. If the acoustic medium adjoins a structure, structural-acoustic coupling occurs at the interface. It is recommended that you use surface-based tie constraints to enforce the coupling; however, if you are conducting a structural-acoustic submodeling analysis where the acoustic domain forms the submodel, you must line the interface portion of the submodel boundary with acoustic-structural interface (ASI) elements to enforce the coupling. You can line the interface with ASI elements by creating a skin at the interface and assigning ASI elements to the skin. For detailed information about using skins to create ASI elements on geometry and orphan meshes, see the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Using offset meshes to create skin reinforcements

As an alternative approach, you can use the mesh offset tool in the Edit Mesh toolset to create a skin reinforcement by sharing the nodes of the offset shell with the underlying orphan mesh. However, the equivalent offset shell does not appear in the Model Tree as a skin because using the offset shell to represent a reinforcement skin is just one of many potential uses of the offset shell. If you are creating more than one layer of reinforcement skins, you should specify a distance of zero between the layers so that the layers can share nodes.

Assigning surface properties to skins and stringers

If you want to apply a surface property such as a pressure load to a skin or a stringer, you must apply the property to its underlying surface or edge instead. Because skins and stringers share nodes with the surfaces or edges to which they apply, Abaqus/CAE automatically propagates any applied surface properties to skins and stringers as well. For more information about defining surface properties, see *Assigning sections, orientations, normals, and tangents to a part*.

Creating and editing skin reinforcements

During the initial creation of a skin, you select one or more faces that you want to reinforce and, if desired, the number of skins that you want to create.
After the skin is created, you can modify it in the following ways:

- In the Property module, you can modify a skin by assigning it a section (a homogeneous shell section, a composite shell section, a membrane section, a surface section, or a gasket section), a material orientation, or a rebar reference orientation. You can also reverse the normal direction of a skin by assigning a normal direction to its underlying face. When you reverse the normal direction for a face, all skins assigned to that face inherit the new normal direction as well.
- In the Mesh module, you can assign an element type to the skin. When you are prompted to select the regions to be assigned element types, you must select **Skins** from the list of object types in the **Selection** toolbar. If you assign a shell section to a skin, you must assign shell elements to the skin. Similarly, if you assign a membrane, surface, or gasket section, you must assign membrane, surface, or gasket elements.

Abaqus/CAE names skins automatically, but you can rename them from the **Skin Manager**, the **Special** menu, or from the Model Tree. If you create multiple skins at once, Abaqus/CAE appends a unique identifier to the name to identify each skin.

You can obtain data from a skin by using the field and history output request editors in the Step module. In the **Domain** section of the editor, select **Skin** and choose the desired skin from the menu that appears. For more information, see *Creating an output request*.

1. From the main menu bar in the Property module, select **Special->Skin->Create**.

   
   Tip: You can also click the  tool in the Property module toolbox.

2. Use one of the following methods to select the faces to which you want to add skin reinforcements:

Selecting individual faces:

1. Click the arrow next to the **Select entities on which to create a skin** field in the prompt area, and select **individually** from the list that appears.
2. Select a face to which you want to add skin reinforcement.
3. **[Shift] + Click** on additional faces to add them to your selection.
4. If necessary, **[Ctrl] + Click** on selected faces to unselect them.
5. When you have finished selecting faces, click mouse button 2.

Specifying an existing set:

1. Click **Sets** on the right side of the prompt area.

   Abaqus/CAE displays the **Region Selection** dialog box containing a list of sets that you have created.
2. Select the set of faces to which you want to add skin reinforcement, and click **Continue**.


Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

Selecting faces using the angle method:

1. Click the arrow next to the **Select entities on which to create a skin** field in the prompt area, and select **by face angle** from the list that appears.
2. Enter an angle (from 0° to 90°), and select a face. Abaqus/CAE selects every adjacent face from the selected face until the angle between the faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects* for more information.)
3. After you use the **by face angle** method, you can **[Shift] + Click** on additional faces to add them to your selection or **[Ctrl] + Click** on selected faces to remove them from your selection. (See *Combining selection techniques* for more information.)



Tip: You can use the **Selection** toolbar to limit the types of objects that you can select in the viewport. For more information, see *Selecting objects within the viewport*.

3. If desired, toggle on **Multiple skins** in the Prompt area to create more than one skin reinforcement for the selected face or faces.
4. Click **Done** to create the skin. If you are creating multiple skins, enter the number of skin layers you want to create, and click mouse button 2.

Additional information

- *Assigning a section*
- *Assigning a material orientation*
- *Assigning a rebar reference orientation*
- *Assigning shell/membrane normal directions*

Creating and editing stringer reinforcements

You can create a stringer immediately after you define the part; stringers do not initially require material or section assignments.
After you have created a stringer, you can then assign it a section, beam section orientation, material orientation, and tangent direction using the options under the **Special** menu in the Property module. You can also assign an element type to the stringer in the Mesh module. When you are prompted to select the regions to be assigned element types, you must select **Stringers** from the list of object types in the **Selection** toolbar.

You can create multiple stringers at a time on the same edge. Abaqus/CAE treats these stringers as different layers of stringer reinforcement, and you can assign different qualities to each one.

Abaqus/CAE names stringers automatically, but you can rename them from the **Stringer Manager**, the **Special** menu, or from the Model Tree. If you create multiple stringers at once, Abaqus/CAE appends a unique identifier to the name to identify each stringer.

You can obtain data from a stringer by using the field and history output request editors in the Step module. In the **Domain** section of the editor, select **Stringer** and choose the desired stringer from the menu that appears. For more information, see *Creating an output request*.

1. From the main menu bar in the Property module, select **Special->Stringer->Create**.

   Tip: You can also click the  tool in the Property module toolbox.

2. Use one of the following methods to select the edges to which you want to add stringer reinforcement:

Selecting individual edges:

1. Click the arrow next to the **Select entities on which to create a stringer** field in the prompt area, and select **individually** from the list that appears.
2. Select an edge to which you want to add stringer reinforcement.
3. **[Shift] + Click** on additional edges to add them to your selection.
4. If necessary, **[Ctrl] + Click** on selected edges to unselect them.
5. When you have finished selecting edges, click mouse button 2.

Specifying an existing set:

1. Click **Sets** on the right side of the prompt area.

   Abaqus/CAE displays the **Region Selection** dialog box containing a list of element sets that you have created.
2. Select the set of entities to which you want to add stringer reinforcement, and click **Continue**.

Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

Selecting edges using the angle method:

You can use this method to select geometric edges only. This method is not available for selecting orphan mesh element edges.

1. Click the arrow next to the **Select entities on which to create a stringer** field in the prompt area, and select **by edge angle** from the list that appears.
2. Enter an angle (from 0° to 90°), and select an edge or another entity. Abaqus/CAE selects every adjacent edge from the selected edge until the angle between the edges is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects* for more information.)
3. After you use the **by edge angle** method, you can **[Shift] + Click** on additional edges to add them to your selection or **[Ctrl] + Click** on selected edges to remove them from your selection. (See *Combining selection techniques* for more information.)



Tip: You can use the **Selection** toolbar to limit the types of objects that you can select in the viewport. For more information, see *Selecting objects within the viewport*.

3. If desired, toggle on **Multiple stringers** to create more than one stringer for the selected edge or edges.
4. Click **Done** to create the stringer. If you are creating multiple stringers, enter the number of stringers you want to create, and click mouse button 2.

Additional information

- *Assigning a section*
- *Assigning a beam orientation*
- *Assigning a material orientation*
- *Assigning beam/truss tangent directions*

Springs and dashpots

This section provides information on how to model springs and dashpots.

In this section:

- *Defining springs and dashpots*
- *Managing springs and dashpots*
- *Creating springs and dashpots connecting two points*
- *Editing springs and dashpots connecting two points*
- *Creating springs and dashpots connecting points to ground*
- *Editing the region to which springs and dashpots connecting points to ground are applied*

Defining springs and dashpots

You can define springs and dashpots that exhibit the same linear behavior. You can also define both spring and dashpot behaviors on the same set of points. If you define both spring and dashpot behaviors, they act in parallel. You can model springs and dashpots using the following connectivity types:

- Connect two points and follow the line of action between the points
- Connect two points and act in a fixed direction (Abaqus/Standard analyses only)
- Connect points to ground (Abaqus/Standard analyses only)

For more information, see *Springs*, and *Dashpots*.

Select **Special->Springs/Dashpots->Create** from the main menu bar in the Property module or the Interaction module to define springs and dashpots. Select **Edit** from the same menu to make changes to an existing definition. Springs and dashpots created on a part are instanced along with the part.

If you want to include the spring and dashpot results data in the output database generated by the analysis, you must use the **History Output Request** editor in the Step module. Springs and dashpots appear in the Model Tree in the **Engineering Features** container under the part (if created in the Property module) or under the assembly (if created in the Interaction module).

For detailed information, see the following sections:

- *Modifying history output requests*
- *Creating springs and dashpots connecting two points*
- *Editing springs and dashpots connecting two points*
- *Creating springs and dashpots connecting points to ground*
- *Editing the region to which springs and dashpots connecting points to ground are applied*

Managing springs and dashpots

The **Springs/Dashpots Manager** allows you to create and manage springs and dashpots. The manager includes a list of the names and types of springs and dashpots that you have defined. The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** buttons in the manager allow you to create new springs and dashpots or to edit, copy, rename, and delete existing ones. The icons in the column along the left side of the manager allow you to suppress and resume existing springs and dashpots. You can also initiate these procedures using the **Special->Springs/Dashpots** menu from the main menu bar in the Property module or the Interaction module. After you select a management operation from the main menu bar, the procedure is exactly the same as if you had clicked the corresponding button inside the manager dialog box.

Additional information

- *Suppressing and resuming objects*
- *Creating springs and dashpots connecting two points*
- *Editing springs and dashpots connecting two points*
- *Creating springs and dashpots connecting points to ground*
- *Editing the region to which springs and dashpots connecting points to ground are applied*

Creating springs and dashpots connecting two points

You can define springs and dashpots that connect two points and exhibit the same linear behavior independent of field variables.
For more information, see *Springs* and *Dashpots*.

You can obtain history data of stress and strain from a spring/dashpot by using the history output request editor in the Step module. In the **Domain** section of the editor, select **Springs/Dashpots** and choose the desired spring/dashpot from the menu that appears. For more information, see *Creating an output request*.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Springs/Dashpots->Create**.
2. In the **Create Springs/Dashpots** dialog box that appears, name the springs/dashpots, select **Connect two points**, and click **Continue**.
3. Select the first point for the first spring/dashpot using one of the following methods:
   - Select the point in the viewport. (For more information, see *Selecting objects within the viewport*.)

     Tip: The  **Select the Entity Closest to the Screen** tool in the **Selection** toolbar is toggled off by default. If you make an ambiguous selection, Abaqus/CAE highlights the point and displays a description of the point in the lower left corner of the viewport. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

     If the model contains a combination of geometry and mesh components, select one of the following from the prompt area:

     - Select **Geometry** to define the spring/dashpot for geometry or for a reference point.
     - Select **Mesh** to define the spring/dashpot for a mesh.

   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select a set that contains only one point, and click **Continue**.

        Note:

        The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. Select the second point for the first spring/dashpot using one of the methods described in the previous step.

   The first spring/dashpot point pair and a dashed line connecting them are highlighted in the viewport.
5. Continue selecting pairs of points for additional springs/dashpots as described in the previous steps. When you have finished selecting point pairs, click **Done** in the prompt area.

   The **Edit Springs/Dashpots** dialog box appears. The point pairs that you selected to define each spring/dashpot are listed in the **Spring/Dashpot Point Pairs** table.
6. From the **Spring/Dashpot Point Pairs** table, you can do the following:

- To define additional springs/dashpots with the same behavior, click  and repeat the point selection procedures described in the previous steps to define the first and second points of each spring/dashpot.
- To edit a point in the table, select the point in the table, double-click on it or click , and reselect a point. The selection highlighting in the viewport is updated to show the newly edited point.
- To identify a specific spring/dashpot in the viewport, select the desired row number. The highlighted dashed line connecting the selected point pairs appears bolder in the viewport.
- To remove springs/dashpots from the table, select the desired row numbers and click .
- To remove all springs/dashpots from the table, click .

7. Click the arrow next to the **Axis** field, and select an option from the list that appears:
   - Select **Follow line of action** to have the axis of each spring/dashpot follow the line of action of the two points.
   - Select **Specify fixed direction** to specify the degree of freedom at each point on the spring/dashpot (Abaqus/Standard analyses only).
8. If you selected **Specify fixed direction**, do the following in the **Direction** portion of the dialog box:
   - Click the arrow next to the **Point 1 degree of freedom** field, and select the degree of freedom with which the springs/dashpots are associated at their first point.
   - Click the arrow next to the **Point 2 degree of freedom** field, and select the degree of freedom with which the springs/dashpots are associated at their second point.
   - The global coordinate system determines the default spring/dashpot orientations. To specify an orientation for the springs/dashpots, click . Specify the orientation for the springs/dashpots using one of the following methods:
     - Click **Datum CSYS List** from the prompt area to select a name from a list of datum coordinate systems.
     - Select a datum coordinate system in the viewport.
     - Click **Use Global CSYS** from the prompt area to use the global coordinate system.

     Abaqus/CAE displays the local coordinate system specified for springs/dashpots in the viewport.
9. In the **Property** portion of the dialog box, check the appropriate box to include:
   - **Spring stiffness**. Enter the force per relative displacement for springs.
   - **Dashpot coefficient**. Enter the force per relative velocity for dashpots.

   If you define both spring and dashpot behaviors, they act in parallel.
10. Click **OK** to create the springs/dashpots and to close the **Edit Springs/Dashpots** dialog box.

    Symbols appear in the viewport that represent the springs/dashpots that you just created.

Additional information

• *Editing springs and dashpots connecting two points*

- *Controlling the display of attributes*
- *Symbols used to represent special engineering features*

Editing springs and dashpots connecting two points

You can edit springs and dashpots that connect two points to add more point pairs, delete point pairs, or edit a point of an existing point pair.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Springs/Dashpots->Edit->*name*** to display the **Edit Springs/Dashpots** dialog box.

   The symbols representing springs/dashpots are highlighted in the viewport.

2. From the **Spring/Dashpot Point Pairs** table, you can do the following:

   - To define additional springs/dashpots with the same behavior, click  and repeat the point selection procedures described in the previous steps to define the first and second points of each spring/dashpot.
   - To edit a point in the table, select the point in the table, double-click on it or click , and reselect a point. The selection highlighting in the viewport is updated to show the newly edited point.
   - To identify a specific spring/dashpot in the viewport, select the desired row number. The highlighted dashed line connecting the selected point pairs appears bolder in the viewport.
   - To remove springs/dashpots from the table, select the desired row numbers (**[Shift] + Click** to select multiple rows) and click . **[Ctrl] + Click** a row number to unselect it.
   - To remove all springs/dashpots from the table, click .

3. Finish editing the springs/dashpots definition as desired, and click **OK**.

   The symbols representing springs/dashpots in the viewport appear on the newly edited points.

Additional information

- *Controlling the display of attributes*
- *Symbols used to represent special engineering features*

Creating springs and dashpots connecting points to ground

In an Abaqus/Standard analysis you can define springs and dashpots that connect points to ground and exhibit the same linear behavior independent of field variables.
For more information, see *Springs* and *Dashpots*.

You can obtain history data of stress and strain from a spring/dashpot by using the history output request editor in the Step module. In the **Domain** section of the editor, select **Springs/Dashpots** and choose the desired spring/dashpot from the menu that appears. For more information, see *Creating an output request*.

1. From the main menu bar in the Property module or the Interaction module, select **Special->Springs/Dashpots->Create**.
2. In the **Create Springs/Dashpots** dialog box that appears, name the springs/dashpots, select **Connect points to ground**, and click **Continue**.
3. Select the points to connect to ground using one of the following methods:

   - Select the points in the viewport. (For more information, see *Selecting objects within the viewport*.)

     Tip: The **Select the Entity Closest to the Screen** tool in the **Selection** toolbar is toggled off by default. If you make an ambiguous selection, Abaqus/CAE highlights the point and displays a description of the point in the lower left corner of the viewport. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

     If the model contains a combination of geometry and mesh components, select one of the following from the prompt area:

     - Select **Geometry** to define the spring/dashpot for geometry or for a reference point.
     - Select **Mesh** to define the spring/dashpot for a mesh.

   - To select from a list of existing sets, do the following:

     1. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

        Note:

        The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   The **Edit Springs/Dashpots** dialog box appears.
4. In the **Direction** portion of the dialog box, do the following:

   - Click the arrow next to the **Degree of freedom** field, and select the degree of freedom with which the springs/dashpots are associated.
   - The global coordinate system determines the default spring/dashpot orientations. To specify an orientation for the springs/dashpots, click . Specify the orientation for the springs/dashpots using one of the following methods:

- Click **Datum CSYS List** from the prompt area to select a name from a list of datum coordinate systems.
- Select a datum coordinate system in the viewport.
- Click **Use Global CSYS** from the prompt area to use the global coordinate system.

Abaqus/CAE displays the local coordinate system specified for springs/dashpots in the viewport.

5. In the **Property** portion of the dialog box, check the appropriate box to include:
   - **Spring stiffness**. Enter the force per relative displacement for springs.
   - **Dashpot coefficient**. Enter the force per relative velocity for dashpots.

   If you define both spring and dashpot behaviors, they act in parallel.
6. Click **OK** to create the springs/dashpots and to close the **Edit Springs/Dashpots** dialog box. Symbols appear in the viewport that represent the springs/dashpots that you just created.

Additional information

• *Editing the region to which springs and dashpots connecting points to ground are applied*
• *Controlling the display of attributes*
• *Symbols used to represent special engineering features*

Editing the region to which springs and dashpots connecting points to ground are applied

You can edit the region to which springs and dashpots that connect points to ground are applied (Abaqus/Standard analyses only).

1. From the main menu bar in the Property module or Interaction module, select **Special->Springs/Dashpots->Edit->*name*** to display the **Edit Springs/Dashpots** dialog box.
2. In the top part of the dialog box, click .
3. Edit the region using one of the following methods:
   - Select and unselect points in the viewport. When you have finished editing the region, click **Done** in the prompt area. (For more information, see *Selecting objects within the viewport*.)
   - To select from a list of existing sets, do the following:
     1. Click **Sets** on the right side of the prompt area.

        Abaqus/CAE displays the **Region Selection** dialog box containing a list of available sets.
     2. Select the set of interest, and click **Continue**.

     Note:

     The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
4. In the dialog box, finish editing the springs/dashpots definition as desired and then click **OK**.

   The symbols representing springs/dashpots in the viewport change to appear on the newly edited region.

Additional information

- *Creating springs and dashpots connecting two points*
- *Symbols used to represent special engineering features*

Submodeling

You use submodeling to study in detail an area of interest in your model; for example, a region of high stress. In most cases you will mesh the region of interest with a finer mesh, and the submodel can provide an accurate, detailed solution. You can also change the modeling space from a shell global model to a more representative solid submodel—shell-to-solid submodeling.

Creating a submodel is a two-step process. First you create and analyze the global model. You then create the submodel and drive the boundaries of the submodel with time-dependent variables that were saved during the analysis of the global model. You can drive submodel boundaries either with boundary conditions or in some cases with stresses from the global model. Submodeling is described in detail in *About Submodeling*. *Shell-to-solid submodeling and shell-to-solid coupling of a pipe joint*, includes an example of a submodel created using Abaqus/CAE.

In this section:

- *Analyzing the global model*
- *Creating a submodel*
- *Removing regions*
- *Creating the submodel boundary condition*
- *Creating the submodel load*
- *Modifying the submodel*
- *Analyzing the submodel*
- *Checking the results from the submodel*

Analyzing the global model

You first obtain results for the entire model using a relatively coarse mesh and perhaps simplified geometry. This model is called the global model. Data from the output database that is generated by the global model are used to drive the submodel. As a result, the output requests in the global model must include the driven variables. You can also drive a submodel using the global model results stored in a results file that was generated from the Abaqus execution procedure.

Creating a submodel

When you have successfully analyzed the global model and generated an output database or a results file containing the global model results, you are ready to create the submodel. You start by copying the global model to a new model that you will use to define the submodel. To copy a model, select **Model->Copy->*global model name*** from the main menu bar. Enter the name of the submodel in the **Copy Model** dialog box that appears and click **OK**. The copied model becomes the current model.

You must select the output database or results file from the global analysis that will be used to drive the submodel. From the main menu bar, select **Model->Edit Attributes->*submodel name***. From the **Edit Model Attributes** dialog box that appears, click the **Submodel** tab, and do the following:

- Toggle on **Read data from job**, and enter the name of the output database or results file containing the global model that will drive the submodel. You must provide the file extension (`.odb` or `.fil`) if both an output database and a results file exist and they use the same name.
- In addition, if your shell global model is driving a solid submodel, toggle on **Shell global model drives a solid submodel**.
- Click **OK** to close the **Edit Model Attributes** dialog box.

Removing regions

Remove regions from the submodel that you are not interested in analyzing. Only the regions of interest should remain. You can use several techniques to remove regions from a part.

- Use the cut tools in the Part module. For more information, see *Adding a cut feature*.
- Use the Geometry Edit toolset to remove faces. For more information, see *Editing techniques*.
- You can delete the part in the Part module and create a new part with the same name. You can then create geometric features that represent the submodel; however, you must position the new features in the same location as the original part. For a discussion of the tolerances between the submodel and the global model, see *About Submodeling*. The new part and the original part must be created in three-dimensional modeling space.

  Creating a new part is a useful technique for converting a shell global model to a solid submodel. You can delete the shell part from the copied model and create a new solid part in the same location. You must take care to ensure that the part defined in the solid submodel is contained within the part defined in the shell global model.

Creating the submodel boundary condition

The most common submodeling technique is node-based submodeling, which uses a nodal results field (including displacement, temperature, or pressure degrees of freedom) to interpolate global model results onto the submodel nodes. Node-based submodeling is also a more general technique. To use node-based submodeling, you create a submodel boundary condition.

If you apply a submodel boundary condition to nodes that are constrained by either a displacement/rotation boundary condition or a connector displacement boundary condition on the global model in a previous step and the global model's boundary condition is fixed using the **Fixed at Current Position** method, Abaqus/CAE disregards the submodel boundary condition for those nodes and retains the specifications in the boundary condition on the global model instead. Abaqus/CAE reports this replacement of boundary conditions in the data file for the analysis.

1. Enter the Load module and select **BC->Create** from the main menu bar.
2. From the list of steps, select the step during which the submodel boundary condition will be applied.
3. From the **Category** field, select **Other**.
4. From the **Types for Selected Step** field, select **Submodel** and click **Continue**.
5. From the model, select the regions to which the boundary condition will be applied. In most cases you apply the boundary condition to the edges and faces that were created when you cut away regions from the global model. You can prescribe other boundary conditions to the same regions; for example, a symmetry boundary condition. The prescribed boundary conditions take precedence over the submodel boundary condition.
6. From the **Edit Boundary Condition** dialog box that appears, do the following:
   a. In the **Driving region** field, do one of the following:
      - Select **Automatic** to allow Abaqus/CAE to create the driving region by searching all regions in the global model that lie in the vicinity of the submodel.
      - Select **Specify** to specify a set name that will be used as the driving region. You must give the complete name of the set. The syntax for the set name is `assembly_name.part_name-1.set_name`, assuming that you are defining the driving region on the first instance of the part.

   b. If you are driving a solid submodel with a shell global model, you must enter the maximum value of the shell thickness in the global model in the **Shell thickness** field.
   c. In the **Exterior tolerance** field, do the following:
      - Enter the **absolute** exterior tolerance. This is the absolute value by which a driven node of the submodel may lie outside the elements of the global model. The default value is the relative exterior tolerance.
      - Enter the **relative** exterior tolerance. This is the fraction of the average element size in the global model by which a driven node of the submodel may lie outside the elements of the global model. The default value is .05.

      For more information, see *About Submodeling*.
   d. If you are driving a solid submodel with a solid global model or if you are driving a shell submodel with a shell global model, you must enter a comma-separated list indicating the degrees of freedom that are being driven; for example, `1,2,3`. You cannot leave this field blank.
   e. If you are driving a solid submodel with a shell global model, you can provide the thickness of the center zone size around the shell midsurface. The default value is 10% of the maximum shell thickness in the global model as defined in the **Shell thickness** field.
   f. In the **Global step number** field, enter an integer representing the step number in the global analysis from which the values of the driven variables will be read.

g. If you created the boundary condition in a static, linear perturbation step, you can specify the increment in the global analysis step that will be the basis for calculating the values for the driven variables. The default value of zero corresponds to the last increment of the previous step.

h. If the time period of the submodel analysis is different from the time period of the global analysis, you can choose to scale the time period of the global step to match the time period of the submodel step. For example, Abaqus determines the displacements of the global model at a time 20% into the global step and applies those displacements at a time 20% into the submodel step.

If you do not choose to scale the time period of the global step to match the time period of the submodel step, Abaqus applies the displacements of the global model at the same time during the submodel step. For example, Abaqus determines the displacements of the global model one second into the global step and applies those displacements one second into the submodel step. This behavior is probably not desired if the two time periods are different. You choose to scale the time period by toggling on **Scale time period of global step to time period of submodel step**.

i. If you want Abaqus to ignore driven nodes found to lie outside the region of elements of the global model after accounting for the exterior search tolerance, toggle on **Intersection only**. This parameter is available only for node-based submodeling.

Additional information

- *Defining a displacement/rotation boundary condition*
- *Defining a connector displacement boundary condition*

Creating the submodel load

The surface-based submodeling technique is an alternative submodeling technique that uses the stress field to interpolate global model results onto the submodel integration points on the driven element-based surface facets. To use surface-based submodeling, you create a submodel load.

1. Enter the Load module and select **Load->Create** from the main menu bar.
2. From the list of steps, select the step during which the submodel load will be applied.
3. From the **Category** field, select **Other**.
4. From the **Types for Selected Step** field, select **Submodel** and click **Continue**.
5. From the model, select the regions to which the load will be applied. In most cases you apply the load to faces that were created when you cut away regions from the global model. If you are applying the load to a face of a shell, Abaqus/CAE asks you to specify the side of the face to which the load will be applied. For more information, see *Specifying a particular side or end of a region*.
6. From the **Edit Load** dialog box that appears, do the following:
   a. In the **Driving region** field, do one of the following:
      - Select **Automatic** to allow Abaqus/CAE to create the driving region by searching all regions in the global model that lie in the vicinity of the submodel.
      - Select **Specify** to specify a set name that will be used as the driving region. You must give the complete name of the set. The syntax for the set name is `assembly_name.part_name-1.set_name`, assuming that you are defining the driving region on the first instance of the part.

   b. In the **Exterior tolerance** field, do the following:
      - Enter the **absolute** exterior tolerance. This is the absolute value by which a driven node of the submodel may lie outside the elements of the global model. The default value is the relative exterior tolerance.
      - Enter the **relative** exterior tolerance. This is the fraction of the average element size in the global model by which a driven node of the submodel may lie outside the elements of the global model. The default value is .05.

      For more information, see *About Submodeling*.
   c. In the **Global step number** field, enter an integer representing the step number in the global analysis from which the values of the driven variables will be read.

Modifying the submodel

You can make the following modifications to the submodel:

- You can use the Step module to change the analysis procedure. The submodel can use either a general procedure or a linear perturbation procedure. For more information, see *About Submodeling*.
- In the Load module you must remove any loads, boundary conditions, or initial conditions that were applied to regions of the global model that were removed.
- If a boundary condition is applied outside the region to which you applied the submodel boundary condition, you must ensure that it corresponds to the loading of the global model.
- Similarly, if a load is applied to the submodel, you must ensure that it corresponds to the loading of the global model.
- In most cases you will apply a more refined mesh to the submodel in the Mesh module. You can change the element type assigned to the submodel; however, you cannot change the dimensionality. Both the global model and the submodel must be either two-dimensional or three-dimensional.

Analyzing the submodel

To analyze your submodel in the Job module, do the following:

- Create a new job using the model containing the submodel.
- Submit the new job for analysis.

Checking the results from the submodel

After the analysis is complete, you can use the Visualization module to overlay contour plots from the submodel and the global model. For a meaningful comparison, the layers of the overlay plots should use the same legend scale and the same deformation scale factor. For more information, see *Overlaying multiple plots*.

You should check the following:

The submodel location

You should check that the position of the submodel is correct, relative to the global model. You can check the relative positions using overlay plots from the global and submodel output databases. Alternatively, you can check the relative positions in the Assembly module before you submit the submodel for analysis by creating temporary instances of the parts in the global model. You can view the position of the assembly in the global model relative to the position of the assembly in the submodel. You can then delete or suppress the instances of the parts in the global model before you mesh and analyze the submodel.

The submodel response does not influence the global response

You should check that the submodel response has insignificant impact on the global response. This is the fundamental assumption of submodeling. You can check this by creating overlay contour plots of variables like stress and strain. The contours should be reasonably continuous across the submodel boundary.

Substructures

This section explains how to integrate substructures into your analysis in Abaqus/CAE.

In this section:

- *Using substructures in Abaqus/CAE*
- *Generating a substructure*
- *Specifying the retained nodal degrees of freedom and load cases for a substructure*
- *Importing a substructure into Abaqus/CAE*
- *Using substructure part instances in an assembly*
- *Activating load cases during substructure usage*
- *Recovering field output for substructures*
- *Visualizing substructure output*

Using substructures in Abaqus/CAE

Substructures are collections of elements that have been grouped together, so the internal degrees of freedom have been eliminated for the analysis.
Using a substructure makes model definition easier and analysis faster when you analyze a model that contains identical pieces that appear multiple times (such as the teeth of a gear), because you can use a substructure repeatedly in a model. Substructures are connected to the rest of the model by the retained degrees of freedom at the retained nodes. Factors that determine how many and which nodes and degrees of freedom should be retained are discussed in *Generating Substructures*. Substructure definition in your model follows two sets of steps.

In this section:

- *Creating substructures in your model database*
- *Including substructures in your analysis*

Creating substructures in your model database

You can create substructures in Abaqus/CAE by following these general steps:

1. Create or open the model database in which you want to specify substructures in Abaqus/CAE.
2. In the Step module, create a **Substructure generation** step. Abaqus/CAE converts the entire model into a single substructure. For more information, see *Generating a substructure*.
3. In the Load module, create **Retained nodal dofs** boundary conditions to determine which degrees of freedom will be retained as external degrees of freedom on the substructure. You can also define a load case in the substructure generation step if you want to apply a load to the substructure at a location other than its retained degrees of freedom. For more information, see *Specifying the retained nodal degrees of freedom and load cases for a substructure*.
4. In the Job module, create a new job and submit the analysis.

   When you perform an analysis of an assembly that includes substructure data, Abaqus/CAE creates separate output databases for the results of each substructure part instance and does not include the results from the substructure part instances in the output database for the assembly. The Visualization module provides tools that enable you to integrate the results from the substructure components back into the results from the assembly; for more information, see *Visualizing substructure output*.

Including substructures in your analysis

Substructure usage should be performed in a different model than substructure generation. You can include substructures in your analysis in Abaqus/CAE by following these general steps:

1. Import each substructure that you want to use in your model database from the corresponding `.sim` file. For more information, see *Importing a substructure into Abaqus/CAE*.
2. In the Assembly module, instance each substructure part that you want to add to the assembly, and position the substructure part instances in the desired locations in the assembly. *Using substructure part instances in an assembly*, explains the capabilities and limitations of substructure part instances.
3. In the Load module, activate substructure load cases by creating a **Substructure load** definition. For more information, see *Activating load cases during substructure usage*.
4. In the Step module, create a field output request with **Substructure** as the **Domain**, then select the substructure sets for which you want to recover field data. For more information, see *Recovering field output for substructures*.
5. In the Interaction module, apply constraints to attach the substructure part instance to the rest of the assembly.

Generating a substructure

The first step in substructure definition is the addition of a **Substructure generate** step in your analysis. The substructure generation step enables you to create a substructure in your model database and, if desired, specify substructure-related options such as the writing of the recovery matrix, stiffness matrix, mass matrix, and load case vectors to a file. These options are described later in this section.

A single analysis can include multiple substructure generate steps, and Abaqus/CAE creates corresponding output database files for each step. Multiple preloading steps can precede every substructure generation step in your analysis. If you want to specify retained eigenmodes for substructure generation, you must also include a frequency extraction step in the analysis.

You cannot have a substructure generation step in a model that contains model instances.

Substructure identifier

You must specify a unique identifier for each substructure you create. Substructure identifiers must begin with the letter Z followed by a number that cannot exceed 9999.

Recovery options

You can recover the field output data for a substructure during the usage-level analysis, but you must specify the recovery region during substructure generation. Substructure recovery can be performed only on the sets included in the recovery region. You can specify that recovery be performed on the whole model or for an individual node set or element set. While performing the substructure recovery in the usage model, Abaqus/CAE must have access to the substructure's `.mdl`, `.prt`, and `.stt` files. For more information about these file types, see *Generating Substructures*.

Generation options

You can control several aspects of the substructure generation process, including calculation of gravity load vectors, evaluation of frequency-dependent material properties, and generation of a reduced mass matrix, reduced structural damping matrix, and viscous damping matrix.

Retained eigenmodes

You can specify retained eigenmodes for generation of a coupled acoustic-structural substructure. When you choose to specify retained eigenmodes, Abaqus/CAE enables you to specify eigenmodes by mode range or by frequency range.

Damping

You can specify several global damping controls and substructure damping controls. For global damping you can choose to apply damping settings to acoustic or mechanical options; for substructure damping you can specify separate controls for viscous and structural damping.

Additional information

- *Configuring a substructure generation procedure*

Specifying the retained nodal degrees of freedom and load cases for a substructure

After you defined the substructure generation step or steps for your analysis, you must define a **Retained nodal dofs** boundary condition for a substructure. The retained degrees of freedom for a substructure node are the degrees of freedom that are external and are available for use in the analysis; all other degrees of freedom for the specified node are assumed to be internal to the substructure and do not factor into the analysis. When you import a substructure from this analysis into a model for substructure usage, Abaqus/CAE displays these nodes as light blue crosses, which enables you to pick them easily from a part instance or assembly.

If you want to apply a load to the substructure at a location other than its retained degrees of freedom, you can define a load case in the substructure generation step.

Additional information

- *Defining a retained nodal DOFs boundary condition*
- *Defining a load case*

Importing a substructure into Abaqus/CAE

You can include substructure definitions in a model database and begin to use them for modeling by importing the substructures as new part definitions. Substructure data are available in `.sim` files, and the substructure identifier is included in the file name; for example, in an analysis in which the substructure is named `FAN` and the substructure identifier is `Z400`, the substructure database file is named `FAN_Z400.sim`.

The `.sim` file from which you import a substructure must reside in the same directory as the supporting Abaqus files to which the `.sim` database refers; these supporting files might include data in the formats `.prt`, `.mdl`, or `.stt`.

Substructure import also requires an output database (`.odb`) file for mesh display.

Additional information

- *What kinds of files can be imported and exported from Abaqus/CAE?*
- *Importing a substructure into a model database as a part*

Using substructure part instances in an assembly

Once you import substructure parts into your model database, you can add them to your assembly by instancing them in the same manner you would for any part. Substructure part instances are displayed in a translucent color in the viewport.

You can move and apply constraints to substructure part instances; however, substructure part instances have the following modeling limitations:

- You cannot assign sections to a substructure part instance.
- You cannot apply attributes to a substructure part instance.
- Substructure part instances are not eligible for definition of contact pairs.
- Gravity loads are the only load definition that can be applied to substructure part instances.

Additional information

• *Creating and manipulating part and model instances*

Activating load cases during substructure usage

The **Substructure load** definition enables you to activate the substructure load cases that are specified during the substructure generation step. As you activate a load case, you can scale its load definitions or apply an amplitude to them.

Additional information

• *Defining a substructure load definition to activate a substructure load case*

Recovering field output for substructures

You can specify that Abaqus/CAE write field output data for one or more substructure sets in your analysis.

From the field output editor, select **Substructure** from the **Domain** field, then click  to open the **Select Substructure Sets** dialog box. This dialog box lists only the substructure sets that were defined while generating the substructure. You cannot recover data for sets that you define on substructure part instances in Abaqus/CAE.

Additional information

• *Modifying field output requests*

Visualizing substructure output

Abaqus/CAE creates separate output database (`.odb`) files for each substructure part instance used in the analysis, so you must perform some additional steps if you want to display substructure results in context with the rest of the assembly. The Visualization module provides the following tools that enable you to incorporate substructure results into the rest of the model:

- You can use an overlay plot to display plots of substructure data in the same viewport as a plot of the rest of the assembly.
- You can use the **Combine ODBs** plug-in to combine the data in one or more substructure output database files with the data from the rest of the assembly.

Additional information

- *Producing and modifying overlay plots*
- *Combining data from multiple output databases*

Viewing results

This part describes how to use the Visualization module (also licensed separately as Abaqus/Viewer) to view your model and the results of your analysis.

In this section:

- *Visualization module basics*
- *Viewing diagnostic output*
- *Selecting model data and analysis results to plot*
- *Plotting the undeformed and deformed shapes*
- *Contouring analysis results*
- *Plotting analysis results as symbols*
- *Plotting material orientations*
- *X–Y plotting*
- *Viewing results along a path*
- *Animating plots*
- *Querying the model in the Visualization module*
- *Probing the model*
- *Calculating linearized stresses*
- *Viewing a ply stack plot*
- *Generating tabular data reports*
- *Customizing plot display*
- *Customizing viewport annotations*

Visualization module basics

You can use the Visualization module to view your model and the results of your analysis.

In this section:

- *Understanding the role of the Visualization module*
- *Entering and exiting the Visualization module*
- *Understanding plot states and plot customization*
- *Understanding toolsets in the Visualization module*
- *Understanding Visualization module performance*

Understanding the role of the Visualization module

The Visualization module provides graphical display of finite element models and results. It obtains model information from the current model database or model and result information from an output database. You can control what information is placed in the output database by modifying output requests in the Step module. (For more information, see *What is an output request?*.) You can view data from a model database using a contour plot or a symbol plot; you can view model and results data from an output database by producing any of the plots described in this section.



Undeformed shape

An undeformed shape plot displays the initial shape or the base state of your model.



Deformed shape

A deformed shape plot displays the shape of your model according to the values of a nodal variable such as displacement.



Contours

For an output database, a contour plot displays the values of an analysis variable such as stress or strain at a specified step and frame of your analysis. For a model in the current model database, a contour plot displays the value of a load, a predefined field, or an interaction at a selected step of your model. The Visualization module represents the values as customized colored lines, colored bands, or colored faces on your model.



Symbols

For an output database, a symbol plot displays the magnitude and direction of a particular vector or tensor variable at a specified step and frame of your analysis. For a model in the current model database, a symbol plot displays the magnitude and direction of a load, a predefined field, or an interaction at a specified step of your model. The Visualization module represents the values as symbols (for example, arrows) at locations on your model.



Material orientations

A material orientation plot displays the material directions of elements in your model at a specified step and frame of your analysis. The Visualization module represents the material directions as material orientation triads at the element integration points.



X–Y data

An *X–Y* plot is a two-dimensional graph of one variable versus another.



Time history animation

Time history animation displays a series of plots in rapid succession, giving a movie-like effect. The individual plots vary according to actual result values over time.

 Scale factor animation

Scale factor animation displays a series of plots in rapid succession, giving a movie-like effect. The individual plots vary in the scale factor applied to a particular deformation.

 Harmonic animation

Harmonic animation displays a series of plots in rapid succession, giving a movie-like effect. The individual plots vary according to the angle applied to the complex number results being displayed.

Additional capabilities include:

Visualizing diagnostic information

Diagnostic information helps you determine the causes of nonconvergence in a model. You can view information for each stage of the analysis and use Abaqus/CAE to highlight problematic areas on the model in the viewport.

Understanding probing

Probing displays model data and analysis results as you move the cursor around a model or a model plot; probing an *X*–*Y* plot displays the coordinates of graph points. You can write this information to a file.

Results plotting along a path

A path is a line you define by specifying a series of points through your model. You can view results along the path in the form of an *X*–*Y* plot.

Stress linearization

Stress linearization is the separation of stresses through a section into constant membrane and linear bending stresses. You specify the section as a path through your model, and the Visualization module displays the linearized stresses in the form of an *X*–*Y* plot.

Cutting through your model

View cuts allow you to slice through a model so that you can visualize the interior or selected sections of the model. You can define planar, cylindrical, or spherical view cuts. In addition, you can define a view cut along a constant contour variable value.

X–Y and field output reporting

An *X*–*Y* report is a tabular listing of *X*- and *Y*-data values; a field output report is a tabular listing of field output values.

Visualizing the plies in a composite layup

A ply stack plot is a graphical representation of the plies in a composite layup. The image shows the plies in the layup along with details of each ply, such as its fiber orientation, thickness, and the reference plane. You can also create a ply stack plot in the Property module while you are creating a composite layup.

Plot customization

The Visualization module provides numerous options that you can use to customize your plots.

Entering and exiting the Visualization module

You can enter the Visualization module at any time during an Abaqus/CAE session by clicking **Visualization** in the **Module** list located in the context bar. The **Result**, **Plot**, **Animate**, **Report**, **Options**, and **Tools** menus appear on the main menu bar; and the title bar of the current viewport displays the name of the current output database, if one exists.

You can also enter the Visualization module by opening an existing output database. To open an output database, select **File->Open** from the main menu bar (for more information, see *Opening a model database or an output database*) or click **Results** in the **Job Manager** when results are available for a selected job. When you use one of these methods to enter the Visualization module, Abaqus/CAE displays a plot of the model from the output database in the current viewport.

To exit the Visualization module, select any other module from the **Module** list, or end the session by selecting **File->Exit** from the main menu bar. When you end the session, Abaqus/CAE closes all files and windows. Abaqus saves your plot options only for the duration of the session.

Understanding plot states and plot customization

This section describes how to customize the appearance of a plot by selecting plot customization options.

In this section:

- *What is a plot state?*
- *Activating plot states*
- *Customizing your plots*
- *Customizing multiple viewports*

What is a plot state?

The Visualization module offers several distinct types of plots for viewing your model and results. These plot types are:

- *Undeformed shape*
- *Deformed shape*
- *Contour*
- *Symbol*
- *Material orientation*
- *History or X–Y data*
- *Time history animation*
- *Scale factor animation*
- *Harmonic animation*

Each of these plots corresponds to a *plot state*. Plot states are important because some of the customization options provided by the Visualization module pertain only to a particular plot state.

Additional information

- *Activating plot states*
- *Customizing your plots*

Activating plot states

A plot state combines all of the active customization options to produce a plot in the viewport. Some options are common to all plot types, some are applied only when you choose to superimpose the deformed and undeformed model shapes, and some are specific to the current plot type. You enter a particular plot state by producing a plot of the corresponding type. For example, if you produce an undeformed plot, the current viewport will then be in the undeformed plot state. The plot state of a viewport persists and Abaqus/CAE updates it with any changes you make to the customization options until you produce a plot in some other state in that viewport. If you create multiple viewports, each viewport can contain a different plot state. In addition, you can choose to allow multiple plot states in a single viewport by plotting both the undeformed and deformed shape for a single plot type or by displaying multiple plot types in the viewport.

Additional information

• *Customizing your plots*

Customizing your plots

The Visualization module provides numerous customization options, which are available through the **Viewport**, **Options**, and **View** menus of the main menu bar. These options fall into three categories:

Plot state–dependent options

Plot type options affect only a particular plot state. These are separate options affecting contour, symbol, and material orientation plots; *X–Y* curves; *X–Y* graphs; time history animations; scale factor animations; and harmonic animations.

Plot state–independent options

Plot state–independent options are those that affect all plots collectively. These are common options governing the viewpoint, graphics, individual item coloring, display body appearance, and such general characteristics as plot legends, model labels, and the appearance of text blocks giving the model's title and state.

Superimpose options

Superimpose options are a special set of plot state–independent options that affect only the undeformed plot state when you choose to superimpose it on a deformed, contour, symbol, or material orientation plot. These options allow you to control many common customization options to distinguish the undeformed plot from the deformed plot when both shapes are displayed.

Plot state–dependent options affect such plot attributes as the contour intervals, limits, and colors (for example, color spectrums for contour plots or axis colors for material orientation plots). You control these attributes separately for each plot state using the options associated with that state. To choose the contour type, for example, you must use the **Contour** plot options. To do so, you can select **Options->Contour** from the main menu bar. Alternatively, you can use the **Contour Options** tool  from the toolbox. The **Options** tools provide quick access to the plot state–dependent and plot state–independent customization options.

Plot state–dependent options affect only plots in the associated state. If you select axis colors from the material orientation plot options dialog box, those colors will affect only material orientation plots. If there is a material orientation plot in the current viewport, you will see the effect of your changes when you click **Apply** or **OK** in the material orientation plot options dialog box. However, if the current viewport does not contain a material orientation plot, you will not see the effect of your changes until you produce one.

Plot state–independent options affect plots across all plot states. For example, if you select **Viewport->Viewport Annotation Options** from the main menu bar to suppress the appearance of the view triad, the view triad will be suppressed for all plots. Settings in the **Common Plot Options** dialog box are also plot state–independent; when you change the render style in this dialog box, the new render style is used for all plots.

Superimpose options affect the undeformed plot when you superimpose an undeformed plot on a deformed plot in one of the plot states. For example, you can independently set the render style, edge display, and fill color; and you can apply an offset between the undeformed and deformed shape symbol plots when both are displayed.

Select **File->Save Options** from the main menu bar to save your plot state–dependent, plot state–independent, and superimpose customization options. Saving your customization options allows you to apply them to subsequent Abaqus/CAE sessions. For more information, see *Saving your display options settings*. For more information on the plot customization options available in the Visualization module, see *Customizing plot display*.

Customizing multiple viewports

When you create a new viewport, it initially inherits the customization options of the current viewport. For example, if you *establish the **Filled** render style* for plots in the current viewport and then create a new viewport, subsequent plots in the new viewport will appear in the filled render style. New viewports inherit the plot state of the current viewport—until you change customization options, plot states, or output databases, new viewports appear identical to the viewport that was active when you created them.

After a new viewport has been established, the plot state and any *subsequent* customizations are independent of other viewports by default. Multiple viewports can each be in a separate plot state; if you use multiple viewports, you must first designate a particular viewport as current to change its display. Customization selections you apply affect only the current viewport. When you designate a viewport as current, the options dialog boxes are refreshed to show the state of options associated with that viewport. For more information on working with viewports, see *Working with viewports*.

You can also link multiple viewports in your session to manipulate multiple objects simultaneously and to display the same plot state and the same plot options in different viewports. For more information, see *Linking viewports for view manipulation*.

Additional information

• *Activating plot states*

Understanding toolsets in the Visualization module

Toolsets in the Visualization module provide additional control over data display.
Unless otherwise stated, the toolsets described in this section are for postprocessing of model and results data from output databases only; only selected toolsets are for use with model data from the current model database.

The following toolsets are available in the Visualization module:

- The Color Code toolset allows you to customize the edge and fill color of individual elements. For more information, see *Coloring nodes or elements in the Visualization module*.
- The Coordinate System toolset allows you to create local coordinate systems for use in postprocessing. For more information, see *Creating coordinate systems during postprocessing*.
- The Create Set toolset allows you to create node and element sets during postprocessing. For more information, see *Creating node and element sets during postprocessing*.
- The Display Group toolset allows you to selectively plot one or more items from a model or from an output database. For more information, see *Using display groups to display subsets of your model*.
- The Field Output toolset allows you to perform operations on the field output available in an output database. For more information, see *Creating and saving new field output*.
- The Free Body toolset allows you to create and delete free body cuts, display or hide them in the viewport, and customize several aspects of their appearance. For more information, see *The Free Body toolset*.
- The Job Diagnostics toolset allows you to access the diagnostic information written to the output database during an Abaqus/Standard analysis job. For more information, see *Viewing diagnostic output*.
- The Path toolset allows you to specify a path through your model along which you can obtain and view *X–Y* data. For more information, see *Viewing results along a path*.
- The Query toolset allows you to obtain information about your model, both for data from the current model database or for data from an output database. For more information, see *Querying the model in the Visualization module*.
- The Stream toolset allows you to display streamlines to investigate velocity or vorticity in a fluid flow analysis. For more information, see *The Stream toolset*.
- The View Cut toolset allows you to create cuts through a model so that you can visualize the interior or selected sections of the model. This functionality is available for model data from the current model database or from an output database. For more information, see *Cutting through a model*.
- The XY Data toolset allows you to create and operate on *X–Y* data objects. For more information, see *X–Y plotting*.

Understanding Visualization module performance

In general, the speed of postprocessing and graphical results display in the Visualization module is more than satisfactory for most models. However, it is often necessary to balance high performance levels with detailed results plotting. Depending on your postprocessing requirements, you may wish to modify the default display options at the expense of performance. Many options are available that can affect the speed of graphics display.

To maximize performance in the Visualization module, we recommend the following:

- Cache results in memory during postprocessing to speed up the generation of images on the screen. See *Understanding results caching*.
- Use the texture-mapped contour method whenever possible. Likewise, avoid using display options that are not supported by the texture-mapped method: line-type contours, contour edges, CAXA or SAXA elements, or shrinking elements about their centroid. If such display options are used, Abaqus/CAE will override the user setting for texture-mapped contours and use the slower tessellated method instead. See *Understanding how contours are rendered*.
- Be selective about the model labels and symbols that you choose to display. The more model entities that are drawn, the longer the screen refresh will take. This applies to element edges as well. See *Controlling the display of model entities*.
- Use a high results averaging threshold. The more contiguous the results are, the faster the contour display will be. See *Understanding result value averaging*.
- Use the status field output variable rather than creating display groups based on result values to achieve high-performance time history animation. Abaqus/CAE recomputes result-based display groups as each result frame is displayed, so using this type of display group degrades animation performance. The status field output variable allows you to remove elements that meet a result-based failure criteria from your model plots. See *Selecting the status field output variable*.
- Do not use a remote display. This configuration of Abaqus/CAE is not supported; its use is at the sole discretion of the user. Performance optimization can never be achieved using remote display.

In some cases performance may not be optimal for element-based results plotting on very large models because of physical memory limits on your machine. Increasing the physical memory or exiting other applications that are consuming memory can help to restore optimal performance.

Viewing diagnostic output

Abaqus/CAE provides a visual diagnostics tool to help you understand the convergence behavior of your job. You can use diagnostic output to assess the quality of analysis results or to locate the source of convergence problems in a model.

This chapter explains how you locate and use diagnostic output within Abaqus/CAE.

In this section:

- *Understanding job diagnostics*
- *Generating diagnostic information*
- *Interpreting diagnostic information*
- *Accessing diagnostic information*

Understanding job diagnostics

Abaqus writes diagnostic information to the output database, along with any other output that you request, as it attempts to analyze your model.
The diagnostic information in the output database is a subset of the diagnostic information that is written to the message and status files. You can use the **Job Diagnostics** dialog box to access the diagnostic information written to the output database during an Abaqus/Standard or Abaqus/Explicit analysis job.

For an Abaqus/Standard analysis, diagnostic information is available for the job and for each step, increment, attempt, and iteration. *Figure 1* shows contact diagnostics for an Abaqus/Standard analysis iteration. You can choose the type of contact information that you want to view, and you can select a column to sort the table data.

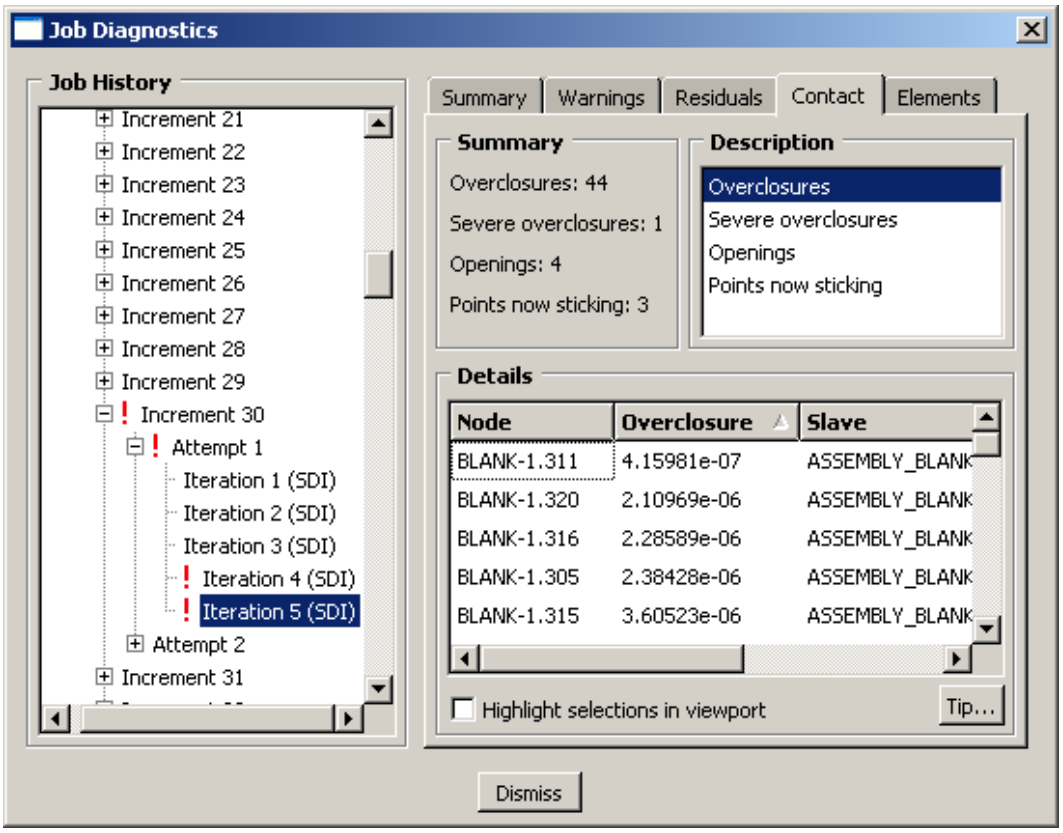


Figure 1: The Job Diagnostics dialog box.

For an Abaqus/Explicit analysis, diagnostic information is available for the job and for each step. Because of the large number of increments in a typical Abaqus/Explicit analysis, diagnostic information is recorded for summary (heartbeat) increments and for any other increments in which diagnostic information is generated. The interval between summary increments depends on the CPU time and the amount of output specified in the analysis. The diagnostic data are written to the status file and to the output database.

You can use the **Job Diagnostics** dialog box to determine when an analysis ended and whether any warnings were issued. If warnings were issued during the job, you can view the warnings and assess whether your results may have been affected. The **Job Diagnostics** dialog box also provides more detailed information to explain the meanings of or possible causes for most warnings and errors and, if the warnings are associated with nodes or elements, to help you locate them on the model in the viewport.

You can view diagnostic information during an analysis or after it ends. However, the **Job Diagnostics** dialog box does not update automatically. If you view diagnostic information while an analysis is running, you must close and then reopen the output database as the analysis progresses to display any new diagnostic information.

Generating diagnostic information

Abaqus generates diagnostic information during an analysis job; Abaqus/CAE reads the diagnostic information that is stored in the output database. Thus, you can use the **Job Diagnostics** dialog box to view convergence information only for procedures that generate information in the output database. Job diagnostics are stored in the output database for the following Abaqus/Standard procedures:

- **Coupled temp-displacement**
- **Geostatic**
- **Soils**
- **Static, General**
- **Static, Linear perturbation**
- **Static, Riks**
- **Visco**

Diagnostic information for other Abaqus/Standard analysis procedures is located in the message file and cannot be viewed in Abaqus/CAE.

Diagnostic information is available in Abaqus/CAE for all Abaqus/Explicit analysis procedures. An anneal procedure does not generate any diagnostic data, so no diagnostics are displayed for an anneal step.

Detailed diagnostic information is saved to the output database by default for the supported analysis procedures in both Abaqus/Standard and Abaqus/Explicit. If you do not want to include diagnostic information, use the **Keywords Editor** to edit the input file to include the following line in the model data:

```
*OUTPUT, DIAGNOSTICS=NO
```

You can also use the **Keywords Editor** to change the default diagnostic output parameters for Abaqus/Explicit by including one or more lines for the keyword **DIAGNOSTICS* and specifying the desired parameter values.

For more information on using the **Keywords Editor**, see *Adding unsupported keywords to your Abaqus/CAE model*.

Interpreting diagnostic information

This section describes the individual pages in the **Job Diagnostics** dialog box (the available pages depend on the analysis type and results).

In a typical Abaqus/Standard analysis a load is applied to the model in increments, and Abaqus attempts to calculate the model's response to each incremental load. Abaqus further reduces the response calculations by performing iterations to approach the result for an increment. If the iterations are not approaching a solution (converging), Abaqus stops and attempts to solve again, this time with a smaller load increment. If Abaqus makes too many attempts without a solution, the analysis is ended. For more information about load increments, see *Analysis Solution and Control*.

In an Abaqus/Explicit analysis, incrementation is based on a large number of small time increments. The time incrementation is made according to an estimated stable time increment size based on the size of the elements in the model. Abaqus/Explicit uses a central-difference time integration rule to integrate the equations of motion, so there is no need for multiple attempts and iterations in each increment.

Viewing the diagnostic information can help you determine the causes of convergence problems so that you can make the necessary corrections in the model. Diagnostic information also indicates potential problems and areas for improvement even when a converged solution is reached. With proper interpretation of the available diagnostic information, you can improve a model to achieve the results that match your analysis intent.

In this section:

- *Diagnostics summary*
- *Incrementation*
- *Warnings and errors*
- *Residuals*
- *Contact*
- *Elements*
- *Other*

Diagnostics summary

The **Summary** page in the **Job Diagnostics** dialog box is always available. The summary includes attributes of the item that is currently highlighted in the **Job History** tree as well as an indication of the diagnostic information available in the other pages of the dialog box. The information that appears for each job, step, increment, attempt, and iteration for an Abaqus/Standard analysis or for each job, step, and increment for an Abaqus/Explicit analysis varies as follows, becoming more specific as you move from the job to an iteration.

Job

When you first open the **Job Diagnostics** dialog box, the job item is highlighted in the **Job History** tree and the **Summary** page is visible. The job name and status and the analysis code and release are displayed. For an Abaqus/Explicit job the Abaqus/Explicit precision (single or double) and the number of domains for parallel job execution are displayed. If there are warnings or errors, the total number of each is also displayed.

Step

The summary for each step displays the step name, step number, analysis procedure, and number of warnings (if any) in the step. Depending on the procedure additional information may also be displayed as part of the step summary. For example, the summary of a general nonlinear step will also include the step time that has been completed, the number of increments that were completed, the time incrementation method (automatic or fixed) that was used, whether nonlinear geometry was accounted for during the step, and the extrapolation type used for a previous state at the start of each increment. See *The Step module* for more information.

Increment

The summary for each increment displays the increment number and the number of warnings (if any) in the increment. For an Abaqus/Standard analysis the number of attempts is also displayed. The increment summary also indicates the convergence status; if the increment converged, the increment size and the completed step time are displayed.

Attempt

The summary for each attempt in an Abaqus/Standard analysis displays the attempt number, attempt size, number of warnings (if any), and number of iterations. Severe discontinuity iterations and equilibrium iterations are listed separately, along with the total number of iterations. If Abaqus is unable to find a solution, it makes a cutback in the increment size and begins a new attempt; if Abaqus makes a cutback, the attempt summary indicates the reason for the cutback. There are no attempts in an Abaqus/Explicit analysis.

Iteration

The iteration summary in an Abaqus/Standard analysis displays the convergence status. If the iteration did not converge, the summary indicates the other pages (**Warnings**, **Residuals**, **Contact**, and **Elements**) that provide detailed information about the convergence criteria that were not satisfied. There are no iterations in an Abaqus/Explicit analysis.

Additional information

- *Viewing diagnostic output*
- *Interpreting diagnostic information*
- *About Convergence and Time Integration Criteria*

Incrementation

The **Incrementation** page in the **Job Diagnostics** dialog box displays the incrementation control settings and the resulting incrementation used by Abaqus during the analysis. The **Incrementation** page is available only if you highlight a step in the **Job History** tree.

The **Status** table contents depend on whether you are viewing diagnostic information for an Abaqus/Standard or Abaqus/Explicit analysis step. For an Abaqus/Standard step the table displays each increment along with the number of attempts, the number of severe discontinuity and equilibrium iterations, the total number of iterations, the step time, and the increment size. For an Abaqus/Explicit step the table displays all summary increments and any additional increments for which diagnostic information was recorded. The critical element, the stable time increment for that element, the elapsed step time, the total time, and the kinetic energy are also listed for each increment. For more information on incrementation diagnostics in Abaqus/Explicit, see *Explicit Dynamic Analysis*.

If you select a column of data and click **Plot selected column**, Abaqus/CAE creates an *X*–*Y* plot using the increment number (first column) for the *X*-axis and the selected column for the *Y*-axis.



Note:

You cannot plot the critical element numbers.

Additional information

- *Viewing diagnostic output*
- *Using the step editor*

Warnings and errors

The **Warnings** and **Errors** pages in the **Job Diagnostics** dialog box both display detailed information about undesirable conditions that were encountered during an analysis. Warnings include information about conditions that may lead to questionable analysis results. Errors include information about conditions that caused Abaqus to terminate the analysis prematurely.

The contents of the **Warnings** page depend on the item that is highlighted in the **Job History** tree; if the job is highlighted, the **Warnings** page displays all warnings saved to the output database for the entire job. If you highlight a step, increment, attempt, or iteration in the **Job History** tree, Abaqus/CAE displays information about only those warnings associated with the highlighted item. The **Errors** page is available only when the job item is highlighted; it displays information about all errors saved to the output database for the entire job.



Note:

Only a subset of the warnings and errors written to the message and status files by Abaqus/Standard and Abaqus/Explicit are saved to the output database for access through the **Job Diagnostics** dialog box.

Each **Warnings** or **Errors** page includes a **Summary** table that provides a concise description of all the problems associated with the item that you highlighted in the **Job History** tree. If there is more than one type of problem, you can filter the list of problems based on a common problem **Category**.

The **Details** region of the **Warnings** and **Errors** pages displays more information corresponding to the item that is highlighted in the **Summary** table. Depending on the type of warning or error that you selected, Abaqus/CAE displays either a statement or a table providing detailed information about the problem. If the **Details** region displays tabular information including nodes and elements, you can toggle **Highlight selections in viewport** and select items to view them in the current viewport.

Figure 1 shows the **Warnings** page for a Abaqus/Standard analysis step; the first warning occurs on the step, and the remaining warnings indicate the increment numbers where they occur. In this case the **Details** section includes the coupling release variation for a numerical singularity.

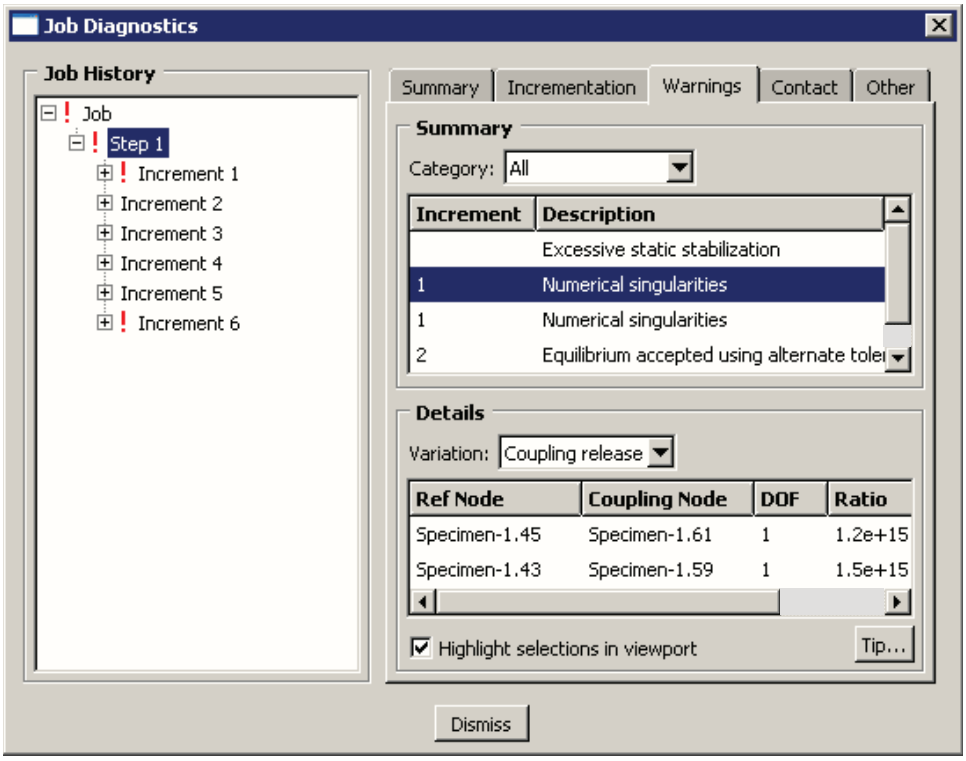


Figure 1: The Warnings page.

Additional information

- *Viewing diagnostic output*
- *Explicit Dynamic Analysis*
- *About Convergence and Time Integration Criteria*
- *Overconstraint Checks*
- *Common Difficulties Associated with Contact Modeling in Abaqus/Standard*

Residuals

The **Residuals** page in the **Job Diagnostics** dialog box displays information about the quantities that Abaqus/Standard uses to determine whether an iteration has produced an equilibrium solution.

Residuals represent the difference between the internal and external forces acting on a model. If the residuals are small, Abaqus accepts the iteration as converged. The tolerances used to determine whether a solution is converged are very important. The tolerances must be small enough to provide an accurate solution but large enough to achieve the solution within a reasonable number of iterations. Before accepting an iteration as converged, Abaqus further requires that corrections to the primary solution variables and constraint equation compatibility errors must also be small.

When the equilibrium iterations do not converge, the node where the maximum residual occurs during the final iterations is usually the best place to begin searching for the problem. There are many conditions that may prevent the equilibrium iterations from converging; diagnosing the source of the problem requires a certain amount of experience.

Residual information, including the maximum residual value in each iteration, is summarized in tabular form for each attempt that contains residual diagnostics. Your selections in the **Equations** and **Variables** fields control the data in the table. You can plot columns from the table by selecting **Plot selected column**. When you have located a problematic iteration, select it in the **Job History** tree. Then you can select items from the **Residuals** page for the iteration and click **Highlight selections in viewport** to view the regions in the current viewport.

Additional information

- *Viewing diagnostic output*
- *General solution controls*
- *Direct Linear Equation Solver*
- *Solving Nonlinear Problems*
- *About Convergence and Time Integration Criteria*

Contact

The **Contact** page in the **Job Diagnostics** dialog box displays information about regions of the model where changes in the contact status prevent Abaqus from accepting the solution for an Abaqus/Standard step. The **Contact** page can also indicate diagnostics such as initial contact overclosures and adjustments for an Abaqus/Explicit step.

Abaqus/Standard contact information is available for any iterations that are followed by `(SDI)` (Severe Discontinuity Iteration) in the **Job History** tree. Contact information may also be available for other iterations where contact impacts the analysis, such as when stick-slip friction behavior is present. Use the contact information to locate regions in your model where Abaqus could not establish the correct conditions. You may need to edit the contact controls to resolve a contact problem. For more information, see *Customizing contact controls*.

Contact information, including the total number of problems in each iteration, is summarized in tabular form for each attempt that contains contact diagnostics. Your selection in the **Description** field controls the data in the table, and you can plot columns from the table by selecting **Plot selected column**. When you have located a problematic iteration, select it in the **Job History** tree. Then you can select items from the **Contact** page for the iteration and click **Highlight selections in viewport** to view the regions in the current viewport. Click the column headings to sort the information in the **Details** field for an iteration.

For an Abaqus/Explicit step the contact summary may include diagnostics such as the number of initial overclosures, unresolved initial overclosures, and initial contact adjustments. Your selection in the **Description** field controls the data in the table. For example, if you select initial overclosures or unresolved initial overclosures, the table lists the overclosed nodes and elements, the element face that the nodes penetrated, and the overclosure amount. You can select items from the **Details** field and click **Highlight selections in viewport** to view the contact nodes and elements in the current viewport.

Additional information

- *Customizing contact controls*
- *Viewing diagnostic output*
- *About Convergence and Time Integration Criteria*
- *Contact Diagnostics in an Abaqus/Standard Analysis*
- *Common Difficulties Associated with Contact Modeling in Abaqus/Standard*

Elements

The **Elements** page in the **Job Diagnostics** dialog box displays information about regions of the model where problems with the element and material point calculations may be preventing Abaqus from finding a converged solution for an Abaqus/Standard step. Abaqus/Explicit displays information for the critical elements—the elements with the lowest stable time limits.



Note:

Only a subset of the element and material point diagnostics written to the message and status files by Abaqus/Standard and Abaqus/Explicit are accessible through the **Job Diagnostics** dialog box.

Select **Highlight selections in viewport** to locate elements in the current viewport. Your selections in the **Element Diagnostics** and **Details** fields determine the elements that Abaqus/CAE will highlight.

Additional information

- *Viewing diagnostic output*
- *About Convergence and Time Integration Criteria*
- *Common Difficulties Associated with Contact Modeling in Abaqus/Standard*

Other

The **Other** page in the **Job Diagnostics** dialog box appears only when you select a **Step** from the **Job History** tree for an Abaqus/Standard analysis. It displays information about the matrix solver used for the analysis and the characteristic element length in the mesh. Depending on the procedure type, other pertinent information such as the mass of the model and the center of mass are also provided. This reference information can help you determine the possible cause or extent of a problem. For example, if the total mass or center of mass are not correct, there is a problem with the material definition, sections, or section assignments in the Property module.

Additional information

- *Viewing diagnostic output*
- *About Convergence and Time Integration Criteria*
- *Common Difficulties Associated with Contact Modeling in Abaqus/Standard*

Accessing diagnostic information

You can use the **Job Diagnostics** dialog box to access diagnostic information that Abaqus writes to the output database during an analysis job.

1. From the main menu bar in the Visualization module, select **Tools->Job Diagnostics**.

   The **Job Diagnostics** dialog box appears.

2. Click the "+" symbols in the **Job History** tree to expand the tree to see the steps, increments, attempts, and iterations in the job history.
3. Click an item in the **Job History** tree to highlight it.

   Abaqus/CAE displays additional tabs on the right side of the dialog box containing diagnostic information associated with the selected item.

4. Click the tabs on the right side of the dialog box. Abaqus/CAE displays various types of diagnostic information corresponding to the item that is selected in the tree. The following sections describe the information available in each tabbed page:

   - *Diagnostics summary*
   - *Warnings and errors*
   - *Residuals*
   - *Contact*
   - *Elements*
   - *Other*

5. Click **Dismiss** to close the **Job Diagnostics** dialog box.

Selecting model data and analysis results to plot

Abaqus/CAE obtains model data from the current model database and model data and analysis results from the output database. Results need not be available to produce an undeformed plot; in this case output database information from a **datacheck** run is sufficient.

This chapter explains how to select model data and analysis results for display.

In this section:

- *Selecting results from an output database*
- *Selecting results from the current model database*
- *Selecting the results step and frame*
- *Customizing the display of steps and frames in the results*
- *Selecting the field output to display*
- *Selecting result options*
- *Creating and saving new field output*
- *Creating coordinate systems during postprocessing*
- *Creating node and element sets during postprocessing*

Selecting results from an output database

Abaqus/CAE reads analysis results from the output database.
Output database results consist of those you have saved during the analysis as field and history output variables. If, for example, you have written data to the field output portion of the output database after every 10 increments, you can now select results at 10 increment intervals only. These increment intervals are called frames.

In addition to the analysis results you have saved, you can operate on output database field output to create new results. You can display the values of both output database field output variables and field output variables that you have created in the form of a deformed, contour, symbol, or *X–Y* plot; as *X–Y* data obtained along a path through your model; by probing any model or *X–Y* plot; or in a tabular report. Furthermore, you can display a subset of your model based on field output values, and you can color code a portion of your model according to these values.

You can display output database history output values in the form of an *X–Y* plot.

Use the **Result** menu from the main menu bar to access the options affecting results; the following menu items are available:

- **Step/Frame**: Control the step and frame at which Abaqus/CAE obtains results.
- **Active Steps/Frames**: Control the subset of steps and frames that Abaqus/CAE displays when stepping through results frame by frame, in animations, and in *X–Y* plots.
- **Section Points**: Control which section points provide results for integration point variables.
- **Field Output**: The **Field Output** dialog box contains the following tabs:
  - **Primary Variable**: Control the variable and, if applicable, the invariant or component for which Abaqus/CAE displays results.
  - **Deformed Variable**: Control the variable Abaqus/CAE uses to display the deformed model shape.
  - **Symbol Variable**: Control the vector or tensor variable and optionally the component Abaqus/CAE uses for symbol plots.
  - **Status Variable**: Control the removal of elements that meet the specified result-based failure criteria from model plots.
- **History Output**: Select history output for *X–Y* plotting.
- **Options**: The **Result Options** dialog box contains the following tabs:
  - **Computation**: Display field output or discontinuities, control the averaging of element-based field output results, and control the computation of results at region boundaries.
  - **Transformation**: Apply coordinate system transformations to field output results.
  - **Complex Form**: Control the numeric form in which Abaqus/CAE displays complex number results.
  - **Caching**: Control whether analysis results are cached in memory during postprocessing and how often the output database should be checked for updated results.

Additional information

- *Opening a model database or an output database*
- *Selecting the results step and frame*
- *Customizing the display of steps and frames in the results*
- *Selecting the field output to display*
- *Selecting result options*
- *Creating and saving new field output*

Selecting results from the current model database

Abaqus/CAE can read selected data from any of the models in the current model database.
When you select a model in the Visualization module, Abaqus/CAE makes a subset of the loads, predefined fields, boundary conditions, and interactions specified in that model available for selection as field output variables. Once selected, you can plot contours or symbols for the selected item in a particular step of your analysis. Abaqus/CAE creates a single dummy frame for each step in the model database.

Only a subset of the loads, predefined fields, boundary conditions, and interactions you can define in a model are available for visualization; and the attributes in this section can be displayed when their propagation status is **Created in this step**, **Propagated from a previous step**, or **Modified in this step**. The following attributes can be displayed in contour plots or symbol plots in the Visualization module:

Supported loads

Unless stated otherwise, the loads below are supported for display for all distributions except user-defined distributions. Abaqus/CAE does not consider any amplitude data associated with the load definition when you display it in the Visualization module. Abaqus/CAE displays **(L)** before the name of each load that is available for display in the current step.

- Concentrated force
- Moment
- Concentrated charge
- Concentrated heat flux
- Surface charge
- Concentrated concentration flux
- Surface heat flux (for all distributions except total flux)
- Surface concentration flux
- Pressure load (for all distributions except total force)

If you defined a load using a user-specified coordinate system, the modified orientation is reflected in the Visualization module as well. Abaqus/CAE also customizes the display of any load that is defined using an expression field as its custom distribution; display of loads with mapped fields is not supported.

Predefined fields

Abaqus/CAE displays **(P)** before the name of each predefined field that is available for display in the current step.

- Temperature (all distributions except **From results or output database file** and only the **Constant through section** variation.)
- Velocity (translational velocity components only)

If you defined a predefined field using an expression field or a mapped field, the custom distribution is reflected in the Visualization module as well.

Boundary conditions

Abaqus/CAE displays **(B)** before the name of each boundary condition that is available for display in the current step. For boundary conditions that include degrees of freedom, such as displacement/rotation, you can select the

individual degree of freedom you want to display after selecting the boundary condition. The individual components available for display are indicated below in parentheses where applicable.

You can display data for any boundary condition with parameters that support distributions:

- Displacement/rotation (translational and rotational displacements)
- Velocity/angular velocity (translational and angular velocity)
- Acceleration/angular acceleration (translational and angular acceleration)
- Fluid inlet/outlet
- Fluid wall
- Temperature
- Pore pressure
- Electric potential
- Normalized concentration
- Acoustic pressure
- Connector material flow

Interactions

Abaqus/CAE displays **(I)** before the name of each interaction that is available for display in the current step. Surface film condition (film coefficient and sink temperature) is the only supported interaction.

Additional information

- *Opening a model database or an output database*
- *Selecting the results step and frame*
- *Customizing the display of steps and frames in the results*
- *Selecting the field output to display*
- *Selecting result options*
- *Creating and saving new field output*

Selecting the results step and frame

You can control the step and frame at which Abaqus/CAE obtains model data and results from an output database.

You can select a specific step and frame using the **Step/Frame** dialog box, or you can step through frames using controls available in the context bar. If viewport linking is activated, you can link frame display across viewports. For more information, see *Linking viewports*.



Note: If you are displaying model data from the current model database in the Visualization module, you can select a specific step using the **Step/Frame** dialog box, but the frame controls are disabled. Model database data does not include individual frames.

In this section:

- *Selecting a specific results step and frame*
- *Stepping through frames*

Selecting a specific results step and frame

You can select a specific step and frame at which Abaqus obtains model data and results from an output database, and you can select a specific step at which Abaqus obtains model data from a model in the current model database.

In an output database, available steps and frames consist of those for which analysis results have been saved; in a model database, all steps are available for data display. To view field output variables that you have created during the Abaqus/CAE session, if any, you must select **Result->Step/Frame->Session Step**. To view parametric shape variations defined in your analysis, if any, you must select **Result->Step/Frame->Step 0**.

It is possible to open an output database for an analysis that is still in progress. As the analysis moves toward completion, the lists of completed steps and frames are updated every time you close and then reopen the **Step/Frame** dialog box.

For information on stepping through results frames, see *Stepping through frames*.

1. Locate the **Step/Frame** options.

   From the main menu bar, select **Result->Step/Frame**. The **Step/Frame** dialog box appears.

   The **Step** portion of the dialog lists the available steps, and the **Frame** portion of the dialog lists the available frames for the current *output database* if an output database is selected.

   Note:

   You can also access these options by clicking  in the **Field Output** dialog box.

2. From the **Step** list, click the step to display.

   The selected step is highlighted, and Abaqus/CAE refreshes the **Frame** list to show only frames available for the selected step.

3. From the **Frame** list, click the frame to display. Select **Frame 0** to display the base state of the current step; for example, to contour initial stresses.

   The selected frame is highlighted.

4. The model plot in the current viewport changes to show your model at the step and frame you have selected. If active, the text in the state block changes to identify the selected step and frame. For more information on the state block, see *Customizing the state block*.

   Abaqus refreshes the ***Field Output*** dialog box to list variables available for the selected frame. (You can access the **Field Output** dialog box by clicking the **Field Output** button on the bottom of the **Step/Frame** dialog box.) Abaqus also refreshes all dialog boxes in which the current step and frame are identified.

   Your changes are saved for the duration of the session.

Additional information

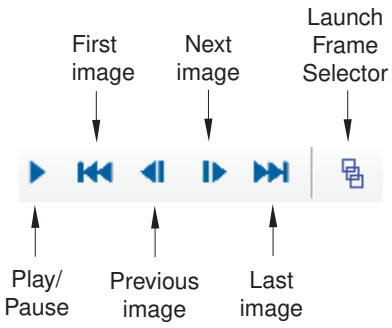
- *Selecting model data and analysis results to plot*
- *Customizing the display of steps and frames in the results*
- *Creating field output by operating on fields*

Stepping through frames

You can use the **ODB Frame** control buttons available in the context bar to step through results frames. Any frames for which analysis results have been saved and that are set to active are available. For information on selecting a specific results frame, see *Selecting a specific results step and frame*. For information on activating results steps and frames, see *Activating and deactivating steps and frames*.

Stepping through frame data is available for output databases only.

1. Locate the **ODB Frame** buttons. These buttons appear on the right side of the context bar when you display an undeformed, deformed, contour, or symbol plot.



2. Click one of the following frame buttons:
   - **First** to display results from the first active frame of the current step. This button has no effect if you are already displaying results from the first frame of the step.
   - **Previous** to display results from the previous active frame. If you are currently displaying results from the first frame of a step, Abaqus/CAE will display results from the last active frame of the previous step. This button has no effect if you are already displaying results from the first frame of the first step.
   - **Next** to display results from the next frame. If you are currently displaying results from the last active frame of a step, Abaqus/CAE will display results from the first frame of the next step. This button has no effect if you are already displaying results from the last frame of the last step.
   - **Last** to display results from the last frame of the current step. This button has no effect if you are already displaying results from the last frame of the step.
   - **Frame Selector** to launch the Frame Selector dialog box, which enables you to navigate directly to a particular frame by entering its frame number or by dragging a slider to the frame you want to display. For more information, see *Navigating to a specific frame in the animation*.

   The model plot in the current viewport changes to show your model at the step and frame you have selected. If active, the text in the *state block* changes to identify the selected step and frame. Abaqus refreshes the **Step/Frame** dialog box, highlighting the selected step and frame and the **Field Output** dialog box, listing variables available for the frame you have selected. Abaqus also refreshes all dialog boxes in which the current step and frame are identified.
3. Continue clicking frame buttons to step through available frames.

Additional information

- *Selecting model data and analysis results to plot*
- *Customizing the display of steps and frames in the results*
- *Customizing the state block*

Customizing the display of steps and frames in the results

You can customize your display of an output database's steps and frames by activating only a subset of these steps and frames or by changing the duration or arc length of one or more steps.
Abaqus/CAE displays only active steps and frames when you examine output data by stepping through data frame by frame, when you animate the data, and when you generate *X–Y* history data from field data.

In this section:

- *Activating and deactivating steps and frames*
- *Changing the period of a step*

Activating and deactivating steps and frames

You can customize the display of analysis results by activating a subset of the steps and frames in the output database. Abaqus/CAE displays only active steps and frames when you examine analysis results by stepping through frames, animating the data, or creating *X–Y* history data from field data. The subset of active steps and frames persists throughout your session and applies to every display of a model database's results.

You can activate and deactivate steps or frames either from the Results Tree or from the **Active Steps/Frames** dialog box. Both tools also enable you to expand and collapse steps by clicking on the "plus" and "minus" signs to the left of each step name. Expanding a step reveals its frames, enabling you to activate some of the frames rather than the entire step.

The Results Tree and **Active Steps/Frames** dialog box also provide visual cues to indicate the active status of steps or frames in the output database. The Results Tree flags inactive frames with a red "X" that appears to the left of the frame shortcut; frames not displayed in this manner are currently active. In the **Active Steps/Frames** dialog box, step and frame status is indicated by check marks that appear in the area between the plus/minus sign and the step's name. A green check mark display for a step means that all of its frames are active; a green check mark for an individual frame means that the frame is active; a dark grey check mark on a light grey background means that some of the step's frames are active; and when no check mark appears, the step is completely inactive. This visual cue can help you assess a step's active status when the step container is collapsed. *Figure 1* shows the **Active Steps/Frames** dialog box for an output database with two steps completely active, one step partially active, and two steps inactive.

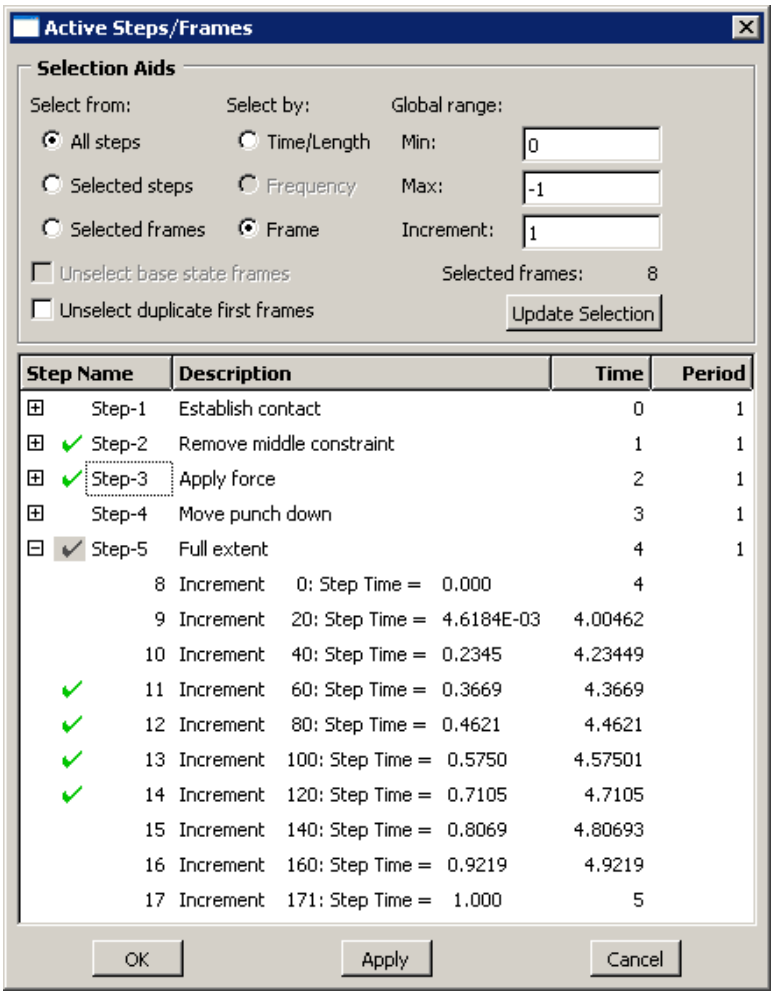


Figure 1: Active Steps/Frames dialog box.

Each row in the **Active Steps/Frames** dialog box also provides the following information about the step:

- The **Step Name** and **Description** columns reflect the values that were defined for this step when the model was created.
- The value in the **Time** column either provides the step's starting time or describes the nature of the step. For a time-based step, this column displays the starting time for the step. If the step is not time-based, this column describes whether the step is a Modal, Frequency, or Arc (Riks analysis) step.
- The value in the **Period** column also depends on the step type. For a time-based step, the period is the step's total duration. For a Riks analysis step, the period is the step's total arc length. Modal and frequency steps have a dash (−) displayed in this column to indicate that a period is not applicable for this step.

  You can modify the period value for time-based and Riks analysis steps, provided the period does not equal 0. See *Changing the period of a step*, for more information.

The **Active Steps/Frames** dialog box provides two methods of activating steps and frames: you can use the filtering tools in the **Selection Aids** portion of the dialog box to activate a set of steps or frames, and you can activate individual step and frame rows by clicking their rows in the lower portion of the dialog box. Activating steps or frames using the selection aids clears the subset of steps and frames that you currently have active. If you plan to select steps and frames manually, do so after you choose a subset using the selection aids.

Selection Aids

The **Selection Aids** filtering options enable you to choose a subset of steps and frames according to several selection criteria.

- The **Select from** options define the candidate set of steps and frames for your search. You can search across all steps in the output database, or search only within the subset of steps or frames that are currently selected in the lower portion of the **Active Steps/Frames** dialog box.
- The **Select by** options specify the variable for which you want to search for matching steps and frames. You can search for any steps or frames that occurred within a particular range of time. For frequency extraction steps, you can search for any steps or frames during which the model exhibited a particular frequency. Alternatively, you can search for frames by their frame number.
- The **Global range** fields enable you to specify the upper and lower bounds for your search and to define the increment between matching steps or frames. For example, if you want to activate all even-numbered frames between a lower bound of frame 0 and an upper bound of frame 20, enter 2 as the increment.
- You can also deactivate base state frames in the output database by selecting **Unselect base state frames**. This option is available only when the output database contains linear perturbation steps.
- Some steps may have initial frames that duplicate the final frame of the previous step. Unselecting these duplicate first frames can make your data analysis smoother and more realistic. Choose **Unselect duplicate first frames** to unselect all duplicate first frames throughout all steps in the database.

Selecting steps and frames manually

You can activate and deactivate individual steps and frames by clicking rows in the lower portion of the **Active Steps/Frames** dialog box. Clicking a single step activates or deactivates all of its frames, while clicking a frame activates or deactivates that frame only. If you click and drag across several steps or frames, Abaqus/CAE inverts the activation status of all selected steps of frames you select.



Tip: You can also activate or deactivate steps and frames from the Results Tree. Highlight the steps or frames that you want to toggle, click mouse button 3, then select **Activate all** or **Deactivate all** to change the active status of all frames within the selected steps or select **Activate** or **Deactivate** to change the active status of the selected frames.

The Results Tree indicates inactive frames with a red “X” that appears to the left of the frame shortcut. User-defined session steps, which cannot be deactivated, are indicated with a red exclamation point; these steps are not displayed in the **Active Steps/Frames** dialog box.

Manually activating and deactivating steps and frames in the dialog box can supplement and refine the choices you make by using the selection aids. In addition, clicking mouse button 3 in the lower portion of the dialog box provides access to the following manual selection shortcuts:

- **Expand All** expands all of the steps containers, selected and unselected, to reveal the frames of every step in the output database.
- **Collapse All** collapses all of the selected and unselected step containers in the output database.
- **Select All** selects every frame of every step in the output database.
- **Deselect All** deselects every frame of every step in the output database.
- **Reset Selection** reverts the active steps and frames to the subset that was active when you last clicked **Apply**.
- **Reset Periods** reverts the period values to the settings that were active when you last clicked **Apply**.

1. Locate the **Active Steps/Frames** dialog box.

   From the main menu bar, select **Result->Active Steps/Frames**. The **Active Steps/Frames** dialog box appears.

   The lower portion of the dialog box shows the steps and, for expanded steps, the frames in the output database. The first time you examine an output database’s steps and frames during a session, all of the steps and frames are active.

2. From the **Selection Aids** portion of the dialog box, choose the data set, variable, and global range values for your search.
3. Click **Update Selection**.

   Abaqus/CAE updates the subset of active steps and frames.

4. Activate or deactivate steps manually in the lower portion of the dialog box. You can toggle the activate status of a single step or frame by clicking the appropriate row, or you can toggle the activation status of multiple rows by clicking and dragging several rows.
5. Click **Apply**.

   Abaqus/CAE updates all viewports that display this output database with the newly selected subset of active steps and frames.

Additional information

• *Selecting the results step and frame*
• *Changing the period of a step*
• *Reading X–Y data from output database field output*
• *Animating plots*

Changing the period of a step

The meaning of a step's period value depends on whether the step is a Riks analysis step or a time-based step. For Riks analysis steps, the period specifies the total arc length for the step. For time-based steps, the period specifies the step's total duration, so changing this value affects how long this step is displayed and changes the starting times for all subsequent steps. Changing the time period of a single step also changes the duration of the entire animation or frame-by-frame display, affects the auto-computed start and end time of time-based animations, and affects the way the animation of this model synchronizes with other models. You might want to adjust the time period if you want to synchronize events in different models that occur at different times in the common time line.

Changes to the period values last for the duration of your session. Abaqus/CAE does not save these changes to the output database.

You cannot edit the period value of a Riks analysis or time-based step if the value equals 0. In addition, modal and frequency extraction steps do not have period values; Abaqus/CAE displays an uneditable dash (−) in the **Period** column for steps of these types.



Tip: If you want to revert the period values to the output database's settings, click mouse button 3 in the lower portion of the window and select **Reset Periods**.

1. Locate the **Active Steps/Frames** dialog box.

   From the main menu bar, select **Result->Active Steps/Frames**. The **Active Steps/Frames** dialog box appears.

   The lower portion of the dialog box shows the steps and, for expanded steps, the frames in the output database. The first time you examine an output database's steps and frames during a session, all of the steps and frames are active.

2. From the lower portion of this dialog box, click the value in **Period** column for a step, then enter a new value in the field. Repeat this change for every step period that you want to customize.
3. Click **Apply**.

   Abaqus/CAE updates all viewports that display this output database with the revised period and time values.

Additional information

- *Selecting the results step and frame*
- *Reading X–Y data from output database field output*
- *Animating plots*

Selecting the field output to display

This section explains how to select field output variables to display.

To learn how to select output database history output to produce an *X–Y* plot, see *Reading X–Y data from output database history output*. To learn how to select output database field output to produce an *X–Y* plot, see *Reading X–Y data from output database field output*.

In this section:

- *Selecting field output variables*
- *Using the field output toolbar*
- *Selecting the primary field output variable*
- *Selecting the deformed field output variable*
- *Selecting the symbol field output variable*
- *Selecting the status field output variable*
- *Selecting the stream field output variable*
- *Selecting complex results*
- *Selecting section point data*
- *Selecting contact output*

Selecting field output variables

Contour plots, model probing, view cuts based on an isosurface, and *X*–*Y* plots of results along a model path all show the values of a particular field output variable at a specified step and frame of your analysis.
Similarly, when you form a display group or specify color coding based on results, or when you display a load or predefined field from data in the current model database, these results pertain to a particular field output variable. The variable whose values are shown is called the *primary field output variable*.

Deformed plots show the shape of your model based on the values of a nodal variable (such as displacement) at a specified step and frame of your analysis. The variable whose values are shown is called the *deformed field output variable*.

Symbol plots show the magnitude and directions of a particular vector or tensor variable at a specified step and frame of the analysis. A symbol plot can also the magnitude and direction of a selected load or predefined field in a model from the current model database. The variable whose values are shown is called the *symbol field output variable*.

You can specify result-based criteria for element failure for a selected field output variable and remove elements that meet the failure criteria from model plots. The variable for which you define the failure criteria is called the *status field output variable*.

You can specify velocity or vorticity data for stream display. The variable for which you display these data is called the *stream field output variable*.

You can choose to display contour and symbol plots on either the undeformed or deformed model shape. When you use the deformed model shape, the contours or symbols represent the values of the primary field output variable or symbol field output variable, respectively, while the shape of the underlying model is determined by the values of the deformed field output variable.

1. Locate the **Field Output** options.

   From the main menu bar, select**Result->Field Output**. The **Field Output** dialog box appears.

   Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears or by clicking  in the **Field Output** toolbar.

2. Select the primary, deformed, symbol, and status field output variables that you want as described in the following sections:

   - *Selecting the primary field output variable*
   - *Selecting the deformed field output variable*
   - *Selecting the symbol field output variable*
   - *Selecting the status field output variable*
   - *Selecting the stream field output variable*
   - *Selecting complex results*
   - *Selecting section point data*
   - *Selecting contact output*

   The model plot in the current viewport changes to show the variables you have selected. If active, the text in the legend and state block changes to identify the variables associated with the plot. For more

information on the legend and state block, see *Customizing the legend*, and *Customizing the state block*. In addition, Abaqus refreshes all dialog boxes in which the currently selected variables are identified.

Additional information

- *Selecting model data and analysis results to plot*
- *Using the field output toolbar*
- *Creating or editing a view cut*

Using the field output toolbar

You can use the **Field Output** toolbar to access the basic functionality of the **Field Output** dialog box. From the toolbar, you can

- choose the type of field output variables to manipulate (***Primary***, ***Deformed***, or ***Symbol***);
- choose the variable name from a list of the available field output variables;
- choose the refinement level, such as invariants and components for the selected primary variable, if available; and
- choose whether the viewport plot state should be synchronized with the toolbar selections.

The **Status** and **Stream** variable types are the only field output that you cannot select from the toolbar. The **Status** and **Stream** variable types are available in the **Field Output** dialog box; the status variable allows you to specify criteria that Abaqus/CAE uses to remove failed elements from the model display, and the stream variable determines the field output displayed in streamlines for an analysis of fluid flow data. To open the **Field Output** dialog box, click , located on the left side of the **Field Output** toolbar.



Figure 1: The Field Output toolbar.

As you make selections from the toolbar, Abaqus/CAE updates the current viewport to display the output; the viewport plot state is also updated if a change in plot state is needed and if the plot state synchronization toggle () is enabled. For example, selecting **Primary** as the variable type changes the plot state to display contours on the deformed model if the viewport does not already contain a contour plot. If you disable synchronization, Abaqus/CAE still displays your newly selected field output variable in the current viewport but does not change the plot state.

Additional information

- *Selecting model data and analysis results to plot*
- *Selecting field output variables*

Selecting the primary field output variable

You can select the variable to display for *contour plots*, *model probing*, view cuts based on an isosurface, and for which to *obtain results along a model path*; this variable is called the primary field output variable. Abaqus/CAE also uses the primary field output variable to *form display groups*, to *apply color coding* based on results, and for display of loads, predefined fields, or interactions from a model in the current model database.

When an output database is selected, Abaqus/CAE lists for your selection all variables available at the current *step and frame* of your *output database* by default. An asterisk to the left of the description indicates that the variable includes complex number results.

When a model in the current model database is selected, Abaqus/CAE lists for your selection all loads, predefined fields, boundary conditions, and interactions available in the current step of your model by default. All of these selectable items are preceded by a letter in parentheses to distinguish them by category: **(L)** for loads, **(P)** for predefined fields, **(B)** for boundary conditions, and **(I)** for interactions. Only the real part of a complex load or predefined field is available for display.

Use the **Primary Variable** options in the **Field Output** dialog box to choose the variable and, if applicable, the invariant or component that you want. For information on individual output variable identifiers, see *Output Variables*.

1. Locate the options that control the primary field output variable.

   From the main menu bar, select**Result->Field Output**. Click the **Primary Variable** tab in the dialog box that appears.

   The **Primary Variable** options appear.

   Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears.

   To see the complete descriptions of the variables listed, increase the width of the dialog box by dragging one corner.

2. To control which variables appear in the **Name** and **Description** list:
   a. Toggle **List only variables with results** to display a list that is limited by the storage location of the variables. Limiting the list helps you select variables by presenting, for example, only integration point quantities.

      When **List only variables with results** is on, filter options become available in the pull-down menu.

   b. Click the **List only variables with results** arrow to reveal the filter options.
   c. Click the text stating the location of the variables you want to include in the **Name** and **Description** list.

      The text appears in the **List only variables with results** box, and the **Name** and **Description** list is refreshed to include only variables having that location.

3. From the **Name** and **Description** list, click the name of the analysis variable that you want. An asterisk to the left of the description in the list indicates that the variable includes complex number results.

   The selected variable is highlighted. If applicable, the **Component** and **Invariant** lists on the bottom of the dialog box are refreshed to display available components or invariants, respectively.

4. If items are listed in the **Component** or **Invariant** list, click the component or invariant that you want.

   The selected component or invariant is highlighted.

   Note: For S and E field output, there are two invariants, **Max. Principal (abs)** and **Max. In-Plane Principal (abs)**, which are available only in the Visualization module. **Max. Principal**

(abs) is the largest principal value when the absolute value of all principal values are compared. **Max. In-Plane Principal (abs)** is the largest principal value when the absolute value of all in-plane principal values are compared. The out-of-plane principal value is not considered when the **Max. In-Plane Principal (abs)** value is computed.

5. When active, a contour plot in the current viewport changes to show values for the analysis variable you have specified. If active, the text in the legend and state block changes to identify the variable associated with the plot. For more information on the legend and state block, see *Customizing the legend*, and *Customizing the state block*. In addition, Abaqus refreshes all dialog boxes in which the current primary variable is identified.

   Your changes are saved for the duration of the session.

Additional information

• *Selecting model data and analysis results to plot*

Selecting the deformed field output variable

You can display the deformed model by producing a deformed plot or by selecting the deformed model as the underlying shape of a contour or symbol plot. The shape of your deformed model is based on the values of the particular deformed field output variable that you select. An asterisk to the left of the description in the **Output Variable** list indicates that the variable includes complex number results. For information on individual output variable identifiers, see *Output Variables*.

1. Locate the options that control the deformed field output variable.

   From the main menu bar, select **Result->Field Output**. Click the **Deformed Variable** tab in the dialog box that appears.

   The **Deformed Variable** options become available.

   Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears.

   Abaqus/CAE lists by name and description all variables available at the current *step and frame* of your *output database* that can be used for a deformed plot (nodal vector quantities). To see the complete descriptions of the variables listed, increase the width of the dialog box by dragging one corner.

2. From the **Name** and **Description** list, click the deformed field variable that you want. An asterisk to the left of the description in the list indicates that the variable includes complex number results.

   The selected variable is highlighted.

3. The deformed model shape in the current viewport changes to reflect the values of the deformed field output variable you have specified. If active, the text in the state block changes to identify the variable associated with the plot. For more information on the state block, see *Customizing the state block*.

   Your changes are saved for the duration of the session.

Additional information

- *Selecting model data and analysis results to plot*

Selecting the symbol field output variable

You can select the vector or tensor variable component and optionally select a component to display for a symbol plot; this variable is called the symbol field output variable.

When an output database is selected, Abaqus/CAE lists for your selection all vector and tensor variables available at the current *step and frame* of your *output database* by default; resultant values are displayed in vector variable symbol plots, and all principal components are displayed in tensor variable symbol plots. An asterisk to the left of the description indicates that the variable includes complex number results.

When a model from the current model database is selected, Abaqus/CAE lists for your selection all loads, predefined fields, boundary conditions, and interactions available at the current step of your model by default. All of these selectable items are preceded by a letter in parentheses to distinguish them by category: **(L)** for loads, **(P)** for predefined fields, **(B)** for boundary conditions, and **(I)** for interactions.

Use the **Symbol Variable** options in the **Field Output** dialog box to choose the variable and the specific components that you want. For information on individual output variable identifiers, see *Output Variables*.

When you display a vector variable in a symbol plot, you can choose from the following options:

- Display arrows that represent resultant values of the variable. For example, a symbol plot of total displacement appears on the left side of *Figure 1*.
- Display arrows that represent specific component values of the variable. For example, a symbol plot of displacement in the 1-direction appears on the right side of *Figure 1*.

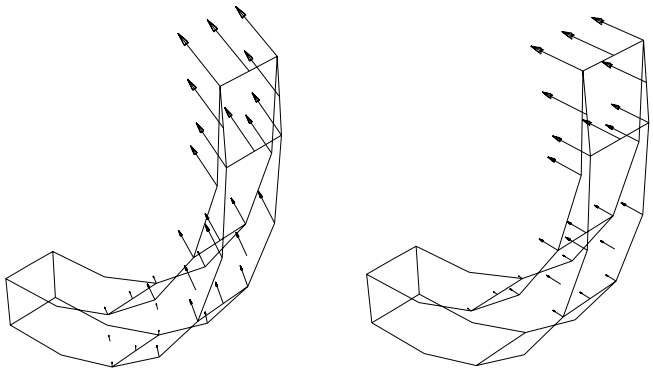


Figure 1: Symbol plots showing total displacement (left) and displacement in the 1-direction (right).

Likewise, when you display a tensor variable in a symbol plot, you can choose from the following options:

- Display arrows that represent each of the principal components of the variable. For example, a symbol plot of maximum, intermediate (mid), and minimum principal stress appears on the left side of *Figure 2*.
- Display arrows that represent a specific principal component of the variable. For example, a symbol plot of only maximum principal stress appears on the right side of *Figure 2*.

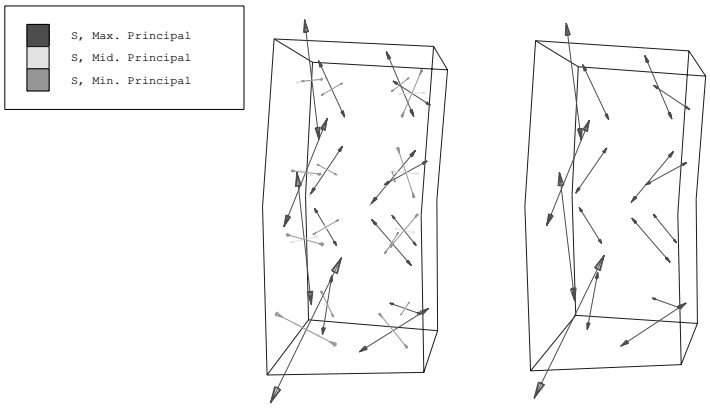


Figure 2: Symbol plot showing all three principal components (left), and symbol plot showing only maximum principal stress (right).

- Display arrows that represent each of the direct components of the variable. For example, a symbol plot of all direct components appears on the left side of *Figure 3*.
- Display arrows that represent a specific direct component of the variable. For example, a symbol plot of only S22 appears on the right side of *Figure 3*.

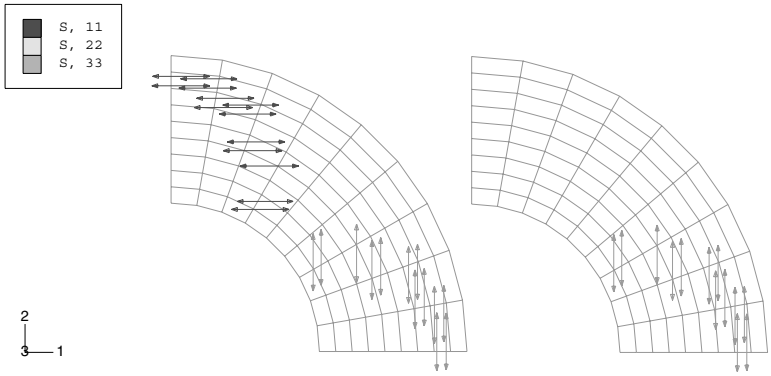


Figure 3: Symbol plot showing all direct components (left), and symbol plot showing only S22 (right).

1. Locate the options that control the symbol field output variable.

   From the main menu bar, select**Result->Field Output**. Click the **Symbol Variable** tab in the dialog box that appears.

   The **Symbol Variable** options appear.

   Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears.

   To see the complete descriptions of the variables listed, increase the width of the dialog box by dragging one corner.

2. To control which variables appear in the **Name** and **Description** list:

   a. Toggle **List only variables with results** to display a list that is limited by the storage location of the variables. Limiting the list helps you select variables by presenting, for example, only integration point quantities.

When **List only variables with results** is on, filter options become available in the pull-down menu.

b. Click the **List only variables with results** arrow to reveal the filter options.

c. Click the text stating the location of the variables you want to include in the **Name** and **Description** list.

The text appears in the **List only variables with results** box, and the **Name** and **Description** list is refreshed to include only variables having that location.

3. From the **Name** and **Description** list, click the name of the analysis variable that you want. An asterisk to the left of the description in the list indicates that the variable includes complex number results.

The selected variable is highlighted. The **Vector Quantity**, **Tensor Quantity**, and **Component** lists on the bottom of the dialog box are refreshed to display available vector quantities, tensor quantities, and components, respectively.

4. If you are creating a vector variable symbol plot, select the components to plot.

- Select **Resultant** to display arrows that represent the resultant of the variable.
- Select **Selected Component** and the component that you want to display arrows that represent a particular component of the variable.

5. If you are creating a tensor variable symbol plot, select the component to plot.

- Select **All Principal Components** to display arrows that represent all available principal components.
- Select **Selected Principal Component** and the component that you want to display only arrows that represent a particular principal component.
- Select **All Direct Components** to display arrows that represent all three direct components.
- Select **Selected Direct Component** and the component that you want to display only arrows that represent a particular direct component.



Note: For S and E field output, there are two invariants, **Max. Principal (abs)** and **Max. In-Plane Principal (abs)**, which are available only in the Visualization module. **Max. Principal (abs)** is the largest principal value when the absolute value of all principal values are compared. **Max. In-Plane Principal (abs)** is the largest principal value when the absolute value of all in-plane principal values are compared. The out-of-plane principal value is not considered when the **Max. In-Plane Principal (abs)** value is computed.

6. Click **Apply** to implement your changes.

The symbol plot in the current viewport changes to show values for the analysis variable you have specified. If active, the text in the legend and state block changes to identify the variable associated with the plot. For more information on the legend and state block, see *Customizing the legend*, and *Customizing the state block*.

Your changes are saved for the duration of the session.

Additional information

• *Selecting model data and analysis results to plot*

Selecting the status field output variable

You can specify result-based criteria for element failure for a selected field output variable; this variable is called the status field output variable. When you enable the use of the status field output variable, elements that meet the specified criteria are considered failed and they are removed from model plots. Use the **Status Variable** options in the **Field Output** dialog box to enable the use of the status variable; to select the variable and, if applicable, an invariant or components; and to specify the result-based criteria for failure. If desired, you can apply the status field output variable to undeformed shape plots. By default, failed elements are not removed when the model is displayed in an undeformed state.

By default, the status field output variable is not enabled unless the output variable STATUS appears in the output database. In this case the use of the status field output variable is enabled by default, the output variable STATUS is selected, and the failure criteria is set to remove elements with values less than or equal to 0.5 (the status of an element is 1.0 if the element is active, 0.0 if the element is not). Elements with a value of 0.0 are not displayed in the model plot in the current viewport.

When selecting a status field output variable, Abaqus/CAE lists for your selection all of the variables (except surface-based field output variables) available at the current *step and frame* of your *output database*. An asterisk to the left of the description indicates that the variable includes complex number results. For information on individual output variable identifiers, see *Output Variables*.

1. Locate the options that control the status field output variable.
   a. From the main menu bar, select**Result**->**Field Output**. Click the **Status Variable** tab in the dialog box that appears.
   b. Toggle **Use status variable** to disable or enable use of the status variable.

      When **Use status variable** is on, the status variable is enabled and the **Status Variable** options become available.

      Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears.

   To see the complete descriptions of the variables listed, increase the width of the dialog box by dragging one corner.
2. Toggle on **Apply to undeformed state** to apply the status field output variable to undeformed shape plots. If your model contains no deformations, you must toggle on this setting to apply the status field output variable.
3. To control which variables appear in the **Name** and **Description** list:
   a. Toggle **List only variables with results** to display a list that is limited by the storage location of the variables. Limiting the list helps you select variables by presenting, for example, only integration point quantities.

      When **List only variables with results** is on, filter options become available in the pull-down menu.
   b. Click the **List only variables with results** arrow to reveal the filter options.
   c. Click the text stating the location of the variables you want to include in the **Name** and **Description** list.

      The text appears in the **List only variables with results** box, and the **Name** and **Description** list is refreshed to include only variables having that location.
4. From the **Name** and **Description** list, click the name of the analysis variable that you want. An asterisk to the left of the description in the list indicates that the variable includes complex number results.

The selected variable is highlighted. If applicable, the **Invariant** and **Component** lists on the bottom of the dialog box are refreshed to display available invariants or components, respectively.

5. If items are listed in the **Invariant** or **Component** list, click the invariant or component that you want.

   The selected invariant or component is highlighted.

   Note: For S and E field output, there are two invariants, **Max. Principal (abs)** and **Max. In-Plane Principal (abs)**, which are available only in the Visualization module. **Max. Principal (abs)** is the largest principal value when the absolute value of all principal values are compared. **Max. In-Plane Principal (abs)** is the largest principal value when the absolute value of all in-plane principal values are compared. The out-of-plane principal value is not considered when the **Max. In-Plane Principal (abs)** value is computed.

6. Choose from the filtering methods in the **Remove elements** field to specify the failure criteria. If any result value of an element meets the specified criteria, that element is considered failed and it is removed from model plots. Element output quantities are compared against the specified criteria; there is no extrapolation or averaging performed on the element data. If a nodal quantity is selected and a value for a given node meets the specified criteria, all elements connected to the failed node are removed.

   - Select  and enter a value in the **Max** field to remove elements with values less than or equal to the specified value.
   - Select  and enter a value in the **Min** field to remove elements with values greater than or equal to the specified value.
   - Select  (inside) and enter values for the upper and lower bounds of the range in the **Min** and **Max** fields to remove elements with values between and including the specified values.
   - Select  (outside) and enter values for the upper and lower bounds of the range in the **Min** and **Max** fields to remove elements with values outside the specified range.

7. Click **Apply** to implement your changes.

   The model plot in the current viewport changes to remove elements with values of the selected status field output variable that meet the specified criteria. If active, the text in the legend and state block changes to identify the variable associated with the plot. For more information on the legend and state block, see *Customizing the legend*, and *Customizing the state block*.

   Your changes are saved for the duration of the session.

Additional information

- *Selecting model data and analysis results to plot*

Selecting the stream field output variable

You can specify the variable for which stream data are computed; this variable is called the stream field output variable. For most analyses velocity (**V**) can be used for the stream data. Only node-centered vector results can be associated with streamline traces.

When selecting a stream field output variable, Abaqus/CAE lists all of the vector variables stored at nodal locations and available at the current *step and frame* of your *output database*.

1. Locate the options that control the stream field output variable.

   From the main menu bar, select **Result->Field Output**. Click the **Stream Variable** tab in the dialog box that appears.

   Tip: You can also access these options by clicking the **Field Output** button in any dialog box in which it appears.

   To see the complete descriptions of the variables listed, increase the width of the dialog box by dragging one corner.

2. From the **Name** and **Description** lists, click the name of the analysis variable that you want.

   The selected variable is highlighted.

3. Click **Apply** to implement your changes.

   Abaqus/CAE recomputes the streamlines in the current viewport based on the selected stream output variable. Your selection for stream output variable is also saved for the duration of the session.

Additional information

• *Selecting model data and analysis results to plot*

Selecting complex results

By default, Abaqus/CAE lists for your selection all variables available at the current *step and frame* of your *output database*. If the current step is a steady-state dynamic analysis, the value of an output variable such as stress (S) or displacement (U) can be a complex number with both real and imaginary components. Use the **Primary Variable**, **Deformed Variable**, **Symbol Variable**, **Status Variable**, and **Stream Variable** options in the **Field Output** dialog box to choose the complex variables that you want to display. The description of each listed variable begins with an asterisk if that variable includes complex results. For information on individual output variable identifiers, see *Output Variables*.

You can display complex results in the form of a *deformed*, *contour*, or *symbol* plot; by *model probing*; by *obtaining results along a model path*; and by forming a display group or specifying color coding based on results. In each of these cases you can choose from several *numeric forms* to display different aspects of the complex results. You can animate complex results using *harmonic animation*. During harmonic animation the real and imaginary components of each complex number are used to calculate and display the value of the result at a sequence of angles representing a half cycle (0° to 180°) or a full cycle (–180° to 180°).

Additional information

- *Selecting model data and analysis results to plot*
- *Producing a harmonic animation*

Selecting section point data

If your model contains shells, beams, or composite layups, you can view data from selected section points when you are displaying a contour, symbol, or material orientation plot.

You can select section points, by category or ply, that will apply across linked viewports. The linking also applies to envelope plots. For more information, see *Linking viewports*.

You can choose either of the following selection methods:

By category

For each section of your model, you first choose the active location of a section point (bottom, top, or both). You then choose from a list of available section points for this category. For example, you can choose to display data from the midpoint of each section. The available section points are related to the section points that you requested when you created the field output request in the Step module. You can also display a plot of the section point containing the critical value (maximum or minimum) of the field output variable across all of the section points. For more information, see *Selecting section point data by category*.

By ply

For a model containing a composite layup, you first choose a ply, and you then choose a location from which to display the field output variable. You can specify one of the following as the section point location in the selected ply: the bottommost section point, the middle or only section point, the topmost section point, or both the topmost and bottommost section points. For more information, see *Selecting section point data by ply*.

In this section:

- *Selecting section point data by category*
- *Selecting section point data by ply*

Selecting section point data by category

When you are displaying a contour plot, you can use the **Section Points** dialog box to control the section points from which Abaqus obtains integration point results and material orientations by selecting a category and the section points that represent the bottom and top locations.

Reinforcement (rebar) layers in membrane, shell, and surface elements are treated as section points for output purposes; each reinforcement layer has a unique name. For example, you can request results from the section points at the top surfaces of certain shells in the model, from the middle section points of certain beams in the model, or from a named reinforcement layer in certain membrane elements in the model.

Contour plots displaying output at two section points vary in appearance depending on the type of shell. In a conventional shell the two contours appear as a double-sided shell with different contours on each side, while in a continuum shell (or a composite solid) the two contours appear as distinct two-dimensional contour fronts at each section point location. Axisymmetric shells, which are composed of line elements swept to form a three-dimensional surface, exhibit neither behavior because they are not considered three-dimensional shells; instead, these shells display output at a single section point only.

In many cases the location of a section point is described in terms of its position relative to the midpoint of the cross-section. For two-dimensional beams this relative position is reported as a fraction of the distance between the midpoint of the cross-section and the top or bottom surface of the section. For shells this relative position is reported as a fraction of the distance between the midpoint of the cross-section and the SPOS or SNEG surface of the section. Reinforcement layers are indicated by their unique name.

For beams with three-dimensional sections, the relative position of a section point can be reported as a fraction of the distance between the midpoint of the cross-section and

- the top or bottom of the section (along the local 2-axis of the section) or
- the left or right side of the section (along the local 1-axis of the section).

For circular beams the section point locations are defined as a relative position with respect to the section radius and an angle rotated from the positive 1-axis about the midpoint of the cross-section.

For more information about where section points are located in different section types, see the following:

- *Beam Elements*
- *Shell Elements*

For more information about reinforcement layers, see *Defining Reinforcement*.

Abaqus/CAE provides the following methods for choosing the section points to display:

- Display a category of elements and a location. Examples of element categories include element types and materials. The location can be the top or bottom surface or both.
- Display the section point containing the critical value (absolute maximum, maximum, or minimum). Abaqus/CAE searches for the critical value across all of the section points in each element. The resulting plot is called an envelope plot.

Additional information

• *Selecting model data and analysis results to plot*

Select the section point using categories and a location

1. Display a contour plot of the desired field output variable. For more information, see *Producing a contour plot*.
2. Locate the **Section Points** dialog box.

From the main menu bar, select **Result->Section Points**.

The **Section Points** dialog box appears.

3. From the **Selection method** field, select **Categories**.

   The elements within each category share common section properties, such as the following:

   - The section type associated with the elements in this category.
   - The material name or composite specification included in the section definition.
   - The number of section points through the section.
   - The cross-section geometry (beams only).

4. From the **Active locations** field, select **Bottom** or **Top** to activate only the **Bottom Location** or **Top Location** section point. For contour plots of integration point variables on three-dimensional shells or composite solids, select **Top and bottom** to display output simultaneously at both section points. Only contour plots of this type can display output for both section points simultaneously; in all other contexts, Abaqus/CAE displays output only at the **Bottom Location** section point when you select **Top and bottom**.

   Abaqus/CAE displays output only at active locations. The **Bottom Location** and **Top Location** fields display the two section points at which you can display output for the selected category. By default, these fields display the SNEG and SPOS section points, respectively.

5. In the **Category** table, select the element category or categories for which you want to change the output location and select the location. If you selected **Top and bottom** from the **Active locations** field, select both locations.

   The **Available Section Points in Cross-Section** field in the bottom half of the dialog box lists the section points from which output was saved for the category and location you have selected. If you have selected more than one category, only those locations that are common to all of the selected categories appear in the list.

6. In the **Available Section Points in Cross-Section** field, select the section point location to display.
7. Click **Apply** to view your settings.

   The model plot in the current viewport changes to display values from the specified section points. The plot legend, if active, changes to identify the specified section points. For information about the plot legend, see *Customizing the legend*.

   Results for subsequent *contour*, *symbol*, and *material orientation* plots; *tabular reports*; model probing; and *X–Y data objects along a path* will be obtained from the section points you have specified.

8. Click **OK** to select the section points and to exit the dialog box.

Select the section point containing the critical value

1. Display a contour plot of the desired field output variable. For more information, see *Producing a contour plot*.
2. Locate the **Section Points** dialog box.

   From the main menu bar, select **Result->Section Points**.

   The **Section Points** dialog box appears.

3. From the **Selection method** radio buttons, select **Categories**.
4. From the **Active locations** radio buttons, select **Envelope**.
5. From the **Criteria** field of the dialog box, select the criterion (absolute maximum, maximum, or minimum) that will determine the section point containing the critical value.

6. From the **Position** field, select the algorithm that Abaqus/CAE will use to determine the section point containing the critical value.
   - If you choose **Integration point**, Abaqus/CAE compares the value at all of the integration points of each section.
   - If you choose **Centroid**, Abaqus/CAE compares the value averaged at the centroid of each section.
   - If you choose **Element nodal**, Abaqus/CAE compares the value extrapolated to the nodes of each section.

7. Click **Apply** to view your settings.

   The model plot in the current viewport displays the critical section point and its value for each element in the mesh. For information about the plot legend, see *Customizing the legend*.

8. If you are displaying a composite layup, you can determine how Abaqus/CAE displays the critical section point by selecting ply options for the contour plot.
   a. From the main menu, select **Options->Contour**.
   b. From the **Contour Plot Options** dialog box that appears, click the **Other** tab.
   c. From the options at the bottom of the tabbed page, do the following:
      - Toggle on **Show labels of plies that match criteria** to see labels for the critical plies in the contour plot.
      - Toggle on **Color by plies that match criteria** to change the value that is being contoured from the field output variable to a ply name. The contour plot updates to show the critical plies in the model. The legend updates to show the names of the plies that are being displayed.

9. Click **OK** to retain your settings and to exit the dialog box.

Selecting section point data by ply

You can use the **Section Points** dialog box to control the section points from which Abaqus obtains integration point results and material orientations by selecting a ply and the location from within the ply at which Abaqus/CAE reads the results.

Contour plots displaying output at both the top and bottom of the selected ply vary in appearance depending on the type of composite layup. In a conventional shell composite layup the two contours appear as a double-sided shell with different contours on each side. In a continuum shell composite layup or a solid composite layup the two contours appear as distinct two-dimensional contour fronts at each section point location.

1. Display a contour plot of the desired field output variable. For more information, see *Producing a contour plot*.
2. Locate the **Section Points** dialog box.

   From the main menu bar, select **Result->Section Points**.

   The **Section Points** dialog box appears.
3. From the **Selection method** radio buttons, select **Plies**.
4. To limit the list for easier selection, type a filter pattern into the **Name filter** field, and press the **[Enter]** key to apply the filter.

   If viewport linking is activated, you can toggle on **Show common list between linked viewports** to display only the plies that are common to all linked viewports.
5. From the list of plies in the layup, select the desired ply.
6. From the **Ply result location** radio buttons, do one of the following:
   - Select **Bottommost** to select the bottommost section point in the selected ply.
   - Select **Middle/Single section point** to select the middle section point or the only section point in the selected ply. The middle section point is the middle of the available locations in the section, not the middle of the total locations. For example, if results are available at TOP and BOTTOM only, the number of available locations is two and the “middle” will correspond to the TOP location.
   - Select **Topmost** to select the topmost section point in the selected ply.
   - Select **Topmost and bottommost** to select both the topmost and bottommost section points in the selected ply.
7. Click **Apply** to view your settings.

   The model plot in the current viewport changes to display values from the specified ply at the specified location. The plot legend, if active, changes to identify the plies. For information about the plot legend, see *Customizing the legend*.
8. Click **OK** to retain your settings and to exit the dialog box.

Additional information

- *Selecting model data and analysis results to plot*
- *Linking viewports*

Selecting contact output

Contact output data for nonkinematic scalar output variables such as CPRESS, COPEN, CSLIP, CSHEAR1, and CSHEAR2 are consolidated from all continuum element–based surfaces into a single data set.
Data for overlapping surfaces are combined. If contact output is requested for membrane or shell elements, the data for the SPOS and SNEG surfaces are contained in separate data sets. When contact surfaces are double-sided or node-based, separate contact output data sets are created for each contact surface pair.

The method that Abaqus/CAE uses to combine overlapping facet data in the consolidated data set depends on the output variable:

- For the variables CPRESS, CSHEAR, and CSLIP, Abaqus/CAE uses the largest value, positive or negative, in the model for the selected variable.
- For the variable COPEN, Abaqus/CAE uses the minimum value for contact opening, which can reflect the smallest opening in the model or, if there are overclosures, the greatest overclosure value.
- For the vector variables CNORMF and CSHEARF, Abaqus/CAE evaluates the sum of all vector values for the individual surface pairs and stores the resultant vector in the consolidated data set.
- For the variable CSTATUS, Abaqus/CAE assigns one of four contact statuses to individual nodes on the main and secondary surfaces.

  Abaqus/CAE uses the following settings in the consolidated data set:

  - **Sticking** if any of the overlapping regions are in contact and sticking,
  - **Slipping** if any of the overlapping regions are in contact and slipping,
  - **Bonded** if any of the overlapping regions are in contact and bonded, and
  - **Not in contact** if the regions are not in contact.

The variable names that appear in Abaqus/CAE for each data set indicate the regions to which the contact output applies. For example, a CPRESS data set for an individual contact surface pair (for a double-sided or node-based contact surface) is named `CPRESS SECONDARY NAME/MAIN NAME`; the data sets for contact on the SPOS or SNEG side of a membrane or shell surface are named `CPRESS SPOS` and `CPRESS SNEG`, respectively; and all other surface variable output is named `CPRESS`.

Contact output data for any individual contact surface pair, without summed contributions from other contact pairs, are accessible using the **Create Field Output** dialog box or the Abaqus Scripting Interface to the output database; these data sets have names of the form `CPRESS SECONDARY NAME/MAIN NAME`.

Because CPRESS, CSHEAR1, and CSHEAR2 are associated with particular surfaces defined in the model, the surfaces must be included in the current display group when these output variable results are probed. These variables can be probed only at the nodes.

Additional information

• *Selecting model data and analysis results to plot*
• *Accessing an Output Database*

Selecting result options

Abaqus/CAE offers several methods for you to display field output results stored in the output database. Some of these methods require that field output data originally saved to the output database at element centroid or integration point locations be calculated at nodal locations. Such calculations apply to line- and banded-type contour plots, probing at nodal locations, forming a display group or color coding based on result values, and extracting element-based *X*–*Y* data along a path. You can control some of the options related to these computations, such as how element-based results are averaged.

In addition, you can choose to apply coordinate system transformations to your field output results; you can choose from several display forms for complex results: the magnitude, phase angle, real component, imaginary component, or value at a specified angle; and you can choose whether or not to cache results in memory to improve performance.

Select **Result->Options** to locate the options that control the result calculations, result transformations, display of complex results, and result caching. This section discusses the result computations and the options that affect these computations.

In this section:

- *Understanding how results are computed*
- *Understanding result value averaging*
- *Understanding complex results*
- *Understanding results caching*
- *Displaying field output values or discontinuities*
- *Controlling result averaging*
- *Controlling computations at region boundaries*
- *Transforming results into a new coordinate system*
- *Controlling the form of complex results*
- *Controlling results caching*

Understanding how results are computed

The computations necessary to display results stored in the output database depend on whether the results are for a node-based quantity, such as displacement or velocity, or for an element-based quantity, such as stress or strain.

How node-based field output results are computed

Node-based field output variables are written to the output database at each node, along with any nodal transformations applied during model creation. For the display of nodal field output variables, Abaqus/CAE reads the required values from the output database for each node included in the plot. By default, these values are displayed in the global coordinate system; you can choose to apply the nodal transformations to the results or to apply a user-specified coordinate system transformation. The final values are then used to produce contours, nodal probe values, display groups or color coding based on results, or *X*–*Y* data along a path.

How element-based field output results are computed

Element-based field output variables are written to the output database at the integration points, the element centroid, or the element nodes, depending on the variable. For the display of element-based field output variables, Abaqus/CAE reads values from the output database for all elements connected to all nodes included in the plot. Computations are then applied to these values to produce contours, nodal probe results, display groups or color coding based on results, or *X*–*Y* data along a path.

For results saved to the output database at the integration points or at the element centroid, the first computation applied is extrapolation. (Results saved at the element nodes do not require extrapolation.) For contour plots only, you can choose quilt-type extrapolation, in which case the remaining computations discussed below do not apply. To learn more about quilt-type extrapolation, see *Understanding how contour values are computed*. For all other methods of results display, Abaqus/CAE extrapolates results to the nodes using weighting appropriate for the element type and shape.

Extrapolated values are generally not as accurate as the values calculated at the integration points. Therefore, adequately detailed meshing is recommended around nodes where accurate nodal values of such element results are needed. You should be particularly careful interpreting output variables extrapolated to the nodes for second-order elements with midside nodes outside the quarter-point region, such as when one edge is collapsed in two dimensions or one face is collapsed in three dimensions.

Extrapolation of element tensor quantities is performed on the individual tensor components in the local material coordinate system. Nodes common to two or more elements receive extrapolated values from all contributing elements. Depending on the characteristics of your model, these contributions may originate from more than one *result region*. If all contributions at a node originate from a single result region, the values are combined as necessary in further computations. If contributions are received from more than one result region, you can choose to respect the region boundary and keep the contributions separate in further computations or to ignore the region boundary and combine the values. The default result regions in an output database duplicate the regions that were used to assign section properties to the model prior to analysis. Alternatively, you can select element sets or display groups to use as result regions. For more information, see *Controlling computations at region boundaries*.

If invariants or components are requested, you can specify whether Abaqus/CAE should use the extrapolated data from each element or the combined data from all contributing elements to compute the invariants. By default, invariants are computed before the extrapolated results are combined (averaged). Contour plots of invariants or components may be affected by the order in which Abaqus performs the computations. For example, values for the von Mises stress may exceed the yield stress of inelastic materials; in addition, the invariant results may not take into account situations where the material orientations vary within a finite element in a non-isoparametric fashion. If invariants are computed after averaging, Abaqus determines the orientations at a node by averaging the contributing element orientations; component values will be affected if the orientations differ between contributing elements.

If you select element sets to define the result regions and invariants will be computed after averaging, the element sets that you select must contain compatible elements. Compatible elements

- share the same basic element type (continuum, shell, beam, etc.),
- use interpolation functions of the same order (first-order elements versus second-order elements), and
- have the same integration scheme (reduced integration, full integration, etc.).

Finally, computations depend on whether you choose to display the field output values or discontinuities; discontinuities are the differences in field output values between adjacent elements.

- **Field Output**: For the display of field output values, the calculated invariants or components at nodes common to two or more elements are averaged conditionally, depending on the compatibility of contributing *result regions* and on options you select. For more information, see *Understanding result value averaging*.
- **Discontinuities**: For the display of discontinuities, the calculated invariants or components at nodes common to two or more elements are compared to determine the greatest difference, depending on the compatibility of contributing result regions and on options you select. Nodes associated with only one element and nodes receiving equivalent values from all contributing elements will show a value of zero in a plot of discontinuities.

For more information, see *Displaying field output values or discontinuities*.

How result transformations are computed

Both node- and element-based results can be transformed into a user-specified coordinate system, and angular transformations can be applied to coordinate- and distance-based nodal vector results; see *Transforming results into a new coordinate system*, for information on applying a transformation to your results.

Element-based results for three-dimensional continuum elements and all node-based results are transformed into the specified coordinate system based on the locations of the results. The 1-, 2-, and 3-directions for the transformed results correspond to the *X*-, *Y*-, and *Z*-directions of a rectangular coordinate system; the *R*-, θ-, and *Z*-directions of a cylindrical coordinate system; and the *R*-, θ-, and ϕ-directions of a spherical coordinate system.

Element-based results for two-dimensional continuum elements, shell elements, and membrane elements are transformed by rotating the results about the element normal at the element result location. The 2-direction for the transformed results is determined by the projection of the rectangular *Y*-direction or the cylindrical or spherical θ-direction onto the element plane. If the *Y*- or θ-direction of the user-defined coordinate system forms an angle less than 30° with the element normal, the next axis that follows the *Y*- or θ-axis in a cyclic permutation of the axes is projected on the element plane instead to form the local material 2-direction, and a warning message is displayed. This method of results transformation is different from the method used by Abaqus/Standard and Abaqus/Explicit to compute local orientations (see *Orientations*).

Element-based results for beam and truss elements cannot be transformed; they are always displayed in the local element orientation system. In addition, element results for rebar and for CAXA, SAX, or SAXA elements are not transformed.

When you transform results to a user-specified coordinate system, you can control the following additional aspects of the coordinate system transformation:

- You can include or exclude the effects of the current deformation in the transformation calculations. Deformation effects are not scaled; Abaqus/CAE performs these calculations using a deformation scale factor of 1.0. Including deformation effects in a transformation can change the orientation of node-based coordinate systems, the projection of coordinate systems on shell and membrane elements, and the orientation of location-dependent cylindrical and location-dependent spherical coordinate systems.
- You can adjust the results to account for the rigid body transformation of the coordinate system. You can adjust the display of primary variable results or results from both the primary and deformed variables.

Rigid body transformation helps you to understand relative displacements when large rigid body displacements are present. Rigid body transformation applies a transformation to the model to negate the displacement and rotation of a user-specified coordinate system that follows nodes (see *Creating coordinate systems during postprocessing*, for more information) as these nodes deform from frame to frame. The rigid body transformation is determined as the translation and rotation necessary to transform the nodes of the specified coordinate system from their current locations back to their original locations.

You can apply angular transformations for coordinate- and distance-based nodal vector results. The angular transformation computes components in terms of R, θ, and Z for cylindrical coordinate systems and in terms of R, θ, and ϕ for spherical coordinate systems.

You can apply layup orientation transformations for results that include output from the field output variable SORIENT and include composite sections. The tensor and vector fields will be transformed into the layup orientation defined in the composite section definition.

Understanding result value averaging

Result value averaging is applicable to the display of element-based field output variables, such as stress or strain, using any of the following methods:

- line- or banded-type contours,
- probing at nodal locations,
- forming a display group or color coding based on result values,
- extracting *X*–*Y* data from fields,
- extracting *X*–*Y* data along a path, or
- generating reports for field data.

The display of node-based field output variables (such as displacement), quilt contour plots, and plots of discontinuities does not involve value averaging.

Extrapolated output database values at nodes common to two or more elements are averaged conditionally. You can use the following options to control the extent to which Abaqus/CAE averages the values at the nodes:

- Select *result regions* and choose whether averaging takes place across region boundaries.
- Choose whether shell and membrane feature edges are treated as region boundaries for averaging.
- Include or exclude contributions from elements that are not displayed.
- Choose whether invariants are computed and components are extracted before or after averaging.
- Select a threshold value to control averaging based on the difference between the contributing element values (this option is available only if you choose to calculate invariants before averaging).

Abaqus/CAE averages values at nodes common to two or more elements when the contributing elements lie in the same result region. The default result regions are the same regions defined when you assign sections to your model; you can also define custom result regions using saved element sets or display groups. You can choose whether or not Abaqus/CAE averages values at nodes common to two or more result regions. You can suppress averaging across regions (use the region boundaries) to emphasize any discontinuities at region boundaries, or you can request averaging across regions (ignore region boundaries) to produce a more continuous effect. For example, *Figure 1* shows a contour plot using region boundaries on the left and averaging across region boundaries on the right.

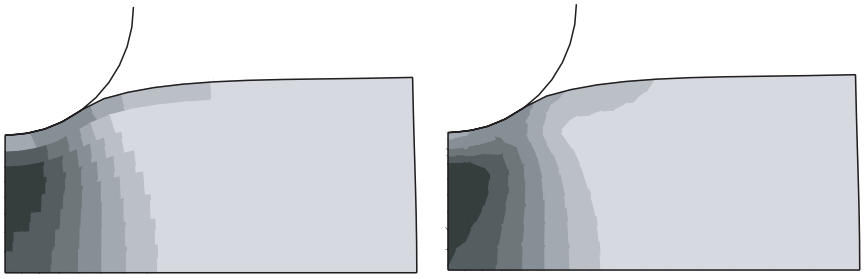


Figure 1: Contour plots using region boundaries and averaging across region boundaries.

If you choose to average across regions, the extent to which values are averaged is controlled by the threshold that you specify as follows:

$$\text{relative nodal variation} = \frac{(\text{maximum at node} - \text{minimum at node})}{(\text{maximum over active regions} - \text{minimum over active regions})}$$

If the relative nodal variation for each node included in the plot is less than your averaging threshold, values of contributing elements are averaged at that node. If the relative nodal variation exceeds your setting, the values are not averaged. When you decide to average across regions, the averaging threshold is considered in relation to the variation of values across the whole model unless you limit averaging to the displayed elements. Setting a high averaging threshold would allow you to smooth out all but the most extreme discontinuities relative to results across your model or all displayed results.

If you choose not to average across regions, the extent to which values are averaged at nodes within each region is controlled by the averaging threshold that you specify as follows:

$$\textbf{relative nodal variation} = \frac{\textbf{(maximum at node} - \textbf{minimum at node)}}{\textbf{(maximum within region} - \textbf{minimum within region)}}$$

If the relative nodal variation for each node included in the plot is less than your averaging threshold, values of contributing elements are averaged at that node. If the relative nodal variation exceeds your setting, the values are not averaged. When you suppress averaging across regions, the averaging threshold is considered in relation to the variation of values within each region rather than across the whole model or all displayed results.

The lower the averaging threshold you apply, the more the displayed values depict the individual element results. Conversely, a higher averaging threshold produces a smoother effect with fewer noticeable discontinuities from element to element.

When using threshold-based averaging in conjunction with the extraction of *X–Y* data from element-based fields, you might want to adjust the threshold value. The element region used when extracting *X–Y* data is reduced to include only the elements that are attached to the specified nodes.

If you choose to compute invariants and extract components after averaging,

- Abaqus/CAE does not apply the averaging threshold;
- tensors are transformed to an average orientation at each node prior to averaging; and
- result regions, if used, must contain compatible elements.

For more information, see *Understanding how results are computed*.

Select **Result->Options** to locate the options that affect averaging. *Controlling result averaging*, and *Controlling computations at region boundaries*, provide details on using the averaging options.

Understanding complex results

If the current step is a steady-state dynamic analysis, the value of an output variable such as stress (S) or displacement (U) can be a complex number with both real and imaginary components. Complex number results are saved during the analysis as pairs of real and imaginary data.

When you use Abaqus/CAE to display complex results, you can choose any of the following forms to view the analysis data:

- **Magnitude** displays the combined magnitude of both the real and imaginary portions of the result value.
- **Phase angle** displays the angle between the positive horizontal axis and the plotted point (*real, imaginary*). This selection is valid only for scalars or for vector and tensor components.
- **Real** displays only the real component. This option is the default setting.
- **Imaginary** displays only the coefficient of the imaginary component of the complex result.
- **Value at angle** displays the combined value of the real and imaginary portion of the result at an angle that you specify. When the angle is 0°, the real part of the result is displayed; when the angle is −90°, the imaginary part of the result is displayed.
- **Envelope: Min** displays the minimum of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.
- **Envelope: Max** displays the maximum of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.
- **Envelope: Max absolute** displays the maximum absolute of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.

The **Magnitude** is calculated after any invariants, such that the magnitude value is the square root of the sum of the squares of the real and imaginary invariant components. Similarly, the other complex forms are also calculated after invariants. **Value at angle** and the complex envelope values are exceptions; they are calculated prior to invariants to preserve the proper physical meaning.

You can also choose to animate complex results using harmonic animation. This technique animates complex field output by displaying the **Value at angle** through a sequence of angles. Abaqus/CAE generates angles ranging from 0° to 180° or from −180° to 180°, according to your specification, and displays the value of the complex results at each angle.

The magnitude and phase angle are related to the real and imaginary components with the usual expressions. For example, the real and imaginary displacement components, U_r and U_i, are related to the magnitude, $\overline{U}$, and phase angle, ϕ, as follows:

$$U_r = \overline{U} cos\phi$$

and

$$U_i = \overline{U} sin\phi.$$

The complex results represent a time-domain variation of the form:

$$\begin{aligned} U(t) &= U_r cos\Omega t - U_i sin\Omega t, \\ &= \overline{U} cos\left(\Omega t + \phi\right), \end{aligned}$$

where Ω is the excitation frequency. This time variation is shown when you request a harmonic animation. The value of the result at a selected angle, θ, is obtained by using $\theta = \Omega t$ so that

$$\begin{aligned} U(\theta) &= U_r cos\theta - U_i sin\theta, \\ &= \overline{U} cos\left(\theta + \phi\right). \end{aligned}$$

The magnitude, $\overline{U}$, for a given variable is displayed when $\theta = -\phi$. Stepping θ from −180° to 180° corresponds to one full animation cycle.

Additional information

- *Selecting complex results*
- *Controlling the form of complex results*
- *Harmonic animation*

Understanding results caching

A cache is a portion of system memory. You can choose to store analysis results in a cache during postprocessing to speed up the generation of images on the screen. When results are cached, Abaqus/CAE can plot results without recomputing values for display or accessing the output database file as often. This can greatly improve performance. However, caching results will increase memory usage. If you find that Abaqus/CAE is consuming a lot of memory on your workstation and consequently hindering the performance of your system, you can disable the results caching options to erase the memory associated with results caches. Clearing this memory can increase overall performance; however, the display of results will be slower. Switching to a new output variable or output database will automatically clear the current results cache. For more information, see *Controlling results caching*.

Abaqus/CAE can display data from an output database while an analysis job is still running and while the database continues to be updated. After most Visualization module operations and during animations, Abaqus/CAE monitors the output database for updated results and updates the current viewport accordingly. If you are displaying data from a remote output database, the performance of Abaqus/CAE may be degraded if the time taken to monitor the database over the network is significant. To increase the performance, you can reduce the frequency with which Abaqus/CAE monitors the output database for updates, or you can disable the monitoring. For more information, see *Accessing an output database on a remote computer*.

Additional information

- *Controlling results caching*

Displaying field output values or discontinuities

For element-based field output variables, such as stress or strain, you can choose to display the field output values themselves or the *differences* in field output values between adjacent elements. This choice is applicable to line- and banded-type contours, probing at nodal locations, forming display groups and color coding based on result values, or extracting *X–Y* data along a path.

The default is to display **Field output** values. If you choose to display **Discontinuities**, the **Averaging** threshold is not applicable.

Figure 1 displays a contour plot of field output values on the left and a contour plot of discontinuities on the right.

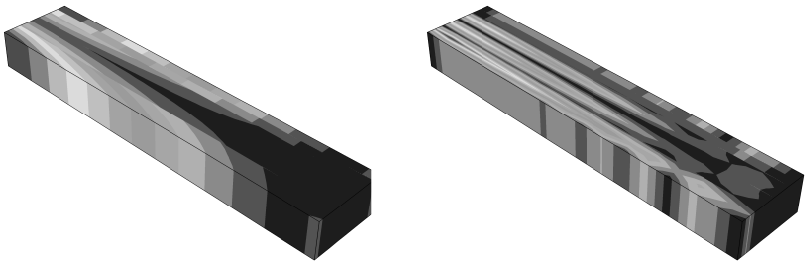


Figure 1: Contour plots showing field output values and discontinuities in values.

1. Locate the **Quantity to Plot** options.

   From the main menu bar, select **Result->Options**; then click the **Computation** tab in the dialog box that appears. The **Quantity to Plot** options are at the top of the page.
2. Click **Field output** or **Discontinuities** to choose the quantity that you want to plot.
3. Abaqus displays results in the current viewport according to your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding how results are computed*

Controlling result averaging

You can control the averaging of element-based field output results, such as stress or strain. Averaging is applicable to creating line- and banded-type contours, probing nodal locations, forming display groups and color coding based on result values, or extracting *X–Y* data along a path.

Element values are first extrapolated to the nodes. Nodes common to two or more elements will receive multiple contributions. To control the averaging of multiple contributions, you can:

- Enable or suppress averaging.
- Define result regions and choose whether averaging will cross them.
- Include or exclude shell and membrane feature edges as additional region boundaries for averaging.
- Suppress averaging of elements that are not displayed.
- Choose whether Abaqus/CAE computes invariants before or after averaging.
- Set the averaging threshold.

The averaging threshold governs the extent of averaging; it is available when Abaqus/CAE computes invariants or extracts components before averaging. If the relative difference between contributions at a node is greater than the threshold percentage you set, Abaqus/CAE will not average the contributing values and your results will appear discontinuous at that node. Use a higher percentage to produce a smoother, more continuous effect.

For information on averaging across regions, see *Controlling computations at region boundaries*.

1. Locate the **Averaging** options.

   From the main menu bar, select **Result->Options**; then click the **Computation** tab in the dialog box that appears. The **Averaging** options appear.

2. To control whether Abaqus/CAE averages output from multiple elements at shared nodes, toggle **Average element output at nodes**.

   Suppress averaging to identify discontinuities in the field output or to establish contour limits based on the extrapolated, unaveraged results.

3. Toggle off **Use region boundaries** to ignore region boundaries and average results across the entire model.

4. If region boundaries are used, specify the result regions to set the boundaries. For more information, see *Controlling computations at region boundaries*.

   Tip: To view the current result regions, color code an undeformed or deformed plot using the **Averaging regions** method of attribute selection. (For more information, see *Coloring all geometry in the Visualization module*.)

5. If region boundaries are used, by default Abaqus/CAE includes shell and membrane feature edges as additional region boundaries. To change the minimum angle used to create feature edges, click  and change the **Feature Angle** setting in the **ODB Display Options** dialog box. Feature edges correspond to the undeformed model and are not recalculated for model deformation. (For more information, see *Defining model feature edges*.)

   To average the results across feature edges, toggle off **Include shell/membrane feature edge boundaries**.

6. To control whether Abaqus/CAE averages nodal results based on all contributing elements or based only on the elements in the current display group, toggle **Average only displayed elements**.

7. Choose whether Abaqus will compute invariants (scalars) and extract components before or after averaging results over the specified regions.

   By default, Abaqus computes the scalars before averaging; if you choose to compute the results after averaging, Abaqus averages all results and their orientations to maintain a valid basis for computing the invariants.

   Tip: To view the nodal-averaged orientations in a contour plot, you can display their labels using the **Contour Plot Options** dialog box. For more information, see *Displaying nodal-averaged orientations*.

8. If you compute scalars before averaging, you can set the **Averaging Threshold (%)** to control the averaging between neighboring elements. A value of 0 suppresses all averaging; a value of 100 averages all results.
9. Click **Apply** to implement your changes.

   Abaqus averages result values for the display in the current viewport according to your specifications. The contour legend, if active, changes to state the averaging threshold that you have specified. For more information on the contour legend, see *Customizing the legend*.

   By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding result value averaging*

Controlling computations at region boundaries

The default result regions in an output database duplicate the regions that were used to assign section properties to the model prior to analysis. Alternatively, you can select element sets or display groups to use as result regions. You can control computations at nodes common to two or more result regions. Choose to ignore region boundaries to computationally combine results for a smoother, more continuous effect. Choose not to ignore region boundaries to visually emphasize model discontinuities. See *Figure 1* for an example of contour plots with and without averaging across regions; averaging across regions occurs when you choose to ignore region boundaries.

1. Locate the **Region Boundaries** options.

   From the main menu bar, select **Result->Options**; then click the **Computation** tab in the dialog box that appears. The **Region Boundaries** options appear within the **Averaging** controls.

2. To govern computations at nodes common to two or more result regions, toggle **Use region boundaries**.

   To use region boundaries means *not* to combine values from neighboring result regions when computing nodal averages or when computing discontinuities.

3. Choose one of the following options to define the result regions:

   Tip: To view the current result regions, color code an undeformed or deformed plot using the **Averaging regions** method of attribute selection. (For more information, see *Coloring all geometry in the Visualization module*.)

   ### ODB regions

   By default, the result regions are the same as the section assignment regions saved in the output database (`.odb`) file. For more information, see *Assigning sections, orientations, normals, and tangents to a part*.

   ### Element sets

   Specify the results regions by selecting element sets.

   ### Display groups

   Specify the result regions by selecting display groups.

   Note:

   If the desired display groups do not exist, select **Tools->Display Group->Create** to create them (for more information, see *Creating or editing a display group*).

   The region boundaries are defined using the display groups as they are defined at the time of selection. If you subsequently modify or delete display groups, the region boundaries do not change unless you redefine them.

4. If you chose the **Element sets** or **Display groups** method to set the region boundaries, complete the following:

   a. Click .

   Abaqus/CAE opens the **Specify Averaging Regions** dialog box.

   b. From the list on the left side of the dialog box, click the desired regions to highlight them in the list. Use **[Shift] + Click** and **[Ctrl] + Click** to select multiple items.

c. Click the right-facing arrow button in the center of the dialog box to copy the highlighted items to the **Selection** column. To move the entire list, click the double right arrow button.

d. To remove region selections, highlight them in the **Selection** column and use the left-facing arrow button.

Any elements that are not included in the region boundary selections will be colored using the "no results" color (for more information, see *Coloring elements with no results*). Selections that overlap will be combined.

5. By default, Abaqus/CAE uses feature edges as additional region boundaries. To change the minimum angle used to create feature edges, click  and change the **Feature Angle** setting in the **ODB Display Options** dialog box. Feature edges correspond to the undeformed model and are not recalculated for model deformation. (For more information, see *Defining model feature edges*.)

To average results across feature edges, toggle off **Include shell/membrane feature edge boundaries**.

6. Click **Apply** to implement your changes.

Abaqus computes result values for the display in the current viewport according to your specifications.

By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding how results are computed*

Transforming results into a new coordinate system

By default, Abaqus/CAE displays element-based field output results in the coordinate systems defined during preprocessing and node-based field output results in the global coordinate system.
If you defined nodal transformations during preprocessing, you can choose to apply these transformations to node-based results. Alternatively, you can choose to transform both element- and node-based results into a user-specified coordinate system or to apply an angular transformation to coordinate- and distance-based nodal vector results.

For vector quantities such as displacement, velocity, and acceleration, which are derived from nodal coordinates, Abaqus/CAE transforms the results to the requested coordinate system using the final quantity (saved vector) instead of the original coordinates. For example, the cylindrical displacement component is computed by projecting the displacement vector along the current r- or θ- direction. By comparison, taking the differential between the transformed nodal coordinates would yield a different result.

When you transform results into a local coordinate system, Abaqus/CAE rotates the global rectangular coordinate system as needed to align with the local system at the point of interest. The transformed results are reported in the rotated global rectangular coordinates.

For transformations to user-specified coordinate systems, Abaqus/CAE includes the effects of the current deformation by default when deformation effects are available. You can exclude these effects if you want transformation calculations to consider only the undeformed state. Deformation effects are not scaled; Abaqus/CAE performs these calculations using a deformation scale factor of 1.0. Including deformation effects in a transformation can change the orientation of node-based coordinate systems, the projection of coordinate systems on shell and membrane elements, and the orientation of location-dependent cylindrical and location-dependent spherical coordinate systems. For more information, see *Transformation of results*.

When you select a user-specified coordinate system for a transformation, you can also adjust the display of primary variable results or results from both the primary and deformed variables to account for the rigid body transformation of the coordinate system. Displaying deformations that account for rigid body transformations enables you to visualize the relative displacements of the model with respect to a moving, user-defined coordinate system. For more information about selecting results variables, see *Selecting the primary field output variable*, and *Selecting the deformed field output variable*.

You can apply angular transformations for coordinate- and distance-based nodal vector results. The angular transformation computes components in terms of R, θ, and Z for cylindrical coordinate systems and in terms of R, θ, and ϕ for spherical coordinate systems.

You can apply layup orientation transformations for results that include output from the field output variable SORIENT and include composite sections. SORIENT will be included as long as element material orientations were requested, which is the default. The tensor and vector fields will be transformed into the layup orientation defined in the composite section definition.

1. Locate the **Transform Type** options.

   From the main menu bar, select **Result->Options**; then click the **Transformation** tab in the dialog box that appears. The **Transform Type** options appear.

2. Select the transform type to use for your results.

   - Select **Default** to use the default coordinate systems defined for your model. Node-based results are displayed in the global system. Element-based results are displayed in the local orientations defined for the model; if none has been defined, the global system is used. A projection of the global system or local orientation is used for two-dimensional continuum elements, shell elements, and membrane elements.
   - Select **Nodal** to consider local directions that are defined at the nodes. Element-based results are displayed in the default coordinate system (as described above). Nodal transformations defined for the model are applied to node-based results; if none is defined, the global system is used.

- Select **User-specified** to transform the results for the entire model into a specified coordinate system.

  To limit the list for easier selection, type a filter pattern into the **Name filter** field, and press the **[Enter]** key to apply the filter.

  Select the transformation to apply from the list of coordinate systems that appears, or click  and select the coordinate system from the viewport. Coordinate systems defined either during model generation or during postprocessing are available. Systems designated with an asterisk are saved to the current output database.
- Select **Angular** to transform results for coordinate- and distance-based nodal vector results in terms of R, θ, and Z or R, θ, and ϕ. Select the transformation to apply from the list of cylindrical and spherical coordinate systems that appears, or click  and select the coordinate system from the viewport. Coordinate systems defined either during model generation or during postprocessing are available. Systems designated with an asterisk are saved to the current output database.
- Select **Layup orientation** to transform tensor and vector fields into the layup orientation defined in the composite section definition.

3. If you selected a user-specified coordinate system, you can also include the effects of deformation in the transformation to that coordinate system.

   Toggle on **Include effects of deformation (when available)** to include the effects of deformation, or toggle off this option to exclude the effects of deformation and to consider only the undeformed state in the transformation.

4. If you selected a user-specified coordinate system, you can also apply rigid body transformations to the presentation of primary and deformed variables.

   From the **Rigid Body Transformations** options, do one of the following:

   - Toggle on **Primary variable** to apply rigid body transformations to the primary variable results.
   - Toggle on **Primary variable** and **Deformed variable** to apply rigid body transformations to the results for both variables.

5. Abaqus displays results in the current viewport in the specified coordinate system or systems. Angular components are given in radians. Angular transformation component results are denoted using (AT: *CSYS-name*), and rigid body transformation component results are denoted using (RT: *CSYS-name*) to differentiate them from the results obtained using the default transformation type.

   By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding how results are computed*
- *Creating coordinate systems during postprocessing*
- *Why are datum coordinate systems so important?*
- *Creating field output by operating on fields*

Controlling the form of complex results

If the current step is a steady-state dynamic analysis, the value of an output variable such as stress (S) or displacement (U) can be a complex number with both real and imaginary components. You can control the form of complex numbers when displaying plots of a model or recording probed values from a model in Abaqus/CAE.

You can choose to view the magnitude, phase angle, real component, imaginary component, value at a selected angle, or complex envelope values of your analysis results. The selected form is applied to all complex results, except harmonic animations, in the current session of Abaqus/CAE until you make a new selection. Harmonic animations, by definition, always use the "value at angle" form regardless of the current numeric form.

1. Locate the **Numeric Form** options.

   From the main menu bar, select **Result->Options**; then click the **Complex Form** tab in the dialog box that appears. The **Numeric Form** options appear.

2. Select the form to use for complex numbers:

   - **Magnitude** to display the combined magnitude of both the real and imaginary portions of the result value.
   - **Phase angle** to display the angle between the positive horizontal axis and the plotted point (*real, imaginary*) that represents the complex number in rectangular coordinates. This selection is valid only for scalars or for vector and tensor components.
   - **Real** to display only the real component of the complex result value. This option is the default setting.
   - **Imaginary** to display only the coefficient of the imaginary component of the complex result value.
   - **Value at angle** to display the combined value of the real and imaginary portion of the result at an angle that you specify. When the angle is 0°, the real part of the result is displayed; when the angle is −90°, the imaginary part of the result is displayed.
   - **Envelope: Min** to display the minimum of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.
   - **Envelope: Max** to display the maximum of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.
   - **Envelope: Max absolute** to display the maximum absolute of the combined value of the real and imaginary portion of the result by varying the angle from 0° to 360°.

3. Abaqus updates the display in the current viewport according to your specifications. By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Selecting complex results*
- *Understanding complex results*

Controlling results caching

You can choose to store analysis results in a cache during postprocessing to speed up the generation of images on the screen. However, caching results will consume system memory and may decrease overall performance. Switching to a new output variable or output database will automatically clear the current results cache.

To increase performance, you can reduce the frequency with which Abaqus/CAE monitors the output database for updates, or you can disable the monitoring altogether. If you are displaying data from a remote output database, the performance of Abaqus/CAE may be degraded if the time taken to monitor the database over the network is significant. For more information, see *Accessing an output database on a remote computer*.

1. Locate the **Results Caching** options.

   From the main menu bar, select **Result->Options**; then click the **Caching** tab in the dialog box that appears.

2. Use the **Caching Options** field to speed up the generation of images on the screen.
   a. Toggle on **Cache deformed variable results** to store the deformed variable results for the current output database in memory.
   b. Toggle on **Cache primary variable results** to store the primary variable results for the current output database in memory.
   c. Toggle on **Cache cut fields** to store the values used for displaying cut models in memory.

3. Use the **ODB Access Options** field to increase performance when accessing an output database.
   a. Toggle off **Check for ODB file updates** to stop Abaqus/CAE from monitoring the output database for updates.
   b. Enter the minimum time interval in seconds between checks of the output database for updates. A larger number will reduce the frequency with which Abaqus/CAE checks the output database for updates.

4. By default, your changes are saved for the duration of the session and will affect all subsequent display of results. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding results caching*

Creating and saving new field output

The field and history output initially available in the output database are limited to those variables you have saved during the analysis. You can augment these analysis results by operating on field output to create new results.

Once created, you can save the new field output to the output database. You must open an output database file with write privileges if you want to save field output from the session to the file.

The following methods can be used to create and save new field output:

- Select **Tools->Field Output->Create From Fields** from the main menu bar or use the **Create Field Output From Fields** tool in the toolbox to create new field output variables by *operating on individual existing field output variables*.
- Select **Tools->Field Output->Create From Frames** from the main menu bar or use the **Create Field Output From Frames** tool in the toolbox to create new field output by *combining results from several output database frames*.
- Select **Tools->Field Output->Save Fields** from the main menu bar to save new field output.

For information on selecting and plotting output database field output, see *Selecting field output variables*.

In this section:

- *Building valid field output expressions*
- *Using operations on field output*
- *Combining results from several frames*
- *Creating field output by operating on fields*
- *Creating field output by operating on frames*
- *Saving field output created during the session*

Building valid field output expressions

To define a new field output variable by operating on individual existing field output variables, you *build an arithmetic expression* in the expression text field of the **Create Field Output** dialog box.

An expression is composed of one or more existing field output variable tags and, optionally, one or more operators, transformations, and scalars. See *Using operations on field output*, for information on supported operators. Expressions are evaluated as Python input; entries containing syntax errors will generate non-specific Python exceptions.

The **Create Field Output** dialog box lists field output by the variable name and by a unique "tag" created by Abaqus/CAE. The tag for a given field output variable is composed of the step and frame numbers prepended to the output variable name; for example, `s1f1_RF` is the reaction force at Step 1, Frame 1. This tag enables you to combine output from different steps and frames in one expression.

Field output variables can consist of different types of data that may not be compatible. The **Create Field Output** dialog box lists the type of each field output variable as one of three general types: scalar, vector, and tensor. The tensor types are further subdivided into five subtypes that describe the dimensionality of the tensor and the components available. The following rules apply:

- Only like data object types and subtypes can appear in an expression.
- Operations on complex data use only the real component.
- The multiplication and division operations are not supported between two vector objects or between two tensor objects.
- Operations on tensors are performed on the tensor component data that Abaqus/CAE reads from the output database. Subsequently, the display of the resulting field output variable may give unexpected values for extrapolated components or for computed invariants. For example, if the components of a stress tensor are negative, applying the absolute value operation to the stress tensor will produce positive values for the stress components, but the values of pressure—an invariant computed after the absolute value operation has been applied—may be negative. Similarly, applying the sine operation to a stress tensor will produce component values within the range {−1, 1}, but the extrapolation used to compute contour values of such components may produce values beyond this range.
- Operations on fields associated with different regions are not supported.

The following examples demonstrate valid field output expressions:

Example 1

To create a field output variable by finding the difference in the stress fields for two increments, type:

```
s1f2_S - s1f1_S
```

in the expression text field of the **Create Field Output** dialog box.

`s1f2_S` and `s1f1_S` are field output variables representing stress at two different increments of a particular step. The result of this equation is a field output variable representing the difference in the stress fields for the two increments.

Example 2

To create a field output variable by expressing pressure in decibels as a function of acoustic pressure, type:

```
20.0 * log10(s1f1_POR/Pref)
```

in the expression text field of the **Create Field Output** dialog box.

`Pref` is the reference pressure. The result of this equation is a field output variable representing pressure in decibels as a function of the analysis variable POR.

Example 3

To create a field output variable by transforming a tensor result to a user-specified coordinate system:

1. Select **Transformation** from the **Function** list on the right side of the **Create Field Output** dialog box.
2. Select a tensor result from the list of field output variables; for example, `s1f10_S`.
3. Select a coordinate system transformation to apply; for example, a fixed coordinate system named `CSYS-1`.

   The expression `s1f10_S.getTransformedField(datumCsys=o_CSYS_1)` appears in the expression text field.

The result of this equation is a field output variable representing the transformation of the stress tensor `s1f10_S` to the `CSYS-1` coordinate system.

Additional information

- *Creating field output by operating on fields*
- *Creating coordinate systems during postprocessing*
- *Understanding how results are computed*

Using operations on field output

This section lists the operations that you can perform on individual field output variables using the **Create Field Output** dialog box and the arguments accepted by each operation.
Arguments to trigonometric functions are assumed to be in radians.

The following keys are used to classify function arguments:

Text keys for operations on field output:

A	The argument can be a field output variable, a floating point number, or an integer.
F	The argument must be a floating point number.
FO	The argument must be a field output variable.

Operations on field output:

+	Perform addition.
–	Perform subtraction or unary negation.
*****	Perform multiplication.
/	Perform division.
abs(A)	Take the absolute value.
acos(A)	Take the arccosine.
asin(A)	Take the arcsine.
atan(A)	Take the arctangent.
cos(A)	Take the cosine.
degreeToRadian(A)	Convert degrees to radians.
exp(A)	Take the natural exponential.
exp10(A)	Take the base 10 exponential.
log(A)	Take the natural logarithm.
log10(A)	Take the base 10 logarithm.
power(FO,F)	Raise a field output object to a power.
radianToDegree(A)	Convert radians to degrees.
sin(A)	Take the sine.
sqrt(A)	Take the square root.
tan(A)	Take the tangent.

Additional information

- *Building valid field output expressions*
- *Creating field output by operating on fields*

Combining results from several frames

To define new field output by combining results from several output database frames, you use the ***Create Field Output From Frames*** dialog box.

You can select from the following operations involving frames:

- Sum values over all frames
- Find the minimum value over all frames
- Find the maximum value over all frames

The **Create Field Output From Frames** dialog box initially contains no output. You select the output database frames to consider; all the results available for these frames can then be combined, or you can select individual field output variables to combine. If you choose to sum the values over the selected frames, Abaqus/CAE will create new field output variables representing the linear combinations of the selected results. Alternatively, you can choose to have Abaqus/CAE determine the minimum or maximum values of the selected results in the range of selected frames. Abaqus does not perform consistency checks on the physical validity of frame operations; for example, the linear superposition of two frames containing the results from load cases with different boundary conditions is allowed even though the combined results may not be physically meaningful.

Operations on tensors are performed in the local coordinate system, if it is available; otherwise, the global system is used. Abaqus/CAE assumes that the local coordinate systems are consistent for nonunary operations involving tensors.

Figure 1 shows an example of creating field output by combining the results from several load cases (see *Load cases* for more information on load cases). A new "load case" is created by summing the results from the selected load cases; you can assign a name to the new load case, as shown in the figure.

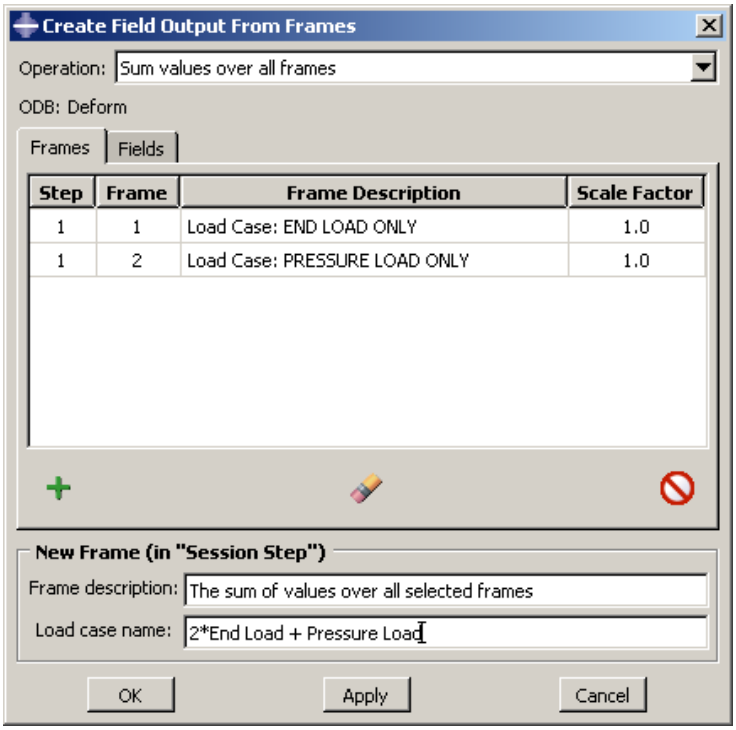


Figure 1: Creating field output by combining the results from several load cases.

Additional information

• *Creating field output by operating on frames*

Creating field output by operating on fields

You can create field output by computing new results using the field output available in the output database. For example, you might create new field output variables to show the differences in the stress fields between two increments. For detailed examples, see *Building valid field output expressions*.

You create new field output variables from fields by operating on analysis results found in the output database. You can display field output variables that you have created in the same ways as output database field output variables: in the form of a deformed, contour, or symbol plot; by probing any model or *X*–*Y* plot; as *X*–*Y* data obtained along a path through your model; or in a tabular report. Field output variables that you have created are saved in the session step until you end the session or close the output database from which the field output originates. You can also save the field output to the output database, in which case you must open the output database with write privileges.



Note: Field output created using complex results will contain only the real portion of the data. You cannot create field output that includes the imaginary part of complex results.

If you create field output that extracts an invariant scalar component of a tensor variable, such as the tensor variable's Mises stress or one of its principal stresses, contour plots of the resulting field output may be different than the original contour plots of the invariant. This difference occurs because these contours are calculated differently. Abaqus/CAE calculates the original invariant by first extrapolating the tensor to the nodes, calculating the invariant values, and averaging the values at the nodes. For the field output, Abaqus/CAE extrapolates the invariant values to the element nodes and averages them, which can result in a different contour plot.



Note: Some field output variables like PEEQ and PEMAG always have positive values at the integration points. For these variables extrapolation of the results from integration points to element nodes can result in negative values at some nodes. To avoid these negative value results, Abaqus/CAE clamps the element nodal values of such fields internally at 0.0 and resets negative values to 0.0. If you operate on these field output variables to create new field output, Abaqus/CAE does not clamp the values at 0.0 for the new resulting field output and the contour plot may show some negative values.

1. Locate the options for creating field output from fields.

   From the main menu bar, select **Tools->Field Output->Create From Fields**.

   The **Create Field Output** dialog box appears.

   Tip: You can also specify field output by using the  tool in the toolbox.

2. The default name for the new field output variable appears in the **Name** text field. To provide a more meaningful name, replace this default with the name of your choice (including blank spaces if you wish).
3. Build your expression in the expression text field using a combination of the following procedures:
   a. Select a function from the menu on the right side of the **Create Field Output** dialog box.

      - Select **Operators** to apply a built-in arithmetic operator to the selected field output.
      - Select **Transformation** to apply a coordinate system transformation to the selected field output.
      - Select **Scalar** to extract a scalar component from the selected field output.

      The list below the menu changes to reflect your selection.

   b. Select field output variables from the list on the left side of the **Create Field Output** dialog box. You can select field output for any step and frame in the analysis.



Tip: Place your cursor over the vertical lines separating each column in the output variable table, and enlarge the column widths to see the complete text.

The field output variable appears in the expression text field.

c. Select operators, transformations, or scalars from the list on the right side of the **Create Field Output** dialog box.

The selected function appears in the expression text field.



Note:

If you select a dynamic coordinate system (a system that follows nodes in your model) from the list of transformations, `deformationField=` appears in the expression text field inside the transform function. You must select a deformation field output variable that Abaqus/CAE will use to compute the position of the coordinate system.

d. Use standard mouse and keyboard editing techniques in the expression text field to position the cursor and configure your expression.

e. Adjust *the syntax of your expression* if necessary: parentheses may be needed.

f. If you make a mistake and wish to start over, click **Clear Expression**.

For more information, see *Building valid field output expressions*.

4. Click **OK** to create your new field output variable and to dismiss the dialog box.

Abaqus creates a new field output variable that is contained in Frame 0 of the session step by default and available from the **Field Output** dialog box. Current results plots, if any, are not affected.

Additional information

- *Selecting a specific results step and frame*
- *Building valid field output expressions*
- *Using operations on field output*
- *Creating coordinate systems during postprocessing*
- *Understanding how results are computed*
- *Transforming results into a new coordinate system*
- *Saving field output created during the session*

Creating field output by operating on frames

You can create field output by combining the results from several frames available in the output database. For example, you might create new field output to show the combined response to several load cases. For a detailed example, see *Combining results from several frames*.

You can sum field output over specified frames, and you can find the minimum or maximum values over specified frames. You can display field output that you have created in the same ways as output database field output variables: in the form of a deformed, contour, or symbol plot; by probing any model or *X–Y* plot; as *X–Y* data obtained along a path through your model; or in a tabular report. Field output that you have created is saved in the session step until you end the session or close the output database from which the field output originates. You can also save the field output to the output database, in which case you must open the output database with write privileges.



Note:

Field output created using complex results will contain only the real portion of the data. You cannot create field output that includes the imaginary part of complex results.

1. Locate the options for creating field output from frames.

   From the main menu bar, select **Tools->Field Output->Create From Frames**.

   The **Create Field Output From Frames** dialog box appears.

    **Tip:** You can also create field output from frames by using the  tool in the toolbox.

   By default, the **Frames** and **Fields** pages in this dialog box are empty.

2. Select the operation you wish to perform from the list in the **Operation** field.

3. Click  to select the frames that will be included in the operation.

   The **Add Frames** dialog box appears.

4. Select the analysis step from the list in the **Step** field.

   A list of the available frames for that step appears.

5. Select the individual frames you would like to include in the operation (see *Selecting multiple items from lists and tables*, for more information), or click **Select All** to include all the available frames.

6. Click **OK** or **Apply** in the **Add Frames** dialog box.

   The frames you selected appear in the **Frames** field of the **Create Field Output From Frames** dialog box. You can use the  button (**Remove Selected**) and  button (**Remove All**) at the bottom of the **Frames** field to modify your frame selections as needed. You can also continue to add frames from the **Add Frames** dialog box.

7. If you have chosen the sum operation, a **Scale Factor** column also appears in the **Frames** field of the **Create Field Output from Frames** dialog box. The default scale factor for every frame is 1.0. You can modify the scale factor for a selected frame by clicking in the **Scale Factor** column and entering a new scale factor value.

8. Click the **Fields** tab in the **Create Field Output From Frames** dialog box to select the field output that will be included in the operation.

   The field output variables available for the selected frames are shown, along with information on the output variable type (tensor, vector, or scalar). If you have chosen the minimum or maximum operation, the invariant/component information for tensor and vector variables and the option to select the output

position are also available. By default, all the available output variables have a check mark in the **Select** column.

a. Click in the **Select** column to unselect the output variables you do not wish to include in the operation.

b. If you have chosen the minimum or maximum operation, you can click in the **Invariant/Component** column and select the invariant or component that you wish to use to determine the minimum or maximum field results from the list that appears at the right. For example, if you select the Mises invariant for a stress variable, Abaqus/CAE creates a new stress field output variable in which the stress tensor at each point is obtained from the frame at which the maximum Mises value occurs (within the list of frames selected in Step 5).

c. If you have chosen the minimum or maximum operation, you can click in the **Position** column and select the output position at which to store the new field output. For example, if you want to create new field output from stress at integration points by finding the maximum values, you can choose to store the field output at **Element Nodal** positions, ensuring that contour plots display the expected results.

d. Use the **Select All**, **Unselect All**, and **Default** buttons to modify your field output selections as needed.

9. You can accept the default frame description provided by Abaqus/CAE, or you can supply your own in the **Frame description** field.

10. By default, Abaqus/CAE does not provide a load case name for the new field output; if your new field output represents an additional load case in your analysis, you can supply a name in the **Load case name** field. This name will be used only for organizational purposes.

11. Click **OK** in the **Create Field Output From Frames** dialog box to create your new field output and to dismiss the dialog box.

Abaqus operates on the selected field output over the selected frames to create new field output that is contained in a new session frame of the session step and available from the **Field Output** dialog box. The descriptions of the new variables indicate the operation by which they were derived. Current results plots, if any, are not affected.

When you choose the minimum or maximum operation, Abaqus also creates a field output variable for each result indicating the frames from which each minimum or maximum value was obtained. (The frames are identified by their index in the list of selected frames.) This variable, named *outputVariable*_Index, is available from the same frame of the session step as the actual results of the minimum or maximum operation and can be queried or plotted in the same way as other field output variables.



Tip: *outputVariable*_Index works better with Element Nodal, Nodal, or Centroid position since, plotting *outputVariable*_Index created at the integration point position is subjected to extrapolation and averaging, which may result in incorrect plots. To know more, please refer to: *Understanding how results are computed*

Additional information

- *Selecting a specific results step and frame*
- *Creating and saving new field output*
- *Saving field output created during the session*

Saving field output created during the session

You can save field output that you created during the session to the output database.
You must open an output database file with write privileges if you want to save field output from the session to the file. If you create new field output by operating on analysis results found in the output database, you can save the field output to an existing step in the output database or you can create a new step.

1. Locate the options for saving field output.

   From the main menu bar, select **Tools->Field Output->Save Fields**.

   The **Save Sessions Field Output Variables** dialog box appears. The field output that you created during the session is displayed in the table.
2. In the **Target step** field, select the step in which you want to save the field output.
3. In the **Target frame** field, select the frame in which you want to save the field output.
4. Select the field output from the table that you want to save to the output database.
5. If you want to create a new step in the output database file, click **Create Step**.
6. In the **Create Odb Step** dialog box that appears, do the following:
   a. Enter a name and description for the new step.
   b. In the **Domain** field, select one of the following options to specify the domain of the step:
      - **TIME**, and enter time period
      - **FREQUENCY**
      - **ARC_LENGTH**
      - **MODAL**
   c. Click **OK** to create the step.
7. Click **OK** to save the fields.

Additional information

- *Opening a model database or an output database*
- *Creating field output by operating on fields*
- *Creating field output by operating on frames*

Creating coordinate systems during postprocessing

You can create local (or "session") coordinate systems in the Visualization module for postprocessing purposes: field output results can be transformed to a specified user-defined coordinate system.
These user-defined coordinate systems can be fixed in space or move with specified nodes as the model deforms. Session coordinate systems persist for the duration of your session only and are available only for a single output database; however, you can save session coordinate systems to their associated output database file for use in subsequent Abaqus sessions.

You can rename or delete session coordinate systems from the **Coordinate System Manager** or the **Session Coordinate Systems** container in the Results Tree for the current output database. You can also display, highlight, and hide individual coordinate systems by using their shortcuts from the Results Tree or by adding them to display groups. See *Managing display groups*, for more information about adding both session-based and output database–based coordinate systems to display groups.

For information on controlling the display of all coordinate systems during postprocessing, see *Controlling the display of model entities*. For information on controlling the display of individual coordinate systems during postprocessing, see *Managing display groups*. For information on results transformation, see *Transforming results into a new coordinate system* and *Creating field output by operating on fields*.

In this section:

- *Methods for creating a coordinate system*
- *Creating a fixed coordinate system*
- *Creating a coordinate system following three nodes*
- *Creating a coordinate system following three nodes on a circle*
- *Creating a coordinate system following a single node*
- *Saving a coordinate system to an output database file*

Methods for creating a coordinate system

Select **Tools->Coordinate System->Create** from the main menu bar in the Visualization module to define a local coordinate system.
You choose the type of coordinate system and follow the prompts in the prompt area to define the coordinate system axes. The following types of coordinate systems are available:

- **Fixed system**
- **System following 3 nodes**
- **System following 3 nodes on a circle**
- **System following a single node**

You can use the **Session Coordinate Systems** container in the Results Tree or the **Coordinate System Manager** to rename or delete user-defined coordinate systems and to save the coordinate systems to an output database file for use in later Abaqus/CAE sessions.

Creating a fixed coordinate system

You define a fixed coordinate system by specifying the origin and two points. The line from the origin to point 1 is the *X*-axis (for a rectangular coordinate system) or the *R*-axis (for a cylindrical or spherical coordinate system); the plane containing the origin, point 1, and point 2 is the *X*–*Y* plane (for a rectangular coordinate system) or the *R*–θ plane (for a cylindrical or spherical coordinate system). The coordinate system remains fixed in space regardless of the motion of the model.

You can either select individual nodes in the viewport or enter the *X*-, *Y*-, and *Z*-coordinates of the points in the prompt area. When you select nodes in the viewport, the nodal coordinates of the current plot state are used to orient the coordinate system; that is, the true coordinates of the node on the undeformed or deformed shape are used (any deformation scale factor applied to the deformed shape is ignored).

1. From the main menu bar, select **Tools->Coordinate System->Create**.

   The **Create Coordinate System** dialog box appears.

   Tip: You can also create a coordinate system by using the tool in the toolbox or the **Session Coordinate Systems** container in the Results Tree for the current output database.

2. In the **Create Coordinate System** dialog box:
   a. Accept the default name for the coordinate system, or enter the name of your choice in the text field.
   b. Toggle on **Fixed system** to specify the motion of the coordinate system.
   c. Select one of the following as the coordinate system type:
      - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Cylindrical**: The *R*-, θ-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Spherical**: The *R*-, θ-, and ϕ-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
   d. Click **Continue**.

   Abaqus/CAE displays prompts in the prompt area to help you define the coordinate system axes.
3. Select a node in the viewport to be the origin; or enter the *X*-, *Y*-, and *Z*-coordinates of the origin in the prompt area.
4. Select a node in the viewport to lie on the *X*-axis (for a rectangular coordinate system) or the *R*-axis (for a cylindrical or spherical coordinate system); or enter the *X*-, *Y*-, and *Z*-coordinates of the point in the prompt area.
5. Select a node in the viewport to lie in the *X*–*Y* plane (for a rectangular coordinate system) or the *R*–θ plane (for a cylindrical or spherical coordinate system); or enter the *X*-, *Y*-, and *Z*-coordinates of the point in the prompt area.

   The coordinate system appears in the viewport, in the list in the **Coordinate System Manager**, and under the **Session Coordinate Systems** container in the Results Tree for the current output database. Abaqus saves user-defined coordinate systems only until the output database is closed unless you save them to the output database file. You cannot edit a coordinate system once it has been created; you can rename or delete coordinate systems that have not been saved to the output database.

Additional information

- *Saving a coordinate system to an output database file*
- *Controlling the display of model entities*

Creating a coordinate system following three nodes

You can define a coordinate system that is attached to nodes on your model by specifying three nodes. The line from node 1 (the origin) to node 2 is the *X*-axis (for a rectangular coordinate system) or the *R*-axis (for a cylindrical or spherical coordinate system); the plane containing the three nodes is the *X*–*Y* plane (for a rectangular coordinate system) or the *R*–θ plane (for a cylindrical or spherical coordinate system). The motion of the coordinate system is defined by the translational motion of the three nodes.

The visual display of coordinate systems that follow nodes on your model is updated to reflect any deformation scale factors applied; however, result transformations based on these coordinate systems use the actual, not the scaled, deformations.

1. From the main menu bar, select **Tools->Coordinate System->Create**.

   The **Create Coordinate System** dialog box appears.

   Tip: You can also create a coordinate system by using the  tool in the toolbox or the **Session Coordinate Systems** container in the Results Tree for the current output database.

2. In the **Create Coordinate System** dialog box:
   a. Accept the default name for the coordinate system, or enter the name of your choice in the text field.
   b. Toggle on **System following 3 nodes** to specify the motion of the coordinate system.
   c. Select one of the following as the coordinate system type:
      - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Cylindrical**: The *R*-, θ-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Spherical**: The *R*-, θ-, and ϕ-axes are interpreted as the 1-, 2-, and 3-axes, respectively.

   d. Click **Continue**.

   Abaqus/CAE displays prompts in the prompt area to help you define the coordinate system axes.
3. Select a node in the viewport to be the origin.
4. Select a node in the viewport to lie on the *X*-axis (for a rectangular coordinate system) or the *R*-axis (for a cylindrical or spherical coordinate system).
5. Select a node in the viewport to lie in the *X*–*Y* plane (for a rectangular coordinate system) or the *R*–θ plane (for a cylindrical or spherical coordinate system).

   The coordinate system appears in the viewport, in the list in the **Coordinate System Manager**, and under the **Session Coordinate Systems** container in the Results Tree for the current output database. Abaqus saves user-defined coordinate systems only until the output database is closed unless you save them to the output database file. You cannot edit a coordinate system once it has been created; you can rename or delete coordinate systems that have not been saved to the output database.

Additional information

- *Saving a coordinate system to an output database file*
- *Controlling the display of model entities*

Creating a coordinate system following three nodes on a circle

You can select three nodes that lie on a circular arc to define a coordinate system that is attached to your model. This method is useful when the model does not contain any nodes along the axis of a hollow cylinder or at the center of a hollow sphere, but you would like to define a coordinate system at those locations. The center of the circle defined by the three nodes is the origin of the coordinate system. The line from the origin to node 1 defines the *X*-axis (for a rectangular coordinate system) or the *R*-axis (for a cylindrical or spherical coordinate system); thus, the order in which you select the nodes determines the direction of increasing θ. Three colinear nodes are not valid. The normal to the circle is parallel to the *Z*-axis (for a rectangular or cylindrical coordinate system) or the ϕ-axis (for a spherical coordinate system). The motion of the coordinate system is defined by the translational motion of the three nodes.

The visual display of coordinate systems that follow nodes on your model is updated to reflect any deformation scale factors applied; however, result transformations based on these coordinate systems use the actual, not the scaled, deformations.

1. From the main menu bar, select **Tools->Coordinate System->Create**.

   The **Create Coordinate System** dialog box appears.

   Tip: You can also create a coordinate system by using the  tool in the toolbox or the **Session Coordinate Systems** container in the Results Tree for the current output database.

2. In the **Create Coordinate System** dialog box:
   a. Accept the default name for the coordinate system, or enter the name of your choice in the text field.
   b. Toggle on **System following 3 nodes on a circle** to specify the motion of the coordinate system.
   c. Select one of the following as the coordinate system type:
      - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Cylindrical**: The *R*-, θ-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Spherical**: The *R*-, θ-, and ϕ-axes are interpreted as the 1-, 2-, and 3-axes, respectively.

   d. Click **Continue**.

   Abaqus/CAE displays prompts in the prompt area to help you define the coordinate system axes.

3. Select three nodes in the viewport that lie on a circular arc.

   The coordinate system appears in the viewport, in the list in the **Coordinate System Manager**, and under the **Session Coordinate Systems** container in the Results Tree for the current output database. Abaqus saves user-defined coordinate systems only until the output database is closed unless you save them to the output database file. You cannot edit a coordinate system once it has been created; you can rename or delete coordinate systems that have not been saved to the output database.

Additional information

- *Saving a coordinate system to an output database file*
- *Controlling the display of model entities*

Creating a coordinate system following a single node

You can define a coordinate system that is attached to a single node on your model. The specified node is the origin of the coordinate system. The degrees of freedom present at the node (translational and rotational) determine the orientation of the coordinate system axes at each step and frame of the analysis. If the node has only translational degrees of freedom, the coordinate system will always be parallel to the global system.

The visual display of coordinate systems that follow nodes on your model is updated to reflect any deformation scale factors applied; however, result transformations based on these coordinate systems use the actual, not the scaled, deformations.

1. From the main menu bar, select **Tools->Coordinate System->Create**.

   The **Create Coordinate System** dialog box appears.

   Tip: You can also create a coordinate system by using the  tool in the toolbox or the **Session Coordinate Systems** container in the Results Tree for the current output database.

2. In the **Create Coordinate System** dialog box:
   a. Accept the default name for the coordinate system, or enter the name of your choice in the text field.
   b. Toggle on **System following a single node** to specify the motion of the coordinate system.
   c. Select one of the following as the coordinate system type:
      - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Cylindrical**: The *R*-, θ-, and *Z*-axes are interpreted as the 1-, 2-, and 3-axes, respectively.
      - **Spherical**: The *R*-, θ-, and ϕ-axes are interpreted as the 1-, 2-, and 3-axes, respectively.

   d. Click **Continue**.

   Abaqus/CAE displays prompts in the prompt area to help you define the coordinate system axes.

3. Select the node in the viewport.

   The coordinate system appears in the viewport, in the list in the **Coordinate System Manager**, and under the **Session Coordinate Systems** container in the Results Tree for the current output database. Abaqus saves user-defined coordinate systems only until the output database is closed unless you save them to the output database file. You cannot edit a coordinate system once it has been created; you can rename or delete coordinate systems that have not been saved to the output database.

Additional information

- *Saving a coordinate system to an output database file*
- *Controlling the display of model entities*

Saving a coordinate system to an output database file

Coordinate systems defined during postprocessing are available by default only until the associated output database file is closed. You must save these coordinate systems to the output database file if you want them to be available in future Abaqus/CAE sessions. Local coordinate systems defined during the analysis are written automatically to the output database. You can only append data to the output database; you cannot modify the contents of the file in any other way (i.e., once a coordinate system has been saved to the output database, it cannot be modified, renamed, or deleted).



Note:

By default, Abaqus/CAE opens output database files as read-only. You must choose to open an output database file with write privileges if you want to save any data, such as coordinate systems, to it (see *Opening a model database or an output database*, for more information).

1. From the main menu bar, select **Tools->Coordinate System->Manager**.

   The **Coordinate System Manager** appears. By default, **Current session** is toggled on and the coordinate systems that have been defined for the current output database during the current session are listed.

2. Select the user-defined coordinate systems that you wish to save, and click **Move to ODB** from the buttons on the right side of the dialog box.

   If the current output database file was opened with write permissions, the selected coordinate systems are moved from the current session list to the current output database list. To see the list of coordinate systems that have been saved to the current output database file, toggle on **Current ODB**.

Creating node and element sets during postprocessing

You can create node and element sets in the Visualization module.
Select **Tools->Create Set** from the main menu bar. You can create and save sets only when the output database (`.odb`) file is opened with write privileges.

In addition, the following rules apply:

- You can create sets only on the rootAssembly instance; however, the instance cannot be named `ASSEMBLY`.
- You cannot edit or delete the sets created in the output database.

1. From the main menu bar in the Visualization module, select **Tools->Create Set**.
2. In the **Create Set** dialog box that appears, do the following:
   a. Enter a name for the set.
   b. Select **Node** or **Element** as the set type.
   c. Click **Continue**.
   d. Select nodes or elements from the viewport. You can select the nodes and elements individually, by angle, or by feature edge. You can also select elements by topology.
   e. Click **Done**.

   The newly created set appears in the Results Tree.

- *Opening a model database or an output database*
- *Using the angle and feature edge method to select multiple objects*
- *Using the topology method to select multiple elements*

Plotting the undeformed and deformed shapes

This section explains undeformed and deformed shape plotting and superimposed plots.

Immediately upon opening an output database, Abaqus/CAE displays the undeformed shape of your model. For general analysis steps an undeformed plot displays the shape of your model without any deformation. For linear perturbation steps an undeformed plot displays the base state of your model. The deformed plot shows the shape of your model according to the values of a nodal variable such as displacement. The undeformed and deformed shapes are also used to display each of the other plot states available in Abaqus/CAE. You can view contours, symbols, or material orientations on either the undeformed or the deformed shape or superimpose them to view the differences in the shapes.

In this section:

- *Understanding undeformed shape plotting*
- *Understanding deformed shape plotting*
- *Using common plot options*
- *Producing an undeformed shape plot*
- *Producing a deformed shape plot*
- *Superimposing deformed and undeformed model plots*

Understanding undeformed shape plotting

Abaqus/CAE obtains the information needed to produce an undeformed plot from the output database. Immediately upon opening an output database, Abaqus/CAE displays the undeformed plot state of your model.

You can use the plot state-independent options to customize your plot. Select **Options->Common** in the main menu bar to choose the plot state-independent customization options.

By default, Abaqus/CAE displays your undeformed model at the last frame of the last step available in the output database. If your model varies between steps, perhaps because of the addition or removal of contact surfaces or the application of loads or boundary conditions, you may want to view your undeformed model at steps other than the default. You can choose the step at which to view your undeformed model by selecting **Result->Step/Frame** from the main menu bar. For more information, see *Selecting the results step and frame*.

You can also use Abaqus/CAE to combine your undeformed and deformed model shapes into a single plot. For more information, see *Superimposing deformed and undeformed model plots*. Combining the shapes provides a context for evaluating the deformations.

Additional information

- *Producing an undeformed shape plot*

Understanding deformed shape plotting

A deformed plot shows the shape of your model according to the values of a nodal variable such as displacement. You can choose the nodal variable (called the deformed field output variable) for which Abaqus/CAE displays results, and you can choose the step and frame of these results.

If you do not choose a deformed field output variable, Abaqus/CAE attempts to select a default. Most procedures in Abaqus/Standard and Abaqus/Explicit write displacement to the output database by default; in these cases Abaqus/CAE selects displacement as the nodal vector quantity to use for the default deformed variable. Some procedures—for example, heat transfer—do not write displacement to the output database by default; therefore, Abaqus/CAE does not select a default deformed variable. Abaqus/CAE cannot display a deformed plot if the output database does not contain any variables that can be used to compute a deformed shape.

After obtaining the information needed to produce a deformed plot from the output database, Abaqus/CAE computes the shape of your deformed model by adjusting the coordinates of each node according to:

- The deformed field output variable (see *Selecting the deformed field output variable*).
- The analysis step and frame (see *Selecting a specific results step and frame*).
- Uniform or nonuniform deformation scale factors (see *Scaling deformations*).
- Any user-defined rigid body transformation applied to the deformed field output variable (see *Transforming results into a new coordinate system*).

Like the undeformed plot shape, you can use the plot state–independent options to customize your deformed shape plot. Select **Options->Common** in the main menu bar to choose the plot state–independent customization options.

You can also use Abaqus/CAE to combine your undeformed and deformed model shapes into a single plot. Combining the shapes provides a context for evaluating the deformations. If you choose to superimpose your undeformed and deformed model shapes, you can use the **Superimpose** plot options to control the display of the undeformed shape separately from the deformed shape.



Tip: Use the **Allow Multiple Plot States**  tool to display both model shapes if you want to display one or both shapes without contours, symbols, or material orientations. You can also use this tool to select any combination of plot states from the tools in the Visualization module toolbox (see *Displaying multiple plot states*, for more information).

Using common plot options

You can use the common plot options to customize the appearance of undeformed and deformed plots.

The common options are also used in conjunction with the other plot state–independent options in the **View** menu and the dependent and independent customization options for contour, symbol, and material orientation plots. Select **Options->Common** from the main menu bar or use the  tool in the toolbox to access the **Common Plot Options** dialog box. Click the following tabs to customize the appearance of plots in the current viewport:

- **Basic**: Choose render style, edge visibility, and deformation scale factor (deformed plot only).
- **Color & Style**: Control model edge color and style, model face color, edge style, and edge thickness.
- **Labels**: Control element, face, and node labels and node symbols; and control probe annotation labels.
- **Normals**: Control element and surface normals.
- **Other**: The **Other** page contains the following tabs:
  - **Scaling**: Control model scaling and shrinking.
  - **Translucency**: Control shaded and filled render style translucency.

If you choose to superimpose the undeformed and deformed model shapes, the common options control the display of the deformed shape and the superimpose plot options control the display of the undeformed shape (for more information, see *Superimposing deformed and undeformed model plots*). Plot state–dependent options may override some common options; for example, the color options for contour plots will always override the common color options.

To learn how to customize the render style and other display characteristics of your plots, see *Customizing plot display*.

Producing an undeformed shape plot

For general analysis steps an undeformed plot displays the shape of your model without any deformation. For linear perturbation steps an undeformed plot displays the base state of your model. Since analysis results are not needed; an output database resulting from a **datacheck** run is sufficient to produce an undeformed shape plot. Model and analysis characteristics such as surface definitions can cause the appearance of your undeformed model to vary from step to step; you can select the step and frame at which to display your undeformed plot. For more information, see *Selecting the results step and frame*.

By default, when you *open an output database*, Abaqus/CAE displays the undeformed plot state for the last step and frame of the analysis. The plot is produced using the default display options for the current session (for information on saving display options, see *Saving your display options settings*). To produce an undeformed shape plot for an output database that is already open, select **Plot->Undeformed Shape** from the main menu bar or use the  tool in the toolbox.

1. Use the **File** menu to *open the output database* containing your model data.
2. Use the **Result** menu to select the *results step* to display.
3. Select the plot state–independent *customization options* that you want.

Abaqus automatically refreshes your undeformed plot each time you click **Apply** in the results or plot state–independent dialog boxes.

Producing a deformed shape plot

A deformed plot displays the shape of your model at a specified step and frame of the analysis results, according to the values of the deformed field variable. For more information on selecting a step and frame, see *Selecting a specific results step and frame*. For more information on selecting a deformed field variable, see *Selecting the deformed field output variable*. To learn how to display the minimum and maximum values associated with your deformed plot, see *Customizing the legend*.

1. Use the **File** menu to *open the output database* containing your analysis results.
2. Use the **Result** menu to select the following:
   a. The step and frame at which Abaqus obtains values.
   b. The deformed field variable for which Abaqus obtains values. The variables that have been saved during the analysis are available.
   c. The numeric form in which Abaqus displays complex number results.
3. Select the plot state–independent *customization options* that you want.
4. From the main menu bar, select **Plot->Deformed Shape** to display the deformed plot.



Tip: You can also produce a deformed shape plot using the  tool in the toolbox.

The current viewport displays a customized deformed plot of the specified deformation field variable at the specified step and frame of the output database. The state block, if active, changes to show the name of the variable displayed.

Abaqus automatically refreshes your deformed plot each time you click **Apply** in the results or plot state–independent dialog boxes.

Additional information

- *Producing an undeformed shape plot*
- *Using common plot options*

Superimposing deformed and undeformed model plots

You can combine the deformed and undeformed model shapes in a single plot. Combining the shapes provides a context for displaying and interpreting the contours, symbols, or material orientations. An example of superimposed model shapes is shown in *Figure 1*.

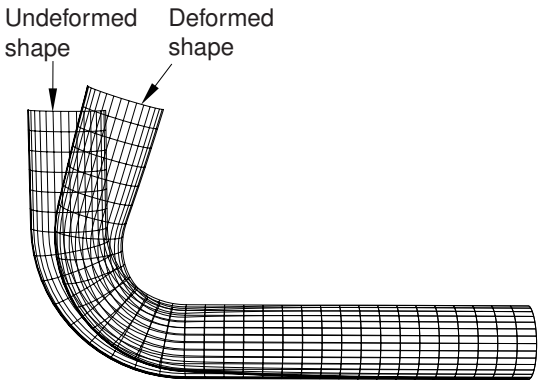


Figure 1: Deformed shape superimposed on the undeformed shape.

To produce a superimposed plot, select **Plot->*Contours, Symbols, or Material Orientations*->On Both Shapes** from the main menu bar or use the , , and tools in the toolbox. To superimpose the undeformed and deformed shapes without contours, symbols, or material orientations or to display any combination of plot types for the same results, use the tool and select all of the desired plot types from the toolbox (for more information, see *Displaying multiple plot states*).

1. From the main menu bar, select **Options->Superimpose** to customize the appearance of the undeformed shape.

   Tip: You can also customize the undeformed plot shape using the tool in the toolbox.

2. Click the following tabs to modify the options:

   - **Basic**: Choose render style and edge visibility.
   - **Color & Style**: Control model edge color and style, model face color, edge style, and edge thickness.
   - **Labels**: Control element, face, and node labels and node symbols.
   - **Normals**: Control element and surface normals.
   - **Other**: The **Other** page contains the following tabs:
     - **Scaling**: Control model scaling and shrinking.
     - **Translucency**: Control shaded and filled render style translucency.
     - **Offset**: Control the offset between the undeformed and deformed shapes.

     For more detailed information on customizing the display characteristics of your plots, see *Customizing plot display*.

   Abaqus/CAE uses the superimpose plot options in place of the common plot options to display the undeformed plot in the viewport.

3. To customize the similar options for the deformed shape, select **Options->Common** or use the  tool in the toolbox.
4. If contours, symbols, or material orientations are active, use the associated plot state–dependent options to customize the display of both the undeformed and the deformed plot.
5. Click **Apply** in each options dialog box to implement your changes in the viewport.

Your superimposed plot options changes are saved for the duration of the session and will affect the undeformed shape in all subsequent plots where both model shapes are displayed.

Additional information

- *Producing an undeformed shape plot*
- *Producing a deformed shape plot*
- *Using common plot options*
- *Customizing plot display*

Contouring analysis results

This section explains contour plotting.

A contour plot displays the values of an analysis variable at a specified step and frame. In addition, a contour plot can be used to show the value of attributes such as loads or predefined fields at a specified step of a model in the current model database. Abaqus/CAE represents the values as customized colored lines, colored bands, colored faces, colored isosurfaces, or tick marks on your model.

In this section:

- *Understanding contour plotting*
- *Using contour plot options*
- *Producing a contour plot*
- *Producing a contour plot of linear beam section stresses*
- *Customizing a contour plot*

Understanding contour plotting

This section discusses the computation and rendering of contour values and the computation of contour limits.

In this section:

- *About contour plotting*
- *Understanding how contour values are computed*
- *Understanding how contours are rendered*
- *Understanding contour limits*

About contour plotting

A contour plot displays results that are stored in the output database or displays model attributes from a model in the current model database.
Each contour plot displays the following values: for an output database, contour plots display the value of a particular field output variable in a specified step and frame of the analysis; for a model in the current model database, contour plots display the value of a selected attribute such as a load in a specified step in your analysis. These values are shown as colored lines, colored bands, or quilt-type colored faces on the surface of your model, or they are shown as isosurfaces that extend line contours into the model, depending on the customization options you select. A contour plot cannot be produced if no elements are visible in the current viewport.

Line, banded, quilt-type, and isosurface contours are shown in *Figure 1*.

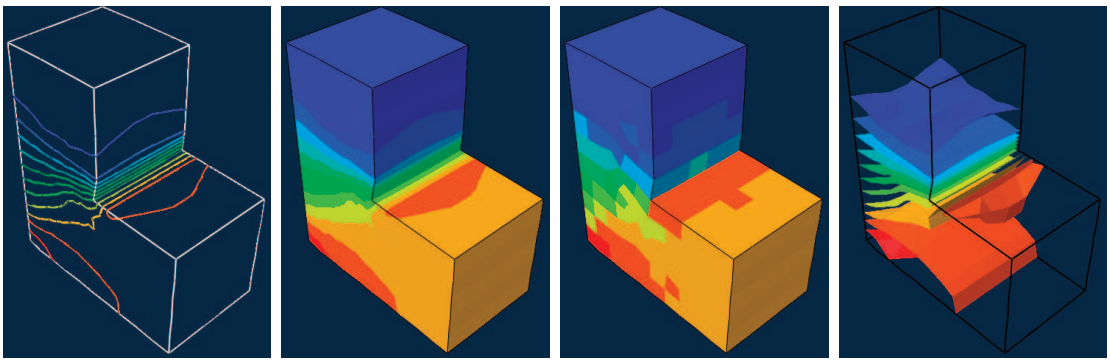


Figure 1: From left to right: contours shown as colored lines, colored bands, colored faces, and colored isosurfaces.

As with other elements, contour lines for line-shaped elements (beams, one-dimensional elements, gasket link elements, and three-dimensional line gasket elements, as well as two-dimensional contact surfaces) are plotted along the elements by default. Line-type contour plots are not recommended for line-shaped elements. Tick mark contour plotting provides an alternative means of visualizing contours on beams and other line-shaped elements. The contour is displayed as a curve plotted between two sets of lines normal to the elements, as shown in *Figure 2*. The contour level is indicated by “tick marks” on these normal lines.

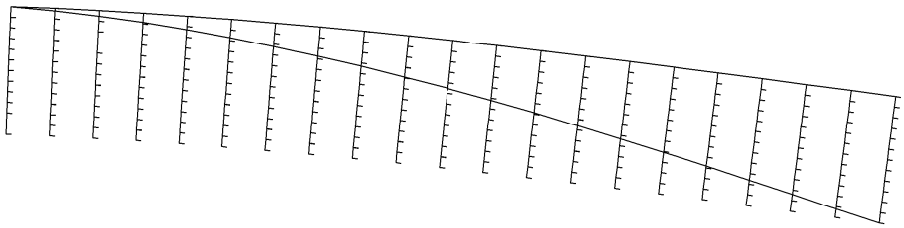


Figure 2: Tick mark contour plot.

The key to interpreting a contour plot is the plot legend. The legend indicates the correspondence between contour values and contour colors.

By default, Abaqus/CAE evenly divides the difference between the minimum and maximum values in the legend into 12 intervals. You can change the number of intervals if necessary. A color is associated with each interval. For a line-type or isosurface-type contour plot each colored line or surface corresponds to a set of locations in the model for

which the field output variable has the value shown in the legend. For a banded contour plot each colored contour band corresponds to a range of values within the bounds indicated by the legend. For a quilt contour plot each colored element face corresponds to a single value within the bounds indicated by the legend for that color. For a tick mark contour plot each colored tick mark corresponds to a single value indicated by the legend.

Abaqus/CAE follows a different convention for contour plots of contact status (CSTATUS). This variable describes whether individual nodes on the main and secondary surfaces are in contact and, if so, whether the surfaces are sticking or slipping at that point. Because three settings are available to describe the contact status at each node—sticking, slipping, or open—Abaqus/CAE displays contact status data using three contour intervals and renders the contour plot by coloring the region around a node according to its contact status.

The components of connector vector output are resolved with respect to different coordinate systems depending on whether the output requested is field output or history output. See *Displaying connectors and connector output in the Visualization module* for how to interpret the vector component values in field plots.

To learn how to produce a contour plot, see *Producing a contour plot*.

Understanding how contour values are computed

The computations necessary to produce a contour plot from results stored in the output database depend on whether the contour plot displays values for a node-based quantity, such as displacement or velocity, or for an element-based quantity, such as stress or strain. This section applies to contour plots of output database data only.

How node-based field output variable contour values are computed

For contour plots of node-based field output variables, contour values are obtained directly from the quantities on the output database. These values are then used without further computation to produce colored lines or colored bands on the surface of your model.

How element-based field output variable contour values are computed

For contour plots of element-based field output variables, Abaqus/CAE applies computations to the output database results to form the contour values. The computations vary according to the following criteria:

- the quantity you choose to plot (field output or discontinuities),
- the averaging options you select,
- the *result regions* you select,
- the compatibility of elements included in the plot, and
- the type of contour plot you request (line, banded, or quilt).

Select **Result->Field Output** from the main menu bar to choose the field output variable. To learn more about this topic, see *Selecting the field output to display*.

Select **Result->Options** from the main menu bar to choose the quantity to plot, the averaging options, the result regions, and to control the handling of incompatible elements during the computation of element-based contour values. To learn more about these topics, see *Selecting result options*.

Select **Options->Contour->Basic** from the main menu bar to choose the type of contour plot that you want. *Choosing line-, banded-, quilt-, or isosurface-type contours*, contains detailed instructions on choosing the contour type.

The type of contour plot that you request governs the extrapolation applied to the results read from the output database. For line-, banded-, and isosurface-type contours of element-based variable values, Abaqus/CAE extrapolates results to the nodes. By default, the results are conditionally averaged after Abaqus/CAE performs any necessary transformations and computes any requested scalar components or invariants; if desired, you can compute invariants after the results are averaged. For quilt-type contours of element-based variable values, Abaqus/CAE extrapolates results to the element faces on the surface of your model and then takes a weighted sum to produce a single value per face. Since quilt contour values are computed for each element face individually with no averaging across element boundaries, a quilt contour plot is an effective means of displaying results on an element-by-element basis. You can choose quilt-type contours only for element-based field output variables.

Understanding how contours are rendered

Contour values are drawn on element faces using one of two methods: texture mapping or tessellation. Texture mapping is a high-performance rendering method that essentially lays an image of the contour values (the texture) over an image of the model, similar to the process of wrapping a present. Tessellation is a method of transforming arbitrary contour values into repeating patterns of distinct shapes, such as triangles or simple polygons; the shape values are computed face by face and can take a long time for large models. Texture mapping is the default and preferred method and will maximize the performance of Abaqus/CAE. Although certain graphics adapters do not support texture mapping properly, Abaqus/CAE can emulate texture mapping in software. However, line-type and isosurface contours, the display of contour edges, shrinking elements about their centroid, and contours on CAXA or SAXA elements are supported only by the tessellation method.

The differences between contour plots generated with texture mapping or tessellation are minimal; you may see slight differences in the appearance of contour bands on quadrilateral element faces. Texture mapping can sometimes introduce precision issues; for example, if the contour value is exactly on a limit, the color may appear as just above or below the value.



Note:

If you display tessellated contours under intense lighting, the colors in the contours might not match the colors displayed in the legend because the light can fade some colors on the surface facets. For more information, see *Controlling model lighting*.

For detailed information on choosing the contour method, see *Choosing the contour method*.

Understanding contour limits

The minimum and maximum values shown in a contour plot can be computed by Abaqus/CAE, or you can specify them.
You can decide to specify one or both limits; for example, to eliminate extremes or to examine variations within a fixed set of bounds.

Contours are shown on visible surfaces of your model only; however, depending on the type of contour plot and the result options that you select, Abaqus/CAE might computer contour values based on all elements in the current display group or on all elements in the model. (For more information, see *Selecting result options*.) The reported minimum and maximum values might, therefore, occur at interior elements. You can display contours on the interior of your model by using display groups to show only the interior elements in the plot or by shrinking all elements about their centroids so that interior elements are visible between the bodies of exterior elements.

Abaqus/CAE computes the minimum and maximum values differently, according to the type of field output variable shown (nodal or element), the type of contour plot (quilt versus line or banded), and options you select. For contour plots of nodal field output variables, the limits are the maximum and minimum nodal values. For quilt-type contour plots, the limits are the maximum and minimum element face values. For line-, banded-, or isosurface-type contours of element field output variables, Abaqus/CAE determines the minimum and maximum by the extrapolated, averaged values as defined by the current averaging criteria. For element field output variables, the contour maximum and minimum values can be given at element faces, integration points, and element centroids as well as at nodes.

When Abaqus/CAE computes the minimum and maximum over values averaged at the nodes, the limits are based on the values used to create the plot. In this case, the legend bounds may vary as you vary the averaging criteria. To learn how to control the averaging criteria, see *Controlling result averaging*.

You can also control which contour limits are used for each frame of a contour plot animation: the limits computed for the first and last frame, the limits computed for the current frame, the recomputed limits for each individual frame, or the limits computed for all frames in the animation. Computing limits based on all frames in the animation requires one full pass through the animation sequence to establish the contour limits used for subsequent passes through the animation.

For detailed information on controlling contour limits, see *Setting contour limits*. To learn how to display the minimum and maximum values associated with your contour plot, see *Customizing the legend*.

Using contour plot options

You can use the contour plot options to customize the appearance of contour plots.

Select **Options->Contour** from the main menu bar or click  in the toolbox to access the **Contour Plot Options** dialog box. Click the following tabs to customize the appearance of contour plots in the current viewport:

- **Basic**: Choose the contour type (including whether to show tick marks for line elements), contour intervals, and contour method.
- **Color & Style**: The **Color & Style** page contains the following tabs:
  - **Model Edges**: Control the color of model edges.
  - **Spectrum**: Choose contour colors, and control the order of the values displayed in the legend.
  - **Line**: For line-type contours control the style and thickness of each line.
  - **Banded / Isosurface**: For banded- or isosurface-type contours control the color, style, and thickness of contour edges.
- **Limits**: Control the computation of contour limits and the display of annotations for the maximum and minimum contour values.
- **Other**: Control tick mark plot display options, the display of nodal averaged vector or tensor orientations, and the display of section point or envelope plots.

Other options such as the render style, edge visibility, deformation scale factor, and translucency are located in the ***Common Plot Options*** dialog box. If you choose to plot contours on both the undeformed and the deformed shape, you can use the ***Superimpose Plot Options*** to customize the appearance of the undeformed shape. See *Superimposing deformed and undeformed model plots*, for more information about superimpose plots. To learn how to customize the render style and underlying model of your contour plot, see *Customizing plot display*. For information on the computation of result values, see *Understanding how results are computed*.

Additional information

- *Customizing a contour plot*

Producing a contour plot

A contour plot can display either the values of an analysis variable at a specified step and frame of an output database or the values of selected attributes at a specified step of a model in the current model database. Abaqus/CAE represents the values as customized colored lines, colored bands, colored faces, or colored isosurfaces on your model or as colored tick marks on lines drawn normal to your model. For more information on selecting an analysis variable, see *Selecting the primary field output variable*. For more information on selecting a specific step and frame, see *Selecting a specific results step and frame*. To learn how to display the minimum and maximum values associated with your contour plot, see *Customizing the legend*.

If your model includes beam elements, some additional steps can be required for proper contour display. See *Producing a contour plot of linear beam section stresses*.

If you are plotting data from a model database, specify only the primary field variable.

1. Use the **File** menu to *open the model database or output database* containing your model data or analysis results.
2. If you are plotting data from a model database, switch to the Visualization module, expand the **Model Database** container of the Results Tree, and select the model you want to use.
3. Use the **Result** menu to select the following:
   a. The *step and frame* to display (or just the step, if a model database is selected).
   b. The *primary field* output variable to display.
   c. The *deformed field* output variable to display (for contours on the deformed shape only).
   d. The *quantity to plot* and *averaging* options (for output databases only).
4. Select the plot state–independent and contour plot *customization options* that you want.
5. If your chosen field output variable includes complex numbers, select the ***Complex Form*** tab in the **Result Options** dialog box to control the numeric form to display. Complex form options apply only to output databases.
6. From the main menu bar, select **Plot->Contours** and choose whether to plot the contours on the undeformed shape, the deformed shape, or both.

   Tip: You can also produce a contour plot using the deformed , undeformed , or superimposed  contour tools in the toolbox.

   The current viewport displays a customized contour plot of the specified field output variable at the specified step and frame of the current output database or at the specified step of the selected model in the current model database.

   Abaqus automatically refreshes your contour plot each time you click **Apply** in the step and frame selector, field output options, plot state–independent options, superimpose plot options (if applicable), or contour plot options dialog boxes.

Additional information

- *Using contour plot options*
- *Selecting model data and analysis results to plot*

Producing a contour plot of linear beam section stresses

If your model includes beam geometry and beam rendering is enabled for your session, you can display field output results along the beam cross-sections as contours. For nodal quantities and element integration point quantities, Abaqus/CAE displays contours that are constant through the thickness of the beam cross-section. For the element-based data, the contour reflects the currently selected section point.

You can also create a contour plot that shows a more realistic stress distribution through the beam based on section force and section moment data, which Abaqus/CAE calculates using linear elastic solid mechanics theory. *Figure 1* shows an example of beam stresses through an I-beam.

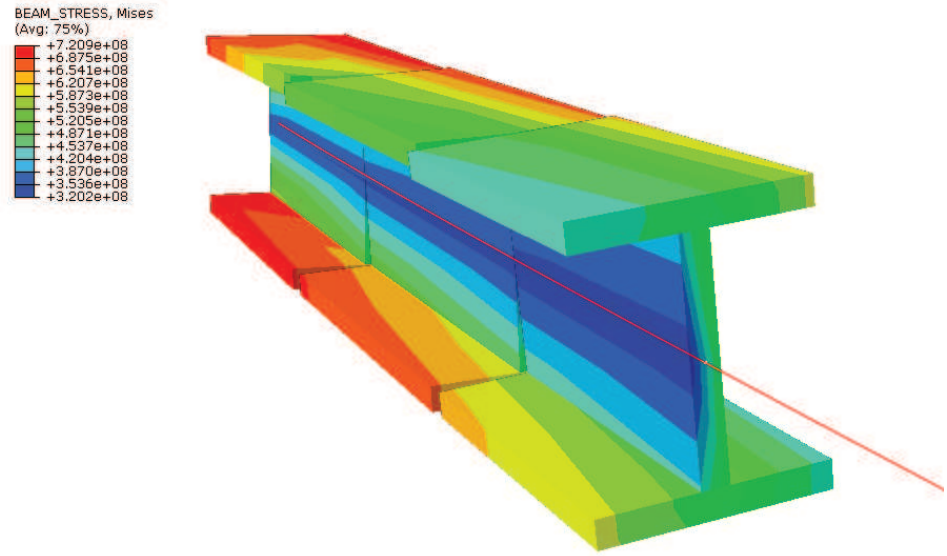


Figure 1: Linear beam section stresses displayed for an I-beam cross-section.

Beam stress contour plots are available only when the selected step and frame include data from the integrated output quantities SF (section force) and SM (section moment). Furthermore, these plots are available only for a subset of profiles in Abaqus/CAE:

- Thin-walled box profiles
- Thin-walled pipe profiles
- Circular profiles
- Rectangular profiles
- I-shaped profiles
- L-shaped profiles
- T-shaped profiles

BEAM_STRESS output is not available for random response and response spectrum procedures.

Abaqus/CAE does not include beam element normals for eigenfrequency extraction in its calculations for beam cross section rendering, so you cannot visualize the torsional and out-of-plane modes for the beam elements in these plots. However, the torsional modes are calculated for the beam elements because the beam elements have torsional stiffness; this information is available in the data (`.dat`) file.

1. Use the **File** menu to *open the output database* containing your analysis results.

2. Use the **Result** menu to select the following:
   a. The *step and frame* to display.
   b. The *primary field* output variable to display.
   c. The *deformed field* output variable to display (for contours on the deformed shape only).
   d. The *quantity to plot* and *averaging* options.
3. Select the plot state–independent and contour plot *customization options* that you want.
4. From the **Output Variable** options, select **BEAM_STRESS**.
5. Choose the invariant or component for which you want to display contour data:
   - Select **Mises** from the **Invariant** options to display von Mises stress.
   - Select **S11** from the **Component** options to display axial stress along the beam.
   - Select **S12** from the **Component** options to display shear stress along the local beam section 2-axis caused by shear force and torsion.
   - Select **S13** from the **Component** options to display shear stress along the local beam section 1-axis caused by shear force and torsion.
6. From the main menu bar, select **Plot->Contours** and choose whether to plot the contours on the undeformed shape, the deformed shape, or both.

   Tip: You can also produce a contour plot using the deformed , undeformed , or superimposed contour tools in the toolbox.

   The current viewport displays a customized contour plot of the specified field output variable at the specified step and frame of the current output database.

   Abaqus automatically refreshes your contour plot each time you click **Apply** in the step and frame selector, field output options, plot state–independent options, superimpose plot options (if applicable), or contour plot options dialog boxes.

Additional information

- *Using contour plot options*
- *Selecting model data and analysis results to plot*
- *Producing a contour plot of linear beam section stresses*

Customizing a contour plot

This section describes how to customize a contour plot. Abaqus/CAE offers numerous options for you to control the contour type, limits, intervals, and colors.

To learn how to customize the render style, translucency, or model edge characteristics of your contour plot, see *Customizing plot display*. For information on adding nodal-averaged orientations to your contour plot, see *Displaying nodal-averaged orientations*. For information on customizing an envelope plot, see *Selecting section point data by category*. For information on the computation of result values, see *Understanding how results are computed*.

In this section:

- *Choosing line-, banded-, quilt-, or isosurface-type contours*
- *Stylizing line-type contours*
- *Customizing banded- and isosurface-type contours*
- *Contouring line-shaped elements with tick mark contour plots*
- *Customizing tick mark contour plots*
- *Choosing the contour method*
- *Setting contour limits*
- *Annotating the minimum and maximum contour values*
- *Controlling how Abaqus/CAE computes contour limits*
- *Customizing contour colors*
- *Creating a new color spectrum*
- *Customizing contour intervals*

Choosing line-, banded-, quilt-, or isosurface-type contours

You can choose **Line**-, **Banded**-, or **Isosurface**-type contours to display the values of nodal (such as displacement) or element (such as stress) field output variables. Line-type contours represent values as customized colored lines on the surface of your model. Banded contours represent values as color-filled bands. Isosurface contours extend line-type contours through the model. Line, banded, and isosurface contours of element-based variable values are computed by extrapolating results to the nodes and conditionally *averaging*. Averaging depends on the characteristics of your model and on options you select. Line-type contours are not recommended for line-shaped elements, such as beams.

You can choose **Quilt**-type contours only for element field output variables. Variable values are extrapolated to element faces on the surface of your model, with no averaging between elements. Quilt contours are useful for evaluating results on an element-by-element basis. Only one color per element is plotted for axisymmetric elements with asymmetric deformation (CAXA or SAXA).

Banded-type contours are the default. You must choose line-, banded-, or isosurface-type contours to produce a plot of:

- Nodal-based field output values.
- Element-based field output values averaged across elements.
- Element-based results discontinuities rather than field output values.

Figure 1 shows the appearance of line-, banded-, quilt-, and isosurface-type contour plots.

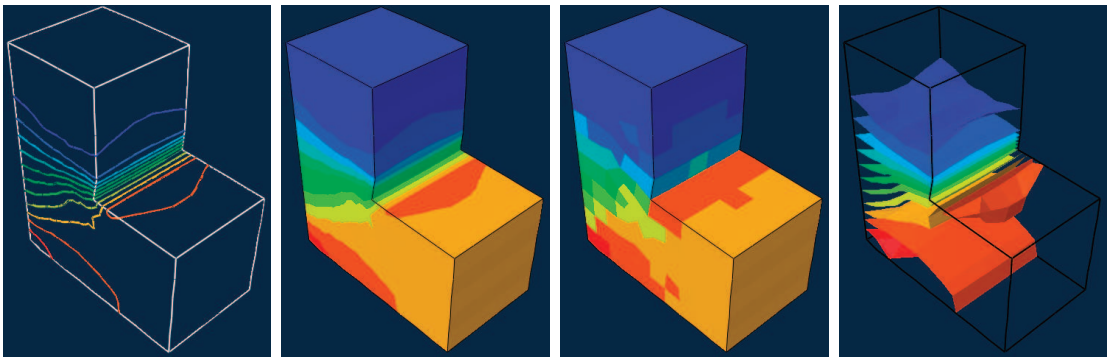


Figure 1: From left to right: line-, banded-, quilt-, and isosurface-type contours.

1. Locate the **Contour Type** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox; then click the **Basic** tab in the dialog box that appears. The **Contour Type** options are in the upper left corner of the page.
2. Click **Line**, **Banded**, **Quilt**, or **Isosurface** to choose the contour type you want.
3. Click **Apply** to implement your changes.

   The contour plot in the current viewport changes to display the contour type you have specified.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding how contour values are computed*
- *Stylizing line-type contours*
- *Customizing banded- and isosurface-type contours*

Stylizing line-type contours

Abaqus/CAE displays line-type contours as customized colored lines on the surface of your model. You can stylize individual contour lines to make them easy to distinguish.

This option is particularly useful when colors are not available; for example, for black and white hard-copy plotting. *Figure 1* shows a contour plot with customized contour lines.

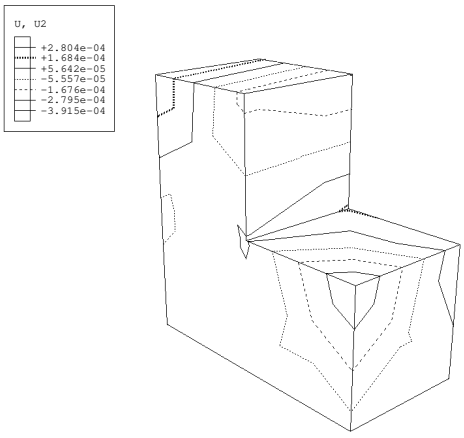


Figure 1: Line-type contour plot with customized contour lines.

You customize the colored lines one color *(interval)* at a time, referring to each interval by its position in the contour legend. Interval number 1 is the first colored line starting from the bottom of the contour plot legend. For information on the contour legend, see *Customizing the legend*.

1. Locate the **Line** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox. Click the **Color & Style** tab in the dialog box that appears; then click the **Line** tab. The **Line** options appear.

2. To choose the interval you want to customize, click the **Interval** arrows until the desired interval number appears in the **Interval** box.

   Note:

   The **Style** and **Thickness** buttons change to show the current style and thickness of this interval, respectively. The numbers above and below the **Interval** box change to correspond to the adjacent interval numbers. The area above and below the **Style** and **Thickness** buttons changes to show the style and thickness, respectively, of these adjacent intervals.

3. Choose the style of the interval line:
   a. Click the **Style** button to reveal the style choices.
   b. Click the style you want.

      The specified style appears on the **Style** button.

4. Choose the thickness of the interval line:
   a. Click the **Thickness** button to reveal the thickness choices.
   b. Click the thickness you want.

      The specified thickness appears on the **Thickness** button.

5. Repeat Steps 2 through 4 to customize additional intervals.

6. Click **Apply** to implement your changes.

   The contour lines of the line-type contour plot in the current viewport change to reflect your style and thickness specifications. The contour legend, if active, will also change to show your selections. For more information on the contour legend, see *Customizing the legend*. In addition, tick mark axes, if present, will change to show your selections. For more information on tick mark contours, see *Contouring line-shaped elements with tick mark contour plots*.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*
- *Customizing contour intervals*

Customizing banded- and isosurface-type contours

Abaqus/CAE displays banded-type contours as colored bands on the surface of your model and displays isosurface contours by extending line-type contours through the model to create surfaces. You can display or suppress lines at the edges of each contour band or isosurface; these lines are called contour edges. Contour edges visually separate the colors used to represent contour values in a banded-type contour plot; for isosurfaces, where the separation is often more distinct, the display of contour edges is more decorative. You can further customize the color, style, and thickness of contour edges. For example, in *Figure 1* the contour plot on the left has contour edges suppressed, while the plot on the right has contour edges displayed.

Note:

Contour edges are not applicable to *continuous interval* contour plots.

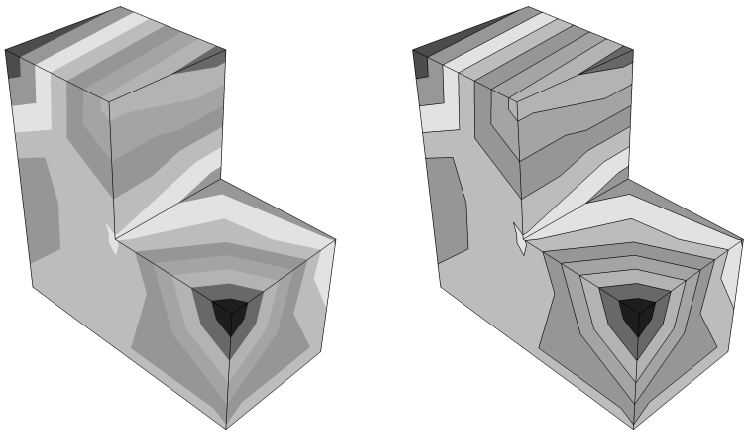


Figure 1: Banded-type contour plots with contour edges suppressed and displayed.

1. Locate the **Banded / Isosurface** options.

   Select **Options->Contour** from the main menu bar or click in the toolbox. Click the **Color & Style** tab in the dialog box that appears; then click the **Banded / Isosurface** tab. The **Banded / Isosurface** options appear.

2. Toggle **Show contour edges** to display or suppress contour edges.

   When **Show contour edges** is on, the contour edge options become available.

3. Choose the color of the contour edges:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Contour Plot Options** dialog box.

4. Choose the style of the contour edges:

   a. Click the **Style** button to reveal the contour edge style options.

   b. Click the edge style you want.

The specified edge style appears on the **Style** button.

5. Choose the thickness of the contour edges:
   a. Click the **Thickness** button to reveal the contour edge thickness options.
   b. Click the edge thickness you want.

      The specified edge thickness appears on the **Thickness** button.

6. Click **Apply** to implement your changes.

   The contour edges of the banded contour plot in the current viewport change to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Choosing line-, banded-, quilt-, or isosurface-type contours*
- *Customizing contour intervals*
- *Understanding contour plotting*

Contouring line-shaped elements with tick mark contour plots

Contour lines drawn on line-shaped elements are difficult to visualize clearly. To see the contours of line-shaped elements (beams, one-dimensional elements, gasket link elements, three-dimensional line gasket elements, and two-dimensional contact surfaces) better, you can choose to display the contour lines using tick mark contour plots.

A tick mark contour plot shows the contour curve drawn between sets of lines that are normal to the actual elements. The contour level is indicated by "tick marks" on these normal lines. Abaqus displays the contour curve only when it falls within the limits you have specified. For more information on limiting the range of the plot, see *Setting contour limits*.

Figure 1 shows the appearance of a tick mark contour plot.

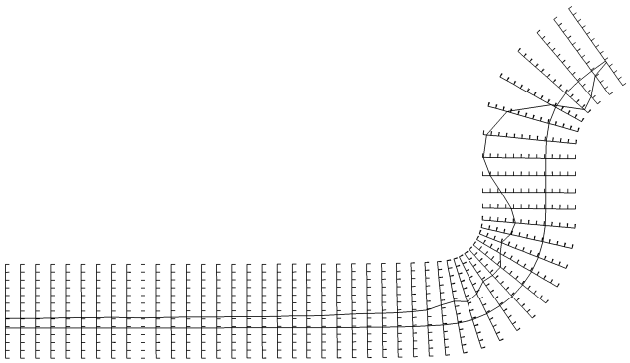


Figure 1: Tick mark contour plot.

The normal lines, or tick mark axes, are drawn at nodes. At part boundaries, two tick mark axes are drawn. Within a part, if the angle between two adjacent line elements is less than the specified feature angle, a single tick mark axis is drawn in the direction of the average normal of elements sharing the node. Otherwise, a tick mark axis is drawn normal to each element sharing the node. When averaging normals, only displayed elements of the same type are considered in the calculation.

1. Locate the **Contour Type** options.

   Select **Options->Contour** from the main menu bar or click in the toolbox; then click the **Basic** tab in the dialog box that appears. The **Contour Type** options are in the upper left corner of the page.

2. Toggle **Show tick marks for line elements**.
3. Click **Apply** to implement your changes.

   The contour plot in the current viewport changes to display tick mark plots for the line elements in the model.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing tick mark contour plots*

Customizing tick mark contour plots

When you choose to show tick marks for line-shaped elements in your model, Abaqus/CAE indicates the contour levels as tick marks on lines (axes) drawn normal to the elements and displays the contour curve between the axes. You can specify the length of the tick mark axes, the color of the contour curve, the contour value (tick mark) that will intersect the element, and the normal direction in which the axes will be drawn on beam elements. In addition, for line-type contour plots, style selections you apply to individual contour lines will appear on the tick mark axes. For more information, see *Stylizing line-type contours*.

1. Locate the **Tick Marks** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox. Click the **Other** tab in the dialog box that appears; then click the **Tick Marks** tab. The **Tick Marks** options appear.

2. To choose the normal direction in which the tick mark axes will be drawn on beam elements, toggle the **N1** or **N2** button in the **Orientation** field.

    **Note:**

   This option will be ignored when tick mark contour plots are displayed on line elements other than beams. In this case, the tick mark axes are drawn in the element normal direction.

3. Choose the length of the tick mark axes:
   a. Click the **Axis length** button to reveal the length choices: **Short**, **Medium**, or **Long**.
   b. Click the length you want.

      The specified length appears on the **Axis length** button.

4. Choose the contour value that will intersect the element:
   a. Enter the desired base contour value in the **Base Value** field.

      The value specified will be checked to determine whether it falls within the contour range. If it is above the maximum contour value, the maximum contour value will be used as the base value; if it is below the minimum contour value, the minimum contour value will be used as the base value. By default, zero is used as the base value.

5. Choose the color of the tick mark curve:
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Contour Plot Options** dialog box.

6. Click **Apply** to implement your changes.

   The contour lines of the tick mark contour plot in the current viewport change to reflect your specifications.

By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*

Choosing the contour method

Contour values are drawn on element faces using either the texture mapping or tessellation method. Texture mapping is the default and preferred method and will maximize the performance of Abaqus/CAE. Although certain graphics adapters do not support texture mapping properly, Abaqus/CAE can emulate texture mapping in software. However, certain display options are supported only by the tessellation method; in such cases Abaqus/CAE will override the user setting for texture-mapped contours automatically.

1. Locate the **Contour Method** options.

   Select **Options**->**Contour** from the main menu bar or click  in the toolbox; then click the **Basic** tab in the dialog box that appears. The **Contour Method** options are at the bottom of the page.

2. Choose one of the following rendering methods:

   - **Texture-mapped**. This option uses high-performance graphics rendering capabilities to draw contour values on each element face. If the graphics parameters for your system have been configured to disable texture mapping (see *Tuning graphics cards*), Abaqus/CAE will emulate texture mapping in software.
   - **Tessellated**. This option uses the process of tessellation to draw contour values on each element face.

3. Click **Apply** to implement your changes.

   The contour plot in the current viewport changes to use the contour method you have specified.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding how contours are rendered*
- *Customizing a contour plot*

Setting contour limits

By default, Abaqus/CAE automatically computes the limits of the values shown in your contour plot. You can control the minimum and maximum values Abaqus/CAE displays; for example, to eliminate extremes or to examine variations within a fixed set of bounds.

Contour limits, once set, remain in effect for the duration of the session. To learn how to display the minimum and maximum values associated with your contour plot, see *Customizing the legend*.

You can override the contour limits if you use custom contour interval values; see *Customizing contour intervals* for more information. When you use a custom contour interval, Abaqus/CAE uses the maximum and minimum interval values that you specify in the **Edit Intervals** dialog box as the contour limits for the contour plot.

If you specify the contour limits and subsequently modify the limits, the smaller the number of contour intervals increases the likelihood that your limit modifications will result in changes to the contour plot.

1. Locate the **Contour Limits** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox; then click the **Limits** tab in the dialog box that appears. The contour **Limits** options become available.

2. In the area labeled **Max**, choose one of the following options:

   - Select **Auto-compute** to request that Abaqus compute the maximum contour value.
   - Select **Specify** to specify the maximum value yourself; then enter the maximum value of your choice in the **Specify** text field.

3. In the area labeled **Min**, choose one of the following options:

   - Select **Auto-compute** to request that Abaqus compute the minimum contour value.
   - Select **Specify** to specify the minimum value yourself; then enter the minimum value of your choice in the **Specify** text field.

4. Click **Apply** to implement your changes.

   In the contour plot in the current viewport, Abaqus displays contour values within the limits you have specified. The contour legend, if active, changes to show the limits you have specified. If you have specified a maximum value that is less than the actual maximum or if you have specified a minimum value that is greater than the actual minimum, Abaqus displays the actual maximum or minimum value (respectively) in the legend. For more information on the contour legend, see *Customizing the legend*. To learn how to control the color of values that exceed your limits, see *Customizing contour colors*.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour limits*
- *Annotating the minimum and maximum contour values*
- *Controlling how Abaqus/CAE computes contour limits*

Annotating the minimum and maximum contour values

By default, Abaqus/CAE indicates the general locations of the minimum and maximum contour values through changes in the contour color. The actual locations may be in any part of the model displayed using the extremes of the selected color spectrum. You can toggle on **Show location** to annotate the nodal location of the minimum and/or maximum contour values. The annotations are of the form `Min: value` and `Max: value`, respectively, with a line pointing to the node, face (for quilt plots), or element centroid where the minimum and maximum occur.

Displaying the minimum or maximum value annotations activates the contour legend and the **Show min/max values** option (for more information, see *Customizing the legend*). The annotations are created automatically—you cannot edit them using the **Annotation Manager**. The annotation font and color match those of the legend text, and the node symbols are determined by the node symbol settings in the **Common Plot Options** dialog box. If you display the minimum and maximum contour values and a view cut is activated, the locations of the minimum and maximum values are updated relative to the viewport display. The contour legend is also updated to reflect the new bounds.



Note:

The minimum and maximum locations are not shown in an overlay plot.

1. Locate the **Show location** checkboxes.

   Select **Options->Contour** from the main menu bar or click  in the toolbox; then click the ***Limits tab*** in the dialog box that appears.

2. Toggle on **Show location** in the **Min** and **Max** areas, respectively, to annotate each location on the model.
3. Click **Apply** to implement your changes in the viewport.

   Abaqus/CAE activates the contour legend, including the **Show min/max values** option. The legend displays both the minimum and maximum values, regardless of whether you annotate one or both of them on the model.

4. If desired, use the **Viewport Annotation Options** to customize the annotation font and color (controlled by the legend text options), to hide the minimum and maximum values in the legend, or to hide the entire legend. Use the symbol and size options on the **Labels** tabbed page of the **Common Plot Options** dialog box to edit the displayed node symbols.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*
- *Customizing model labels*
- *Customizing the legend*

Controlling how Abaqus/CAE computes contour limits

By default, Abaqus/CAE automatically computes the minimum and maximum values shown in your contour plot. Automatic computation does not apply if you have *set limits* for both the minimum and maximum contour values.

For contour plots of nodal field output variables, the limits are the minimum and maximum nodal values. For *quilt-type contour* plots, the limits are the minimum and maximum element face values.

For *line-, banded-, or isosurface-type contours* of element field output variables, the limits are based on the values from individual elements extrapolated to and averaged at the nodes. The range may vary due to changes in the results averaging options. To learn how to control the averaging options, see *Controlling result averaging*.

For animations of contour plots, you can control which contour limits are used in each frame of the animation. You can choose to always use the auto-computed limits from the first and last frame, to always use the auto-computed limits from the current frame, or to recompute the limits for each frame.

1. Locate the **Auto-Computed Limits** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox; then click the **Limits** tab in the dialog box that appears. The **Auto-Computed Limits** options are in the lower half of the page.

2. Choose one of the following options for animation contour limits:

   - Select **Use first and last frame limits** to use the auto-computed contour limits from the first and last frame for each frame of the animation.
   - Select **Use current frame limits** to use the auto-computed contour limits from the current frame for each frame of the animation.
   - Select **Recompute limits for each frame** to recompute the contour limits for each frame of the animation based on the result values for the frame.
   - Select **Use limits from all frames** to recompute the auto-computed contour limits from all frames of the animation.

3. Click **Apply** to implement your changes.

   Contour limits for each frame of animations generated for the contour plot in the current viewport will be determined based on the option you selected. The *contour legend*, if active, changes to show the limits Abaqus computes.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour limits*
- *Setting contour limits*
- *Choosing line-, banded-, quilt-, or isosurface-type contours*

Customizing contour colors

By default, Abaqus/CAE represents contour values using the **Rainbow** color spectrum from red (for the maximum value) to blue (for the minimum value). You can select alternative color spectrums or create new color spectrums to use.
See *Creating a new color spectrum* for more information about creating a new color spectrum.

If your results include cyclic data in which positive and negative extreme values are equivalent, select the **Wraparound** color spectrum. This spectrum, with green at both extremes, correctly displays results such as the phase angle of a complex number, where 180° and −180° are the same.

By default, the contour plot legend displays values in descending order with the maximum value at the top. You can choose to reverse the order and display the values in ascending order with the minimum value at the top.

For non-tick mark contour plots you can also choose the colors Abaqus uses to represent values that exceed the limits you have specified for the plot, if any. By default, Abaqus displays such values in shades of gray. (For tick mark contour plots Abaqus displays the contour curve itself only within the limits you have specified.) You can choose to hide these values that are exceeding the limits specified for the plot from the contour legend. For more information on limiting the range of the plot, see *Setting contour limits*.

1. Locate the **Spectrum** options.

   Select **Options->Contour** from the main menu bar or click  in the toolbox. Click the **Color & Style** tab in the dialog box that appears; then click the **Spectrum** tab. The **Spectrum** options appear.

2. To choose the color of values within the limits of the plot, either select a spectrum name or click  to create and use a new color spectrum. See *Creating a new color spectrum* for information about creating a new color spectrum.

   Tip: Alternatively, you can choose or create a contour spectrum from the Results Tree. Click the **Spectrums** container to create a new spectrum, or click the name of any spectrum in the Results Tree to display the model using that spectrum.

3. To display the values in the contour plot legend in ascending order with the minimum value at the top, toggle on **Reverse contour legend range**.
4. To hide the values in the contour plot legend that exceed the specified limits, toggle on **Use specified limits for contour legend**.
5. To control the color of values that exceed the limits of the plot, do one of the following:

   - Click **Specify**; then click the **Greater than max** or **Less than min** color sample  to open the **Select Color** dialog box. Select a new color, and click **OK** to close the **Select Color** dialog box and update the color sample. (For more information, see *Customizing colors*.)

     Or,

   - Click **Use spectrum min/max** to accept the minimum and maximum contour spectrum colors for such values.

6. Click **Apply** to implement your changes.

   Contours in the current viewport change to the colors you have specified. The correspondence between contour colors and contour values appears in the contour legend, if active. For more information on the contour legend, see *Customizing the legend*.

By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*
- *Creating a new color spectrum*
- *Coloring elements with no results*

Creating a new color spectrum

Abaqus/CAE provides seven predefined color spectrums that you can use to customize the colors in a contour plot. If you prefer a different set of colors or a different arrangement, you can create a new color spectrum and apply its colors to the contour plot.

Creating and editing a color spectrum are similar processes. When you create a new color spectrum, Abaqus/CAE populates the **Create Spectrum** dialog box with the contents of the last color spectrum that you either modified or applied to the contour plot. If you had not yet modified a spectrum or created a contour plot in your session, Abaqus/CAE populates a new color spectrum with the colors in the predefined **Rainbow** color spectrum. You can then edit this new color spectrum by adding, modifying, moving, and deleting colors to achieve the content and arrangement that you want.

A spectrum must contain at least two colors and can contain as many colors as you want to include. In many situations the number of colors in your selected color spectrum will not match the number of contour intervals, so Abaqus/CAE cannot perform a one-to-one assignment of colors to contour intervals. In this case Abaqus/CAE performs one of the following actions:

- If your spectrum has more colors than the current number of contour intervals, Abaqus/CAE samples colors from the spectrum to create the contour plot. For example, when a spectrum contains 24 colors and only 12 intervals are selected, the contour plot will display every other color in the spectrum.
- If your spectrum has fewer colors than the current number of contour intervals, Abaqus/CAE interpolates between spectrum colors to determine additional colors to use for the contour intervals. All user-defined color spectrums follow an HSV interpolation scheme, while the **Red to blue** predefined spectrum follows an RGB interpolation scheme. This difference means that the **Red to blue** predefined spectrum and a user-defined spectrum could contain the same two colors, red and blue, but behave differently because their interpolation schemes would create different intermediate interval values. In general, HSV interpolation conforms more closely to human intuition than RGB interpolation does.

Color spectrums are session-specific definitions; therefore, they are available for use in any viewport during your session but are not saved to the output database (`.odb`). You can edit the colors and arrangement of any color spectrum, and you can rename, copy, or delete any user-created color spectrum. Abaqus/CAE does not allow you to rename or delete the seven predefined color spectrums. Abaqus/CAE provides access to color spectrums both from the main menu bar and from the Results Tree.

1. Locate the **Create Spectrum** dialog box.

   From the main menu bar, select **Tools**->**Spectrum**->**Create**.

   The **Create Spectrum** dialog box appears.
2. Specify the **Name** for this new color spectrum.
3. To insert a new color into the color spectrum:

   1. Highlight the color in the color spectrum that you want to precede or follow the color that you add.
   2. Click **Insert Before** or **Insert After** to add a new color in the indicated location.

      The **Select Color** dialog box opens.
   3. Choose a color, and click **OK** to close the **Select Color** dialog box.

   Abaqus/CAE adds the new color in the selected location.
4. To change one of the colors in the color spectrum, double-click the color and select a new color from the **Select Color** dialog box that appears.
5. To move a color within the color spectrum, highlight the color and click either **Move Up** or **Move Down** to move the color one step in the selected direction.
6. To delete a color from the spectrum, highlight the color and click **Delete**.

7. Continue adding, changing, moving, and deleting colors until the color spectrum contains the colors you want in the desired order.
8. Click **OK**.

   Abaqus/CAE adds the new color spectrum, making it available for use in customizing contour plots.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*
- *Customizing contour colors*
- *Coloring elements with no results*

Customizing contour intervals

Abaqus/CAE evenly divides the range between the minimum and maximum contour values into intervals. You can customize the number of intervals and whether the intervals are represented by discrete or continuous colors.

By default, there are 12 discretely colored intervals. *Figure 1* shows a banded-type contour plot with nine uniform contour intervals on the left and with continuous contours on the right.

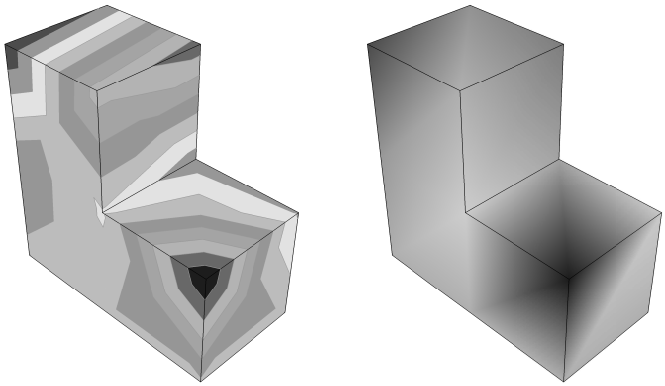


Figure 1: Contour plots showing uniform and continuous contour intervals.

When you represent contour plots using discrete intervals, you can also control the type of progression between interval values. If you choose a uniform interval type, Abaqus/CAE creates a uniform, arithmetic progression between the interval values; if you choose a logarithmic interval type, Abaqus/CAE creates a logarithmic progression between interval values. In addition, you can create a new set of contour intervals with a custom progression that includes any values that you want to include. You can also provide just a few interval values and interpolate additional intervals between the values you define. The interval values you provide must appear in descending order from top to bottom in the **Edit Intervals** dialog box.

If you select a custom interval, Abaqus/CAE overrides the contour limits specified in the **Limits** area of the **Contour Plot Options** dialog box and uses the maximum and minimum interval values as the custom contour limits. See *Setting contour limits*, for more information.

If you select custom contour intervals and subsequently modify the intervals, the smaller the number of contour intervals increases the likelihood that your interval modifications will result in changes to the contour plot.

The contour legend, if active, will include two more intervals than the number you choose. Abaqus/CAE adds intervals at the top and bottom of the legend to indicate any values that exceed the contour plot limits. For more information on the contour legend, see *Customizing the legend*.

This section describes how to customize the contour intervals and how to edit the contour intervals that you use.

1. Locate the **Contour Intervals** options.

   From the main menu bar, select **Options->Contour**; then click the **Basic** tab in the dialog box that appears. The **Contour Intervals** options are on the left side of the page.

2. Choose **Continuous** or **Discrete** to request continuous or discrete contour intervals, respectively.

   Continuous intervals are available only for *banded-type contours*. Abaqus represents your contour values in a smooth, non-delineated color spectrum.

   When you choose **Discrete** intervals, the **Contour Interval** slider becomes available.

3. For **Discrete** intervals, perform the following tasks:

   1. Choose the number of intervals by dragging the **Contour Intervals** slider to the number of intervals (between 2 and 24) that you want.

2. Choose one of the following selections from the **Interval type** list:
   - Select **Uniform** to use an arithmetic progression of values in the contour plot.
   - Select **Log** to use a logarithmic progression of values in the contour plot.
   - Select **User-defined** to use a customized set of values in the contour plot.

3. To customize interval values for the **User-defined** interval type, click .

   The **Edit Intervals** dialog box appears with a default set of contour intervals that you can modify in any of the following ways:
   - Edit any interval value by clicking its row and entering a new value. You must provide interval values in descending order from top to bottom in the dialog box.
   - Click **Insert Before** or **Insert After** to insert another interval before or after the selected interval. New intervals are added without values, so you must supply interval values either by entering them directly in the appropriate row or by interpolating their values.
   - Click **Delete** to remove the selected intervals from the set.
   - Click **Interpolate** to calculate interpolated interval values for all blank intervals in the set. Abaqus/CAE interpolates values using an arithmetic progression.
   - Click **Clear Values** to clear the values for the selected intervals.

   When you have completed your set of contour intervals, click **OK** to close the **Edit Intervals** dialog box.

4. Click **Apply** to implement your changes in the viewport.

   The contours and, if active, the *contour legend* in the current viewport change to reflect the interval options you have selected.

   By default, your changes are saved for the duration of the session and will affect all subsequent contour plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding contour plotting*
- *Customizing a contour plot*
- *Setting contour limits*
- *Customizing the legend*

Plotting analysis results as symbols

A symbol plot shows the magnitude and direction of a particular vector or tensor variable at a specified step and frame of the analysis. In addition, a symbol plot can be used to show the magnitude and direction of attributes such as loads or predefined fields at a specified step of a model in the current model database.
Abaqus/CAE represents the values as symbols (for example, arrows) at locations on your model. This section explains symbol plotting.

In this section:

- *Understanding symbol plotting*
- *Using symbol plot options*
- *Producing a results symbol plot*
- *Producing a symbol plot of free body nodal forces*
- *Customizing symbol plot appearance*

Understanding symbol plotting

Symbol plots allow you to visualize the magnitude and direction of vector and tensor variable results. In addition, a symbol plot can be used to show the magnitude and direction of attributes such as loads or predefined fields at a specified step of a model in the current model database.

Each vector and tensor result value appears as an arrow drawn at the location in the model where the result was obtained or the attribute is located; arrows representing nodal quantities appear at nodes, and arrows representing integration point quantities appear at integration points. (You cannot create a symbol plot of element quantities stored at nodes.)

For example, a symbol plot of the principal stress tensor is shown in *Figure 1*.

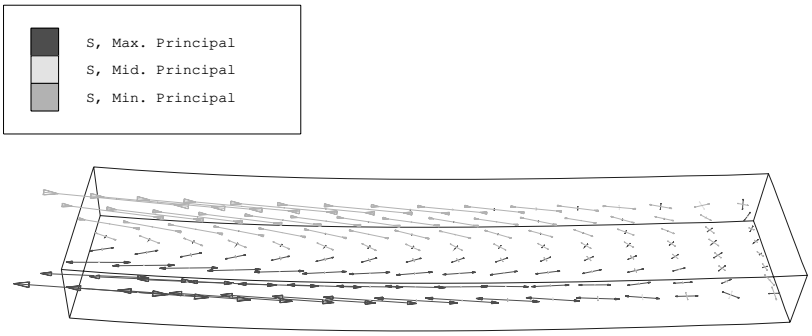


Figure 1: Symbol plot of principal stress.

Symbol plotting behavior changes slightly when you cut through a model using view cuts. If you display a view cut to investigate a symbol plot of a nodal output variable, Abaqus/CAE interpolates values from the nodes closest to the view cut surface and draws these interpolated vector arrows on the view cut surface. Abaqus/CAE performs this interpolation only for nodal output variables and does not perform any interpolation on the cutting surface when you display the portions of the model below the cut or above the cut. For more information on view cut functionality, see *Understanding view cuts*.

The relative sizes of the arrows indicate the magnitude of the result values. The directions of the arrows indicate the global directions of the results. In tensor plots arrows with arrowheads pointing in toward the arrow shaft represent compressive values; arrows with arrowheads pointing out from the arrow shaft represent tensile values.

The symbol plot legend shows you how each arrow color corresponds to a specific range of values. To learn how to display the minimum and maximum values associated with your symbol plot, see *Customizing the legend*.

By default, Abaqus/CAE plots all of a variable's result values in a symbol plot. The maximum value appears as the largest arrow in the plot, and all other arrows are scaled in proportion to that arrow size. However, you can use the **Symbol Plot Options** dialog box to limit the range of values that appear in a plot.

- When you specify a particular maximum value, only arrows representing absolute values less than or equal to that maximum value appear in the plot.

  The largest arrow in the plot will represent the largest absolute result value that is less than or equal to the limit you specify. All other arrows in the plot will be scaled in proportion to that arrow size.
- When you specify a particular minimum value, only arrows representing absolute values greater than that minimum value appear in the plot.

The components of connector vector output are resolved with respect to different coordinate systems depending on whether the output requested is field output or history output. See *Displaying connectors and connector output in the Visualization module* for how to interpret the vector component values in field plots.

Additional information

- *Setting vector and tensor limits*

Using symbol plot options

You can use the symbol plot options to customize the appearance of symbol plots.
For data from an output database, use the ***Symbol Variable*** options in the **Field Output** dialog box to select the vector or tensor quantity and optionally the component to plot.



Note: If you selected a model from the current model databases, symbol plots display the primary variable, not the symbol variable. Changing the symbol variable has no effect unless you plot data from an output database.

Select **Options->Symbol** from the main menu bar or click  in the toolbox to access the **Symbol Plot Options** dialog box. Click the following tabs to customize the appearance of symbol plots in the current viewport:

- **Color & Style**: The **Color & Style** page contains the following tabs:
  - **Vector**: Choose to customize the appearance of arrows that represent resultant values or selected component values of a vector variable.
  - **Tensor**: Choose to customize the appearance of arrows that represent all principal components, all direct components, or selected components of a tensor variable. Tensor arrows can be displayed at the element integration points, centroids, or nodes. If arrows are displayed at the nodes, the scalars are always computed after averaging.
- **Limits**: The **Limits** page contains the following tabs:
  - **Vector**: Choose either to specify the minimum and maximum vector values or to have Abaqus/CAE automatically compute these values.
  - **Tensor**: Choose either to specify the minimum and maximum tensor values or to have Abaqus/CAE automatically compute these values.
- **Labels**: The **Labels** page contains the following tabs:
  - **Vector**: Choose to display numerical labels for the vector values and to customize the color, font, type size, number format, and number of decimal places used for the vector values.
  - **Tensor**: Choose to display numerical labels for the tensor values and to customize the color, font, type size, number format, and number of decimal places used for the tensor values.

Other options such as the render style, edge visibility, deformation scale factor, and translucency are located in the ***Common Plot Options*** dialog box. If you choose to plot symbols on both the undeformed and the deformed shape, you can use the ***Superimpose Plot Options*** to customize the appearance of the undeformed shape. To learn how to customize the render style and other display characteristics of your symbol plot, see *Customizing plot display*. For more information on tensor and field output variables, see *Selecting field output variables*. For more information on selecting the results step, see *Selecting a specific results step and frame*.

Additional information

• *Customizing symbol plot appearance*

Producing a results symbol plot

A symbol plot shows either the magnitude and direction of a vector or tensor variable at a specified step and frame of an output database or the magnitude and direction of model attributes at a specified step of a model in the current model database. Abaqus/CAE represents the values as symbols (for example, arrows) at locations on your model. For more information on tensor and field output variables, see *Selecting field output variables*. For more information on selecting the results step, see *Selecting a specific results step and frame*. To learn how to display the minimum and maximum values associated with your symbol plot, see *Customizing the legend*.

1. *Open the model database or the output database* containing your model data or analysis results.
2. If you are plotting data from a model database, switch to the Visualization module, expand the **Model Database** container of the Results Tree, and select the model you want to use.
3. Select the results *step and frame* to display (or just the step, if the current model database is selected).
4. For output databases, select the *symbol field output variable* to display. For model databases, select the *primary field output variable* as the variable to display.

   The symbol plot state is activated.
5. Select the plot state–independent and symbol plot *customization* options you want.
6. If your chosen field output includes complex numbers, select the ***Complex Form*** tab in the **Result Options** dialog box to control the numeric form to display. This option applies only to plots for output databases.
7. For plots of output database data, select **Plot->Symbols** from the main menu bar to modify whether to plot the symbols on the deformed model shape, the undeformed shape, or both.

   Tip: You can also produce a symbol plot using the deformed , undeformed , or superimposed  symbol tools in the toolbox.

   The current viewport changes to display a customized symbol plot of the specified field output variable at the specified step and frame of the current output database.

   Abaqus automatically refreshes your symbol plot each time you click **Apply** in the step and frame selector dialog box, plot state–independent options, superimpose plot options (if applicable), or symbol plot options dialog boxes.

Additional information

- *Using symbol plot options*

Producing a symbol plot of free body nodal forces

You can create a symbol plot that shows the summation of free body nodal forces at individual nodes in your model. The symbol field output variable FREEBODY is the sum of the individual NFORC*n* components at a node, but with the vector's direction reversed. The FREEBODY output variable also takes into account the displayed elements by default; in contrast, contour plots of NFORC1 consider the whole model.

The FREEBODY symbol plot is a representation of the unbalanced forces acting on the object that is represented by the current display group. This sensitivity to current display group is not exhibited in any other context, such as contour plots.

1. *Open the output database* containing your analysis results with NFORC data included.
2. Select the results *step and frame* to display.
3. If desired, focus in on a subset of your model *using a display group*.
4. Select **FREEBODY** as the *symbol field output variable* to display.

   The symbol plot state is activated.
5. Select the plot state–independent and symbol plot *customization* options you want.
6. From the main menu bar, select **Plot->Symbols** to modify whether to plot the symbols on the deformed model shape, the undeformed shape, or both.

   Tip: You can also produce a symbol plot using the deformed , undeformed , or superimposed  symbol tools in the toolbox.

   The current viewport changes to display a customized symbol plot of free body nodal forces at the specified step and frame of the current output database for the current display group.

   Abaqus automatically refreshes your symbol plot each time you click **Apply** in the step and frame selector dialog box, plot state–independent options, superimpose plot options (if applicable), or symbol plot options dialog boxes.

Additional information

- *Selecting the symbol field output variable*
- *Using symbol plot options*
- *Resultant forces and moments on free body cuts in Abaqus/CAE*

Customizing symbol plot appearance

This section explains how to customize the style of symbol plot vector and tensor arrows, how to choose the quantity to display, and how to control the minimum and maximum values shown in the plot.

To learn how to customize the render style and the underlying model attributes of your symbol plot, see *Customizing plot display*.

In this section:

- *Customizing symbol plot arrows*
- *Setting vector and tensor limits*
- *Customizing vector and tensor labels*

Customizing symbol plot arrows

You can customize the arrow color, maximum arrow length, arrow shaft thickness, and arrowhead appearance for symbol plot vector and tensor arrows. When you customize arrow color for nodal vector symbols, you can select a single color for all arrows in the plot or display arrows with different colors depending on their length. You can use either the size of the model or the size of the screen as the basis for calculations of maximum arrow length.

For tensor symbol plots of element output variables, arrows are typically displayed with three different colors.

1. Locate the vector or tensor **Color & Style** options.

   Select **Options->Symbol** from the main menu bar or click  in the toolbox; then click the **Color & Style** tab in the dialog box that appears.

   - Click the **Vector** tab if you are creating a vector symbol plot.
   - Click the **Tensor** tab if you are creating a tensor symbol plot.

   The options for the symbol type of your choice appear.

2. To display all arrows in the plot with a single color, select **Uniform**, then do one of the following:

   - For a vector plot, click the **Uniform Color** sample , and select the color in which you want the variable or variable component arrows to appear. (For more information, see *Customizing colors*.)
   - For plotting all principal components in a tensor plot, click the color samples labeled **Maximum principal**, **Mid principal**, and **Minimum principal**; and select the colors in which you want each arrow type to appear.
   - For plotting all direct components in a tensor plot, click the color samples labeled **11**, **22**, and **33**; and select the colors in which you want each arrow type to appear.
   - For plotting only one principal or direct component, click the color sample labeled **Color**, and select the color in which you want the component arrows to appear.

3. To display all arrows in the plot using colors that correspond to their length, select **Spectrum**. You can choose one of the predefined color spectrums from the **Spectrum Name** list or define a new spectrum; for more information on selecting and defining a color spectrum, see *Customizing contour colors*, and *Creating a new color spectrum*. You can also drag the **Number of intervals** slider to change the number of intervals available for coloring the various arrows in the symbol plot; for more information about contour interval selection, see *Customizing contour intervals*.
4. Drag the **Size** slider to change the arrow length. The arrow size ranges from 0 to 30; the default arrow size is 6.

   Your selection determines the size of the arrow representing the largest vector or tensor value in the plot. All other vector and tensor arrows in the plot will be scaled to that size.

   Note:

   Abaqus/CAE maintains one size setting for both vectors and tensors; you cannot change the size of one independently of the other.

5. From the **Basis** options, select **Screen size** or **Model size** as the basis for calculations of maximum arrow length. If you select **Screen size**, Abaqus/CAE resizes the arrows as you change the size of the viewport; and if you select **Model size**, Abaqus/CAE resizes the arrows as you zoom in and out.
6. Click the arrow next to the **Thickness** field, and select the arrow shaft thickness of your choice.
7. Click the arrow next to the **Arrowhead** field, and select the arrowhead design of your choice.

8. If you are creating a tensor plot, customize the locations of the tensor arrows so that they are displayed at the **Integration point**, the **Centroid**, or the **Nodal** values for your model.

   Note:

   If you choose the **Nodal** option for tensor variables, the results averaging option **Compute scalars before averaging** is not valid; the scalars are always computed after averaging.

9. If desired, you can display a smaller subset of the arrows in a symbol plot to clarify the presentation of a plot with many arrows. Drag the **Symbol density** slider to a value between **High** and **Low**.
10. Click **Apply** to implement your changes.

    The symbol plot vector or tensor arrows in the current viewport change to reflect your settings.

    By default, your changes are saved for the duration of the session and will affect all subsequent symbol plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Customizing colors*
- *Customizing symbol plot appearance*

Setting vector and tensor limits

By default, Abaqus/CAE includes all of a variable's result values in a symbol plot. However, you can use the **Symbol Plot Options** dialog box to limit the range of values that appear in the plot.

When you specify a particular maximum value, only arrows representing values less than or equal to that maximum value appear in the plot. When you specify a particular minimum value, only arrows representing values greater than that minimum value appear in the plot. Changing the maximum or minimum value is also reflected in the symbol plot legend.

To learn how to display the minimum and maximum values associated with your symbol plot, see *Customizing the legend*.

1. Locate the **Limits** options.

   Select **Options**->**Symbol** from the main menu bar or click  in the toolbox; then click the **Limits** tab in the dialog box that appears. Click either the **Vector** tab or the **Tensor** tab; the symbol plot **Limits** options for the variable type of your choice become available.

2. In the area labeled **Max** in the top half of the page, choose one of the following options:

   - Select **Auto-compute** to use the largest result value as the maximum value.
   - Select **Specify** to specify the maximum value yourself; then enter the maximum value of your choice in the **Specify** text field.

3. In the area labeled **Min** in the bottom half of the page, choose one of the following options:

   - Select **Auto-compute** to use the smallest result value as the minimum value.
   - Select **Specify** to specify the minimum value yourself; then enter the minimum value of your choice in the **Specify** text field.

4. Click **Apply** to implement your changes.

   The symbol plot in the current viewport changes to display arrows only for variable values that fall within the limits you have specified.

   By default, your changes to symbol plot options are saved for the duration of the session and will affect all subsequent symbol plots in the current viewport and in any new viewports created from the current viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Understanding symbol plotting*

Customizing vector and tensor labels

You can display a label next to each vector or tensor arrow in a symbol plot that expresses its value in numerical form. You can also customize these labels by changing their color, font, type size, number format, and number of decimal places.

1. Locate the **Labels** options.

   Select **Options->Symbol** from the main menu bar or click  in the toolbox; then click the **Labels** tab in the dialog box that appears. Click either the **Vector** tab or the **Tensor** tab; the symbol plot **Labels** options for the variable type of your choice become available.

2. Toggle on **Display value next to vector symbol** to show vector labels for the selected type of vector.
3. Select the color of the vector labels for the selected type of vector.

   a. Click the **Text color** color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

4. Select the font of the vector labels for the selected type of vector.

   a. Click **Set Font**.

   Abaqus/CAE displays the **Select Font** dialog box.

   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.

   c. Click **OK** to close the **Select Font** dialog box.

5. To change the format of the vector labels for the selected vector type, click the **Number Format** arrow and select one of the following formats:

   - Select **Scientific** to display legend values in scientific notation, which expresses legend values as a decimal value between 1 and 10 multiplied by a power of 10. When you select this format for the number `501,000`, the value appears in the legend as `5.01e5`.
   - Select **Fixed** to display legend values without using exponent values. When you select this format, the number `501,000` is displayed in the legend as `501000.00`.
   - Select **Engineering** to select engineering notation, which expresses values in a similar format to scientific notation but allows exponent values that are multiples of three only. When you select this format, the number `501,000` is displayed in the legend as `501E3`.

6. Click the **Decimal places** arrows to specify the desired number of decimal places in each label.
7. Click **Apply** to implement your changes.

   The symbol plot in the current viewport changes to display labels with your selected formatting.

   By default, your changes to symbol plot options are saved for the duration of the session and will affect all subsequent symbol plots in the current viewport and in any new viewports created from the current viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Understanding symbol plotting*

Plotting material orientations

A material orientation plot shows the material directions of elements in the model at the element integration points.

The material orientations are plotted at a specified step and frame of the analysis and at a specified section point for shell elements. Material orientation plots can also be created to show the material directions of reinforcements (rebar orientation) in membrane, shell, and surface elements. Abaqus/CAE represents the material orientations as triads. This section explains material orientation plotting.

In this section:

- *Understanding material orientation plotting*
- *Using material orientation plot options*
- *Producing a material orientation plot*
- *Customizing material orientation plot triads*

Understanding material orientation plotting

Material orientation plots allow you to visualize the material directions of the elements in your model.

Abaqus/CAE draws a triad at the element integration points to indicate the material orientation. For example, a material orientation plot is shown in *Figure 1*.

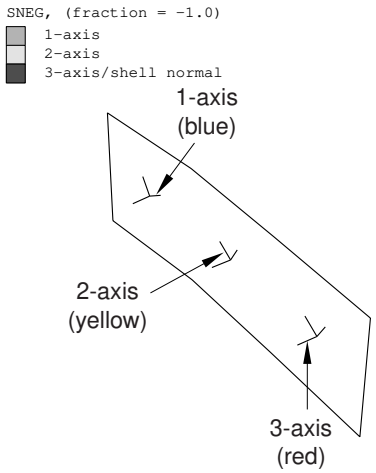


Figure 1: Material orientation plot.

Material orientations are available only for those elements for which field output is requested at the selected section points. Valid elements include all shell elements, as well as solid elements for which you have defined a local orientation. The material orientations are displayed at the current step and frame of the analysis and at the current section point selection for shell elements. If display is enabled for both the top and bottom section point locations, Abaqus/CAE displays material orientations for the bottom location only. For small-strain shell elements the material orientations do not rotate with the deformation of the model (see *About Shell Elements*, for more information). Material orientation plots can also be created for reinforcement layers in membrane, shell, and surface elements.

Additional information

• *Selecting section point data by category*

Using material orientation plot options

You can use the material orientation plot options to customize the appearance of material orientation triads.

Select **Options->Material Orientation** from the main menu bar or click  in the toolbox to access the **Material Orientation Plot Options** dialog box. Use the options in the dialog box to control the visibility, color, length, thickness, and arrow appearance of the material orientation triad axes.

Other options such as the render style, edge visibility, deformation scale factor, and translucency are located in the ***Common Plot Options*** dialog box. If you choose to plot material orientations on both the undeformed and the deformed shape, you can use the ***Superimpose Plot Options*** to customize the appearance of the undeformed shape. For more information on customizing a superimposed plot, see *Superimposing deformed and undeformed model plots*. To learn how to customize the render style and other display characteristics of your material orientation plot, see *Customizing plot display*. For more information on selecting the results step, see *Selecting a specific results step and frame*. For more information on selecting section point locations, see *Selecting section point data by category*.

Additional information

- *Customizing material orientation plot triads*

Producing a material orientation plot

A material orientation plot shows the material directions of the elements in a model at a specified step and frame and a specified section point. Abaqus/CAE represents the material orientations as triads at the element integration points. For more information on selecting the results step, see *Selecting a specific results step and frame*. For more information on selecting section point locations, see *Selecting section point data by category*.

1. *Open the output database* containing your analysis results.
2. Select the results *step and frame* to display.
3. Select the *section point* to display.
4. Select the plot state–independent and material orientation plot *customization* options you want.
5. From the main menu bar, select **Plot->Material Orientations** and choose whether to plot the material orientations on the undeformed shape, the deformed shape, or both.

   Tip: You can also produce a material orientation plot using the deformed , undeformed , or superimposed  material orientation tools in the toolbox.

   The current viewport changes to display a customized material orientation plot.

   Abaqus automatically refreshes your material orientation plot each time you click **Apply** in the step and frame selector, section point selector, plot state–independent options, superimpose plot options (if applicable), or material orientation plot options dialog boxes.

Additional information

- *Using material orientation plot options*

Customizing material orientation plot triads

This section explains how to customize the style of material orientation plot triads.

To learn how to customize the render style and the underlying model attributes of your material orientation plot, see *Customizing plot display*.

You can customize the color, length, thickness, and arrowhead appearance of the material orientation plot triad axes. You can also suppress the appearance of one or more of the triad axes; and for results with composite sections, you can display results using either the material orientation of the entire layup or one of its individual plies.

1. Select **Options->Material Orientation** from the main menu bar or click  in the toolbox.

   The triad options appear.
2. Toggle on the buttons to the left of each of the axes to display or suppress them.
3. Click on the color samples to select the color or colors of your choice for the visible triad axes in the plot.
4. Drag the **Size** slider to increase or decrease the length of the axes in the material orientation triads in your plot. Axis length ranges from 0 to 30; the default length is 6.

   Note:

   You can specify an axis length value larger than 30 by using the Abaqus Scripting Interface. See *Using the Python interpreter*.
5. From the **Basis** options, select **Screen size** or **Model size** as the basis for calculations of axis length in material orientation triads. If you select **Screen size**, Abaqus/CAE resizes the triads as you change the size of the viewport; and if you select **Model size**, Abaqus/CAE resizes the triads as you zoom in and out.
6. Click the arrow next to the **Thickness** field, and select the axis thickness of your choice.
7. Click the arrow next to the **Arrowhead** field, and select the arrowhead design of your choice.
8. If desired, you can display a smaller subset of the material orientation triads to clarify the presentation of a plot with many triads. Drag the **Symbol density** slider to a value between **High** and **Low**.
9. Material orientation based on the SORIENT output variable is plotted when **Section / Layup** is selected. If your results include field output from the SORIENT output variable and output from elements with composite sections, specify the material orientation you want to use. Select **Material / Ply** to display each ply with its own material orientation, or select **Section / Layup** to display the entire composite layup using the orientation specified in the composite layup definition.
10. Click **Apply** to implement your changes.

    The material orientation plot triads in the current viewport change to reflect your settings.

    By default, your changes are saved for the duration of the session and will affect all subsequent material orientation plots. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Customizing colors*
- *Customizing plot display*

X–Y plotting

This chapter explains the concept of *X–Y* plotting and gives details of how to create *X–Y* data objects and how to produce and customize an *X–Y* plot.

In this section:

- *Understanding X–Y plotting*
- *Specifying and saving X–Y data objects*
- *Producing an X–Y plot*
- *Operating on saved X–Y data objects*
- *Customizing X–Y plot axes*
- *Customizing X–Y curve appearance*
- *Customizing X–Y plot appearance*
- *Customizing the X–Y plot title*
- *Customizing the appearance of the X–Y plot legend*
- *Customizing border and fill colors for an X–Y plot*

Understanding X–Y plotting

An *X–Y* data object is a two-dimensional array of data that Abaqus/CAE stores in two columns. You can display *X–Y* data as a graph in an *X–Y* plot or as a table in an *X–Y* report. In addition, you can probe an *X–Y* plot to display the *X*- and *Y*-coordinates of graph points. This section discusses *X–Y* data objects and the various methods you can use to create them. For more information on *X–Y* reports, see *Generating tabular data reports*; for information on probing *X–Y* plots, see *Probing the model*.

In this section:

- *What is an X–Y data object, and what is an X–Y plot?*
- *Understanding how to specify an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Understanding quantity types*

What is an X–Y data object, and what is an X–Y plot?

An *X*–*Y* data object is a collection of ordered pairs that Abaqus/CAE stores in two columns—an *X*-column and a *Y*-column. The *X*–*Y* data can originate from an output database or an ASCII file, or you can enter the data using the keyboard. In addition, you can derive *X*–*Y* data by combining existing *X*–*Y* data objects. For example, you might combine an *X*–*Y* data object containing stress values versus time with a second *X*–*Y* data object containing strain values versus time to produce an *X*–*Y* data object containing stress versus strain at equivalent times. You can save *X*–*Y* data objects; and you can edit, copy, rename, and delete them.

By default, the *X*–*Y* data objects you save are retained for the duration of the session only. If you want to make *X*–*Y* data objects available for subsequent sessions, you can save them to an XML file, to the model database, or to an output database using the **File->Save Session Objects** option; or you can copy them to an output database file using the **Copy to ODB** button in the **XY Data Manager**.

Abaqus/CAE can display *X*–*Y* data in the form of an *X*–*Y* plot. An *X*–*Y* plot is a two-dimensional graph of one variable versus another. Examples of *X*–*Y* plots include temperature versus time, load versus displacement, and stress versus strain. You can display multiple *X*–*Y* data objects in one *X*–*Y* plot, and you can use the customization options to control the appearance of each data object and the overall appearance of the components of an *X*–*Y* plot. These components include the axes; the legend, which is a key that associates each *X*–*Y* curve in the plot with the *X*–*Y* data object it represents; and the *X*–*Y* curves. *Figure 1* shows an *X*–*Y* plot displaying three *X*–*Y* data objects.

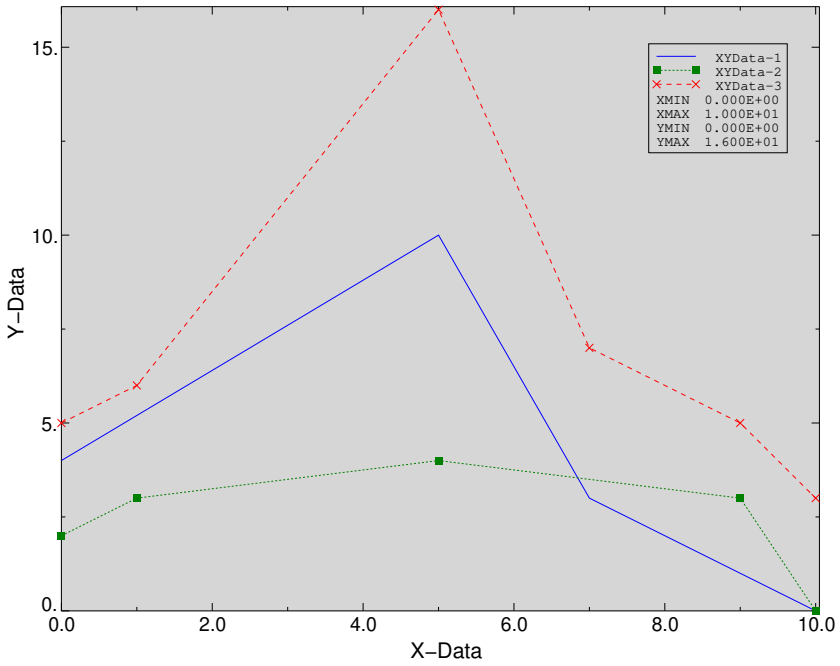


Figure 1: X–Y plot displaying three objects of data.

Additional information

- *Managing session objects and session options*
- *Producing an X–Y plot*
- *Specifying and saving X–Y data objects*
- *Managing session objects and session options*
- *Copying a session X–Y data object to an output database file*

Understanding how to specify an X–Y data object

To specify an *X–Y* data object, you first choose the source of the data and then provide any necessary details. Possible sources are:

ODB history output

Select this method to specify an *X–Y* data object by reading history output results from an output database. You can specify which variables to read from the output database, from which steps of the analysis to read, and the frequency at which to read the data; for example, you can read every third data point. You can also specify a numeric form for any complex-valued output data. For more information, see *Reading X–Y data from output database history output*.

ODB field output

Select this method to specify an *X–Y* data object by reading field output results from an output database. You can specify which variables to read from the output database and for which elements or nodes to read the data. Abaqus/CAE extracts results from the currently active steps and frames; see *Activating and deactivating steps and frames*, for more information. You can also specify a numeric form for any complex-valued output data. For more information, see *Reading X–Y data from output database field output*.

Thickness

Select this method to specify an *X–Y* data object by reading field output results from elements through the thickness of a shell region of your model. Abaqus/CAE extracts results from the current step and frame. You can specify which variables to read from the output database and for which elements to read the data. For more information, see *Reading X–Y data through the thickness of a shell*.

Free body

Select this method to specify an *X–Y* data object by reading field output results from all active free bodies in your session. Abaqus/CAE extracts results from the current step and frame and displays results for the nodal force (NFORC) output variable. For more information, see *Reading X–Y data from all active free body cuts*.

Operate on X–Y data

Select this method to derive a new *X–Y* data object by manipulating previously saved *X–Y* data objects. You specify the new *X–Y* data object by applying functions and mathematical operations to existing data. An example of a function is **Combine**. If you **Combine** an *X–Y* data object containing stress versus time with an *X–Y* data object containing strain versus time, you produce an *X–Y* data object containing stress versus strain at equivalent times. For more information, see *Operating on saved X–Y data objects*.

ASCII file

Select this method to read *X*- and *Y*-values from an existing text file. The file can contain more than two columns of data, separated by commas, spaces, or tabs; and you can specify which columns correspond to the *X*- and *Y*-axis data. In addition, you can specify the frequency at which the data should be read from the file; for example, every third row. For more information, see *Reading X–Y data from an ASCII file*.

Keyboard

Select this method to manually type *X*- and *Y*-values into a simple table editor. Within this method, Abaqus supports several special editing techniques, as well as an option to read data from a file. For more information on this topic, see *Entering X–Y data from the keyboard*.

Path

Select this method to specify an *X–Y* data object by reading field output results at locations along a path through your model. Abaqus obtains results from an output database. You can specify the points, elements, or edges that make up the path and the step, frame, and variable for which to obtain results. For more information, see *Viewing results along a path*.

In addition, you can create an *X–Y* data object while using the table editor to create a material in the Property module. For more information, see *Entering tabular data*.

Once you have specified your *X–Y* data object, you can save it or you can display it in the form of an *X–Y* plot. Saving the *X–Y* data object allows you to subsequently plot, edit, rename, delete, or operate on it; it also allows you to copy the *X–Y* data object to an output database file for use in later Abaqus sessions. For *X–Y* data originating from sources other than output database history output, you must save your data to later produce an *X–Y* plot containing multiple data objects. Saved *X–Y* data objects are retained only for the duration of the session. For the *X–Y* data object to be persistent across sessions, you must copy it to an output database file.

If you have copied an *X–Y* data object to your output database file during an earlier session, you can access that object when you open the file in the current session. You can display the *X–Y* data object in the form of an *X–Y* plot, just as you can any other specified objects. To edit, rename, delete, or operate on a data object that was created in a previous session, you must load it to the current session.

Understanding “Temp” and other X–Y data object names

By default, when you save or plot an *X–Y* data object, Abaqus/CAE names it for you. This name is an important means of identifying and referring to the *X–Y* data object.

X–Y data object names appear in the following instances:

- When you plot *X–Y* data, the *X–Y* plot legend identifies by name all *X–Y* data objects appearing in the plot.
- When you select **Tools->XY Data** from the main menu bar, the **Manager**, **Edit**, **Copy**, **Rename**, **Delete**, and **Plot** tools list all *X–Y* data objects created during the session by name.
- When you *customize the appearance of the curve* representing an *X–Y* data object, you first select the *X–Y* data object by name from those listed in the **XY Curve Options** dialog box.

In most cases Abaqus/CAE is able to provide a meaningful default name for your *X–Y* data object. As you save your *X–Y* data object, Abaqus/CAE shows you the default name and gives you an opportunity to override this default with the name of your choice. Once saved, you can provide a new name for your *X–Y* data object at any time by renaming it. For information on renaming, see *Managing objects using manager dialog boxes*.

There are three basic forms of *X–Y* data object names, as follows:

Temp-*n*

If you produce an *X–Y* plot by clicking **Plot** from any of the following dialog boxes:

- **XY Data from ASCII File**
- **XY Data from Keyboard**
- **XY Data from Path**
- **Plot Expression** from the **Operate on XY Data** dialog box

Abaqus/CAE names your *X–Y* data according to the pattern **Temp-1**, **Temp-2**, etc. This name indicates that the plot represents the data configured in the dialog box, which is considered temporary data whether or not you have clicked **Save As** to save it.

Conversely, if **Temp-***n* appears in your plot legend, you are plotting temporary data. You cannot edit, copy, rename, delete, operate on, or produce a report of temporary data. To accomplish any of these tasks, you must first save your data. You can select curve style preferences for temporary data, but the curve styles will revert to the defaults if you replot the data without having saved it. To learn how to save your data, see *Saving an X–Y data object*.

XYData-*n*

If you create a new *X–Y* data object using the **Save As** button provided in the **XY Data from ASCII File** or **XY Data from Keyboard** dialog boxes, Abaqus/CAE suggests a default name for your *X–Y* data according to the pattern **XYData-1**, **XYData-2**, etc. To establish a more meaningful name, you can use the **Save XY Data As** dialog box to overwrite this default with the name of your choice.

To locate the **XY Data from ASCII File** or **XY Data from Keyboard** dialog boxes, select **Tools->XY Data->Create** from the main menu bar; then click the choice of interest in the dialog box that appears, and click **Continue**.

Meaningful names

You can establish a meaningful name (a name of your choice) for your *X–Y* data object either by specifying a name as you save your data or by subsequently renaming it. As you save your data, Abaqus/CAE displays the

default name in the **Save XY Data As** dialog box; you can overwrite this default. Abaqus/CAE generates meaningful default names for *X–Y* data created from the **History Output** or **XY Data from ODB Field Output** dialog boxes.

Understanding quantity types

A quantity type is a category that describes the data in one of the columns of an *X–Y* data object. Abaqus/CAE includes over eighty predefined quantity types that describe the output of every variable in Abaqus, including commonly used types like temperature, time, stress, and strain. The quantity types associated with each column appear in *X–Y* reports and as the default axis labels in an *X–Y* plot of that data object.

When you create an *X–Y* data object from field output data or history output data, Abaqus/CAE automatically uses the quantity types specified in the output database. When you create an *X–Y* data object by loading data from an ASCII file or by entering data directly from the keyboard, you can specify quantity types for each column as you create the object. In either case, you can change the quantity types associated with either column by editing the data object in the **Edit XY Data** dialog box.

Specifying and saving X–Y data objects

This section explains the various methods of specifying and saving *X–Y* data objects. *X–Y* data objects can originate from analysis results, an external file, keyboard entry, or operations on previously saved *X–Y* data objects.

For information on specifying *X–Y* data objects by obtaining field output results along a path through your model, see *Viewing results along a path*; for information on specifying *X–Y* data objects by operating on existing *X–Y* data, see *Operating on saved X–Y data objects*.

In this section:

- *Reading X–Y data from output database history output*
- *Reading X–Y data from output database field output*
- *Reading X–Y data through the thickness of a shell*
- *Reading X–Y data from all active free body cuts*
- *Reading X–Y data from an ASCII file*
- *Entering X–Y data from the keyboard*
- *Saving an X–Y data object*
- *Copying a session X–Y data object to an output database file*
- *Loading an X–Y data object to the current session*
- *Editing individual data points in an X–Y data object*

Reading X–Y data from output database history output

You can read *X*–*Y* data from history output in the output database. Available data consist of history output that were saved during the analysis.

From this output, you can select the following:

- a variable at a location—for example, U1 (displacement in the *X*-direction) at node 1;
- the step or steps of interest;
- a pattern of frames within those steps—for example, every 10th frame; and
- the time basis.

Abaqus/CAE then reads *X*–*Y* data pairs from the output database at the frames you specify. If you select multiple variables, Abaqus/CAE reads the data pairs and creates a separate *X*–*Y* data object for each variable by default. When you save history output variables, you can choose to operate on the data first; Abaqus/CAE saves the results of the operation as a new *X*–*Y* data object.

For time-based analyses, *X*-values are taken as total time from the start of the analysis or, in the case of a restarted analysis, as total time from the start of the last continuation of the analysis. Alternatively, you can use the step time as the time basis. *Y*-values are taken as the value at that time of the variable at the location you specify.

History output from an Abaqus/Standard analysis is available after 10 increments are written to the output database; history output from an Abaqus/Explicit analysis is available after 100 increments are written to the output database or after an Abaqus/Explicit heartbeat is triggered based on CPU time, whichever occurs first.

1. Locate the **History Output** options.

   Select **Tools->XY Data->Create** from the main menu bar, then choose **ODB history output**.

   Tip: You can also specify history output by selecting **Result->History Output** from the main menu bar.

   The **History Output** dialog box appears.

   Note:

   The **History Output** dialog box lists all history output saved during the analysis, giving the variable name and the location at which it was saved. If this list is empty, history output is not available.

2. Click the **Variables** tab, if it is not already selected. Select one or more variables that you want to read. Variables are listed alphabetically along with their locations. The default is to read the first entry in the list. For more information on selecting multiple items in dialog boxes, see *Selecting multiple items from lists and tables*.

   If your output database contains many history output variables, you can use the filter to reduce the number of output variable names displayed. Click next to the **Name filter** field to see examples of valid filtering syntax.

   The specified variable and location are highlighted.

3. If viewport linking is activated, you can toggle on **Show common list between linked viewports** to display only the variables that are common to all linked viewports.

4. Click **Steps/Frames** to change the steps, frames, or time basis from which Abaqus/CAE reads *X*–*Y* data. For more information, see *Activating and deactivating steps and frames*.

5. If the data include complex numbers, you can select the displayed form using the **Result Options** dialog box. For more information on complex number forms in Abaqus/CAE, see *Controlling the form of complex results*.
6. To evaluate and display the data, click **Plot**.

   An *X–Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.
7. To save the data you have configured, click **Save As**. Any complex data are saved as the current complex form, not as complete complex numbers.

   Note:

   To plot saved *X–Y* data, select **Tools->XY Data->Plot** from the main menu bar and choose the *X–Y* data from the pull-right menu.

   In the **Save XY Data As** dialog box that appears, you can perform any of a number of mathematical, trigonometric, logarithmic, exponential, or other operations on the data before saving (see *Saving an X–Y data object*, for more information).
8. When you have finished, click **Dismiss** to close the **History Output** dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Controlling the form of complex results*
- *Linking viewports*

Reading X–Y data from output database field output

You can read *X–Y* data from field output in the output database.

Available data consist of field output that were saved during the analysis. From this output, you can select the following:

- a variable at an output position—for example, S11 (stress in the 11-direction) at integration points; and
- a single location or a set of locations (elements or nodes).

Abaqus/CAE then reads *X–Y* data pairs from the output database at the currently active steps and frames. If you select multiple variables or multiple locations, Abaqus/CAE reads the data pairs and creates a separate *X–Y* data object for each variable/location pair. For all time-based analyses, *X*-values are taken as total time from the start of the analysis. *Y*-values are taken as the corresponding values of the variable at the specified location.

Element-based field data extracted at nodes are controlled by the current settings for result averaging. However, if you requested the unique nodal position, then result averaging is always 100% to obtain unique values at the nodes. For more information, see *Understanding result value averaging*.

When reading the *X–Y* data for element-based field output at any output position, Abaqus/CAE reads the field values from the specified position if the same position is available in the output database.

1. Locate the **XY Data from ODB Field Output** options.

   Select **Tools->XY Data->Create** from the main menu bar, then choose **ODB field output**.

   The **XY Data from ODB Field Output** dialog box appears.

   Note: The **XY Data from ODB Field Output** dialog box lists all field output saved during the analysis (except quaternion variables). If this list is empty, field output is not available.

2. Click the **Position** arrow to reveal possible positions at which to list variables for your selection; then choose the desired position.

   The list of variables is refreshed to show only those that can be read at the selected position.

3. Select the field output variables to be read using either the check box next to each variable in the list or the **Edit** text field at the bottom of the page.

   ### To use the check box method:

   1. To select a variable and all of its components, click that variable's check box.
   2. To choose among the individual components of a variable, click the arrow next to that variable's check box to list its components; then click individual component check boxes to select them.

   ### To use the edit method:

   In the **Edit** text field, enter the names of the variables and components to be read. To be valid, a variable must be available on the output database for the current position.

4. Click the **Elements/Nodes** tab.

   The **Elements/Nodes** options appear.

5. From the **Item** list at the upper left of the dialog box, select either **Elements** or **Nodes**.

   Abaqus refreshes the **Selection Method** list at the bottom of the dialog box and the item list at the right.

6. Select the specific elements or nodes for which to read output data.

   - To specify elements or nodes by picking them directly from the viewport:

     1. Choose **Pick from viewport** from the **Selection Method** list.

        Abaqus/CAE enters the pick mode. **Select *items* for the display group** (where *items* are elements or nodes) appears in the prompt area.

     2. Select one or more items from the viewport (for more information, see *Selecting objects within the viewport*"). The items are highlighted in the viewport.

        Note:

        If you click **Done** in the prompt area, Abaqus/CAE exits the pick mode and your selections disappear; you must choose a **Selection Method** from the list.

   - To specify elements or nodes by number:

     1. Choose **Element labels** or **Node labels** from the **Selection Method** list.

        The **Part instance** and **Labels** fields appear.

     2. Select the name of the part instance for which you are obtaining results from the list in the **Part instance** field. Type into the **Labels** field a list of element or node numbers separated by commas or a range of numbers such as 1:4.

     3. Verify your selection by clicking **Highlight Items in Viewport**.

   - To specify elements or nodes by set name:

     1. Choose **Element sets** or **Node sets** from the **Selection Method** list.

        Abaqus refreshes the item list at the right. If this list is empty, there are no items that meet your selection criteria.

        Note:

        This list does not include the internal sets generated by Abaqus/CAE.

     2. Select one or more set names from the item list. (For more information, see *Selecting multiple items from lists and tables*.) If your model contains many sets, you can use the filter to reduce the number of set names displayed. Click  next to the **Filter** field to see examples of valid filtering syntax.

     3. Verify your selection by toggling **Highlight items in viewport**.

   - To specify elements or nodes belonging to internal sets (sets created by Abaqus/CAE):

     1. Choose **Internal sets** from the **Selection Method** list.

        Abaqus refreshes the item list at the right. If this list is empty, there are no items that meet your selection criteria.

     2. Select one or more set names from the item list. (For more information, see *Selecting multiple items from lists and tables*.) If your model contains many internal sets, you can use the filter to reduce the number of set names displayed. Click  next to the **Filter** field to see examples of valid filtering syntax.

     3. Verify your selection by toggling **Highlight items in viewport**.

The following table summarizes the restrictions that exist on the availability of element or node data for the output position specified on the **Variables** page:

Output position	Available for
Integration point	Elements
Centroid	Elements
Element nodal	Elements or nodes
Unique nodal	Nodes

7. Click **Active Steps/Frames** to change the steps or frames from which Abaqus/CAE reads *X–Y* data. For more information, see *Activating and deactivating steps and frames*.
8. If the data include complex numbers, you can select the displayed form using the **Result Options** dialog box. For more information on complex number forms in Abaqus/CAE, see *Controlling the form of complex results*.
9. To evaluate and display the data, click **Plot**.

   An *X–Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save** to save it.
10. To save the data you have configured, click **Save**. Any complex data are saved as the current complex form, not as complete complex numbers.

    Note:

    To plot your saved *X–Y* data, select **Tools->XY Data->Plot** from the main menu bar and choose the *X–Y* data from the pull-right menu.

11. When you have finished, click **Dismiss** to close the dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Controlling the form of complex results*

Reading X–Y data through the thickness of a shell

You can read *X–Y* data from field output data that are stored in the elements through the thickness of a shell or composite solid region of your model. Available data consist of field output that were saved during the analysis. From this output, you can select the following:

- a variable at an output position—for example, S11 (stress in the 11-direction) at integration points; and
- a single element or a set of elements.

Abaqus/CAE then reads *X–Y* data pairs from the output database at the current step and frame. If you select multiple variables or multiple locations, Abaqus/CAE reads the data pairs and creates a separate *X–Y* data object for each variable/location pair. *Y*-values are taken as the thickness through the model. *X*-values are taken as the corresponding values of the variable at the specified location.

1. Locate the **Thickness** options.

   Select **Tools->XY Data->Create** from the main menu bar, then choose **Thickness**.

   The **XY Data from Shell Thickness** dialog box appears.

   Note:

   The **XY Data from Shell Thickness** dialog box lists all field output saved during the analysis (except quaternion variables). If this list is empty, field output is not available.

2. Click the **Position** arrow to reveal possible positions at which to list variables for your selection; then choose the desired position.

   The list of variables is refreshed to show only those that can be read at the selected position.

3. Select the field output variables to be read using either the check box next to each variable in the list or the **Edit** text field at the bottom of the page.

   ### To use the check box method:

   1. To select a variable and all of its components, click that variable's check box.
   2. To choose among the individual components of a variable, click the arrow next to that variable's check box to list its components; then click individual component check boxes to select them.

   ### To use the edit method:

   In the **Edit** text field, enter the names of the variables and components to be read. To be valid, a variable must be available in the output database for the current position.

4. Click the **Elements** tab.

   The **Elements** options appear.

5. Select the specific elements for which to read output data.

   - To specify elements by picking them directly from the viewport:

     1. Choose **Pick from viewport** from the **Selection Method** list.

        Abaqus/CAE enters the pick mode. **Select elements for the display group** appears in the prompt area.

2. Select one or more elements from the viewport (for more information see *Selecting objects within the viewport*"). The elements are highlighted in the viewport.

   Note:

   If you click **Done** in the prompt area, Abaqus/CAE exits the pick mode and your selections disappear; you must choose a **Selection Method** from the list.

- To specify elements by number:

  1. Choose **Element labels** from the **Selection Method** list.

     The **Part instance** and **Element Labels** fields appear.

  2. Select the name of the part instance for which you are obtaining results from the list in the **Part instance** field. Type into the **Element Labels** field a list of element numbers separated by commas or a range of numbers such as 1:4.
  3. Verify your selection by clicking **Highlight Items in Viewport**.

- To specify elements by set name:

  1. Choose **Element sets** from the **Selection Method** list.

     Abaqus refreshes the item list at the right. If this list is empty, there are no items that meet your selection criteria.

     Note:

     This list does not include the internal sets generated by Abaqus/CAE.

  2. Select one or more set names from the item list. (For more information, see *Selecting multiple items from lists and tables*.) If your model contains many sets, you can use the filter to reduce the number of set names displayed. Click  next to the **Filter** field to see examples of valid filtering syntax.
  3. Verify your selection by toggling **Highlight items in viewport**.

- To specify elements belonging to internal sets (sets created by Abaqus/CAE):

  1. Choose **Internal sets** from the **Selection Method** list.

     Abaqus refreshes the item list at the right. If this list is empty, there are no items that meet your selection criteria.

  2. Select one or more set names from the item list. (For more information, see *Selecting multiple items from lists and tables*.) If your model contains many internal sets, you can use the filter to reduce the number of set names displayed. Click the **Tip** button next to the **Filter** field to see examples of valid filtering syntax.
  3. Verify your selection by toggling **Highlight items in viewport**.

6. If the data include complex numbers, you can select the displayed form using the **Result Options** dialog box. For more information on complex number forms in Abaqus/CAE, see *Controlling the form of complex results*.
7. To evaluate and display the data, click **Plot**.

An *X–Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save** to save it.

8. To save the data you have configured, click **Save**. Any complex data are saved as the current complex form, not as complete complex numbers.

   Note:

   To plot your saved *X–Y* data, select **Tools->XY Data->Plot** from the main menu bar and choose the *X–Y* data from the pull-right menu.

9. When you have finished, click **Dismiss** to close the dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Composite layups*

Reading X–Y data from all active free body cuts

You can read *X–Y* data from all the active free body cuts defined in your session. The data available for selection include the following:

- data from forces, moments, and heat flow rate; and
- if you select forces or moments, the resultant magnitude and the individual components in the 1-, 2-, and 3-directions; and
- if you select heat flow rate (valid only for free body cuts based on view cuts), the resultant magnitude.

Abaqus/CAE then computes *X–Y* data pairs by performing free body calculations at all pertinent time frames and creates *X–Y* data objects showing force, moment, or heat flow rate data against time. A separate *X–Y* data object is created for every force or moment and component combination you select; if you toggle on both the force and moment options and all four component options, Abaqus/CAE creates eight *X–Y* data objects for each active free body cut in your session. You can activate and deactivate free body cuts in the **Free Body Cut Manager**; for more information, see *Displaying, hiding, and highlighting free body cuts*.

1. Locate the **Free body** options.

   Select **Tools->XY Data->Create** from the main menu bar, and choose **Free body**.

   The **XY Data from Free Body** dialog box appears.
2. Click **Active Steps/Frames** to change the steps or frames from which Abaqus/CAE reads *X–Y* data. For more information, see *Activating and deactivating steps and frames*.
3. From the **Entity** options, toggle on **Force**, **Moment**, **Heat Flow Rate**, or all of these options to display *X–Y* data for the selected entity or entities. The **Heat Flow Rate** option is valid only for free body cuts based on view cuts.
4. From the **Force/Moment Components** options, toggle on the vector components that you want to include in the *X–Y* data set for forces and moments. Options include the resultant magnitude and the individual component values in each direction. For heat flow rate, only the resultant magnitude is included.
5. To evaluate and display the data, click **Plot**.

   An *X–Y* plot of the data you have configured in the dialog box appears in the current viewport.
6. To save the data you have configured as one or more *X–Y* data objects, click **Save**. *X–Y* data objects are saved only for the current session.

   Note:

   To plot your saved *X–Y* data, select **Tools->XY Data->Plot** from the main menu bar, and select the *X–Y* data from the menu.
7. When you have finished, click **Dismiss** to close the dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Displaying, hiding, and highlighting free body cuts*

Reading X–Y data from an ASCII file

You can read *X*–*Y* data from an ASCII text file that contains columns of numerical data separated by commas, tabs, or spaces.
The file can contain more than two columns of data; you can specify which columns (fields) correspond to the *X*- and *Y*-axis data. You can also specify which rows of data to read from the file.

1. Locate the **XY Data from ASCII File** options:

   From the main menu bar, select**Tools->XY Data->Create**. Click **ASCII file** in the dialog box that appears; then click **Continue**.

   Tip: You can also specify *X*–*Y* data by clicking **Create** in the **XY Data Manager** or by using the  tool in the toolbox.

   The **XY Data from ASCII File** dialog box appears.

2. Enter the name of your ASCII file in one of the following ways:

   - Type the name into the **File** window at the top of the dialog box, or
   - Click **Select** to filter and browse existing file names.

     The **ASCII File Selection** dialog box appears.

     Filter and browse existing files. When you locate the file you want, click the name to select it; then click **OK**.

3. Specify which columns of the file to read. Columns can be separated by spaces, tabs, or commas. Multiple consecutive spaces, tabs, and/or commas are interpreted as a single field delimiter.
   a. To specify the column within the file to read as *X*-values, enter an integer in the **Read X values from field** box. The default is field 1, the first column.
   b. To specify the column within the file to read as *Y*-values, enter an integer in the **Read Y values from field** box. The default is field 2, the second column.
4. To assign quantity types to the values in the *X*- or *Y*-columns, expand the **Quantity Types** list for the **X** or **Y** column and select quantity types from the lists that appear.
5. Specify which rows of the file to read; the default is all rows.

   - Click **Read all** to read every row of the file, or
   - Click **Skip**; then enter the number of file rows to skip in the **rows between reads** box. A value of 0 means read all rows; a value of 1 means read every other row. Reading always begins with the first row.

6. To evaluate and display the data, click **Plot**.

   An *X*–*Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.

7. To save the data you have configured, click **Save As**.

   Note: *X*–*Y* data objects have a precision of six significant digits; therefore, only six significant digits are retained when you save an *X*–*Y* data object.

   To plot your saved *X*–*Y* data object, select **Tools->XY Data->Plot** from the main menu bar and choose the *X*–*Y* data object from the pull-right menu.

8. When you have finished, click **Dismiss** to close the dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Understanding quantity types*

Entering X–Y data from the keyboard

You can specify *X*–*Y* data directly from the keyboard.
To do so, you type *X*- and *Y*-values into a simple table editor.

1. Locate the **XY Data from Keyboard** options:

   From the main menu bar, select**Tools->XY Data->Create**. Click **Keyboard** in the dialog box that appears; then click **Continue**.

   Tip: You can also specify *X*–*Y* data by clicking **Create** in the **XY Data Manager** or by using the  tool in the toolbox.

   The **XY Data from Keyboard** dialog box appears.

2. Use standard keyboard and mouse editing techniques to insert, modify, or delete *X*- and *Y*-values in the **XY Data from Keyboard** table. For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.)
3. To assign quantity types to the values in the *X*- or *Y*-columns, expand the **Quantity Types** list for the **X** or **Y** column and select quantity types from the lists that appear.
4. To evaluate and display the data, click **Plot**.

   An *X*–*Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.
5. To save the data you have configured, click **Save As**.

   Note: *X*–*Y* data objects have a precision of six significant digits; therefore, only six significant digits are retained when you save an *X*–*Y* data object.

   To plot your saved *X*–*Y* data object, select **Tools->XY Data->Plot** from the main menu bar and choose the *X*–*Y* data object from the pull-right menu.

6. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Understanding quantity types*

Saving an X–Y data object

Abaqus/CAE provides several dialog boxes to help you specify *X–Y* data.

Each of these dialog boxes contains a **Save As** button, which you can use to save the *X–Y* data object you create. You must save your *X–Y* data to do any of the following:

- Produce an *X–Y* plot containing multiple *X–Y* data objects from any source other than history output.
- Edit, copy, rename, delete, or operate on your *X–Y* data. For history output, you can operate on the data during the save procedure or after it has been saved.
- Establish persistent curve style preferences for your *X–Y* data.
- Produce a report of your *X–Y* data.
- Copy your *X–Y* data to an output database file.

Abaqus/CAE also saves the quantity types specified for each column in your *X–Y* data object, as long as the quantity types you select are among the preset quantity types available in the **XY Data from ASCII File**, **XY Data from Keyboard**, and **Edit XY Data** dialog boxes. If you derive a different quantity type by operating on *X–Y* data objects, Abaqus/CAE will not record the nonstandard quantity type in the saved *X–Y* data object.

Saved data persist only for the duration of the session. To save your *X–Y* data object beyond the duration of the session, you must save the data to a file. You can either copy your *X–Y* data object to an output database (binary) file or write it to a report (ASCII) file. For information on copying *X–Y* data to an output database file, see *Copying a session X–Y data object to an output database file*. For information on writing *X–Y* data to a report file, see *Generating tabular data reports*.

1. Specify your *X–Y* data.
   a. From the main menu bar, select **Tools->XY Data->Create**.

      The **Create XY Data** dialog box appears.

      Tip: You can also specify *X–Y* data by clicking **Create** in the **Data Manager** or by using the  tool in the toolbox.

   b. *Choose the source* of your data from the list, and click **Continue**.

      Abaqus displays the appropriate dialog box.

   c. Enter the necessary data or information in the dialog box.
2. Save your data for the current Abaqus session.
   a. Click **Save As** or **Save** (as available).

      If you are saving field data, do one of the following:

      - To save the data set without changing it, leave the default choice of **as is** in the **Save Operation** list.
      - You can choose to perform any of the *mathematical, trigonometric, logarithmic, exponential, or other operations* available in the **Save Operation** list. The selected operation will be performed on the data before the data set is saved.

      Note: After you perform operations on the data set, the *x*-axis and *y*-axis labels are redefined automatically based on the resulting dimensional changes. You can retain the default labels or customize them by selecting the desired quantity types from the **Quantity Types** menu in the **Edit XY Data** dialog box.

b. Accept the default name, or enter the data object name of your choice in the text field.



Note: If you save data from the **XY Data from ODB Field Output** dialog box, you must accept the default names. If you selected multiple history output variables in the **History Output** dialog box, each *X–Y* data set will be saved with the default name, which is a shortened form of the history output name.

c. If you are saving history data, do one of the following:

- To save the data set without changing it, leave the default choice of **as is** in the **Save Operation** list.
- If you selected a single history output variable, you can choose to perform any of the *mathematical, trigonometric, logarithmic, exponential, or other operations* available in the **Save Operation** list. The selected operation will be performed on the data before it is saved.
- If you selected two history output variables, you can choose to perform any of the *dual-operand operations* available in the **Save Operation** list:
  - *append((XY,XY))*
  - *avg((XY,XY))*
  - *combine((XY,XY))*
  - *maxEnvelope((XY,XY))*
  - *minEnvelope((XY,XY))*
  - *power((XY,XY))*
  - *rng((XY,XY))* (range)
  - *srss((XY,XY))* (square root of the sum of the squares)
  - *sum((XY,XY))*
- If you selected three or more history output variables, you can choose to perform any of the *multi-operand operations* available in the **Save Operation** list:
  - *append((XY,XY,...))*
  - *avg((XY,XY,...))*
  - *maxEnvelope((XY,XY,...))*
  - *minEnvelope((XY,XY,...))*
  - *rng((XY,XY,...))*
  - *srss((XY,XY,...))*
  - *sum((XY,XY,...))*
  - *vectorMagnitude((XY,XY,XY))*

d. If available, toggle on **Plot curves on OK** to plot the *X–Y* data sets when you click **OK**.

e. If available, toggle on **Swap variables** to swap the order of operations.

f. If available, toggle on **Apply to linked viewports** to save the *X–Y* data across all linked viewports.

The viewport number is appended to the *X–Y* data set name; for example, *XyData-1*_VP1, *XyData-1*_VP2, etc.

g. Click **OK** to close the dialog box.

3. Click **Dismiss** or **Cancel** (as available) to close the output dialog box.

Additional information

- *Understanding how to specify an X–Y data object*
- *Copying a session X–Y data object to an output database file*
- *Loading an X–Y data object to the current session*
- *Using X–Y data operations*
- *Linking viewports*

Copying a session X–Y data object to an output database file

You must save an *X–Y* data object to be able to manipulate the data within a session; however, saved data persist only for the duration of the session in which they were created. You must copy saved data to a file if you want it to be available in another session.
You can either copy your data to an output database file or write it to a report file. (For information on writing *X–Y* data to a report file, see *Generating tabular data reports*.) The output database file is binary, so it is efficient to save large amounts of data to this file for future use. You can only append data to the output database; you cannot modify the contents of the file in any other way. It is easy to read data into Abaqus/CAE from an output database file during a later session. You can copy your data to any output database file for which you have write privileges, including any currently open output database files.



Note: By default, Abaqus/CAE opens output database files as read-only. You must choose to open an output database file with write privileges if you want to copy any X–Y data objects to it (see *Opening a model database or an output database* for more information).

1. From the main menu bar, select **Tools->XY Data->Manager**.

   The **XY Data Manager** appears.

   Tip: You can also access the **XY Data Manager** from the  tool in the toolbox.

2. From the list of *previously saved* *X–Y* data objects, select the *X–Y* data object that you wish to copy, and click **Copy to ODB**.

   The **Copy Session XYData to ODB** dialog box appears.

3. Enter the file name of your choice in the appropriate text field, or click **Select** to choose from an existing file name.
4. Accept the default name for the *X–Y* data object, or enter a name of your choice in the text field; then click **OK**.

Additional information

- *Understanding how to specify an X–Y data object*
- *Saving an X–Y data object*
- *Loading an X–Y data object to the current session*

Loading an X–Y data object to the current session

When you create an *X–Y* data object, you must save it to be able to manipulate the data within a session. You can then copy the *X–Y* data object to an output database file so that you can access it from a later session. The **XY Data Manager** lists the *X–Y* data objects that are available from two sources: the current session or the current output database file. If you have created an *X–Y* data object in a previous session and copied it to an output database file, you must load the *X–Y* data object to the current session to manipulate it.

1. From the main menu bar, select **Tools->XY Data->Manager**.

   The **XY Data Manager** appears.

    **Tip:** You can also access the **XY Data Manager** from the  tool in the toolbox.

2. Toggle on **Current ODB:** ***odb_name*** as the **Data Source** to see a list of the *X–Y* data objects associated with the currently open output database file.
3. Select an *X–Y* data object to load.
4. Click **Load to Session**.
5. Toggle on **Current session** as the **Data Source**.

   The loaded *X–Y* data object will be listed as available; you can now edit, copy, rename, or delete this *X–Y* data object.

Additional information

- *Understanding how to specify an X–Y data object*
- *Saving an X–Y data object*
- *Copying a session X–Y data object to an output database file*

Editing individual data points in an X–Y data object

You can edit individual records in any *X–Y* data object in your session, including temporary data objects. Abaqus/CAE presents the data in tabular format in the **Edit XY Data** dialog box.

1. Locate the **Edit XY Data** options:

   From the main menu bar, select **Tools**->**XY Data**->**Edit**->***XY Data Object***, where ***XY Data Object*** is a saved *X–Y* data object in your session. The **Edit XY Data** dialog box appears with the *X–Y* data presented in tabular format.
2. To edit any of the data points, click in the appropriate cell and change the value.
3. To assign quantity types to the values in the *X*- or *Y*-columns, expand the **Quantity Types** list for the **X** or **Y** column and select quantity types from the lists that appear.
4. To save the data you have configured, click **Save**.

Additional information

- *Saving an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Understanding quantity types*

Producing an X–Y plot

To produce an *X–Y* plot, you first specify an *X–Y* data object. You can then choose to simply plot the specified *X–Y* data object or to save the data object and plot it later.

If viewport linking is activated, you can link plot display across viewports.

History and field data objects can include complex numbers; you can control the displayed form of complex numbers using the **Result Options** dialog box. An abbreviation of the complex form is appended to the *Y*-axis title when you plot the variable and becomes part of the saved variable name. For example, if `S-Mises` is the selected field output variable and `Magnitude` is the complex form, the *Y*-axis title is `S-Mises CPX: Mg`. For more information on complex number forms in Abaqus/CAE, see *Controlling the form of complex results*.

Use one of the following methods to produce an *X–Y* plot:

To produce an X–Y plot of history data from an output database:

Select **Tools->XY Data->Create** from the main menu bar to launch the **Create XY Data** dialog box; then choose **ODB history output**. Click **Continue**. From the dialog box that appears, select one or more variables to plot and the step or steps of interest; then click **Plot**.



Tip: To launch the **Create XY Data** dialog box from the Results Tree, click mouse button 3 on the **XYData** container and select **Create** from the list that appears.

You can also specify history output by selecting **Result->History Output** from the main menu bar.

To produce an X–Y plot of field data from an output database:

Select **Tools->XY Data->Create** from the main menu bar; then choose **ODB field output**. Click **Continue**. From the dialog box that appears, select one or more variables to plot, the location or locations from which to read the data, and the step or steps of interest; then click **Plot**.

To produce an X–Y plot of output database results along a path through your model:

To specify a path through your model, select **Tools->Path->Create** from the main menu bar; then configure the dialog boxes that appear. For more information, see *Creating a path through your model*.

After you have created the path, select **Tools->XY Data->Create** from the main menu bar. Choose **Path** as the source of the *X–Y* data object, then click **Continue**. Configure the dialog box that appears, then click **Plot**. For more information, see *Obtaining X–Y data along a path*.

To produce an X–Y plot of new data, not from an output database:

Select **Tools->XY Data->Create** from the main menu bar. Choose either **Operate on XY data**, **ASCII file**, or **Keyboard** as the source of the *X–Y* data object; then click **Continue**. In the dialog box that appears, enter your *X–Y* data object; then click **Plot** (or **Plot Expression** in the **Operate on XY Data** dialog box).

To produce an X–Y plot of one or more saved X–Y data objects:

- To display a single *X–Y* data object, select **Tools->XY Data->Plot** from the main menu bar; and select the *X–Y* data object from the pull-right menu.
- To display multiple *X–Y* data objects on a single *X–Y* plot, select **Tools->XY Data->Manager** from the main menu bar. Select the *X–Y* data objects from the dialog box that appears; then click **Plot**.

You can also perform either of these actions from the Results Tree. To display a single *X–Y* data object using the Results Tree, expand the **XYData** container, click mouse button 3 on the *X–Y* data object you want to plot, and select **Plot** from the list that appears. To display multiple *X–Y* data objects using the Results Tree, highlight multiple *X–Y* objects under the **XYData** container, click mouse button 3 on one of the highlighted objects, and select **Plot** from the list that appears.

Additional information

- *Understanding how to specify an X–Y data object*
- *Understanding "Temp" and other X–Y data object names*
- *Controlling the form of complex results*
- *Linking viewports*

Operating on saved X–Y data objects

You can derive new *X–Y* data by operating on previously saved *X–Y* data objects. This section discusses how to operate on *X–Y* data and presents each of the possible operations you can perform.

You can also operate on history data during the save procedure. For more information, see *Saving an X–Y data object*.

In this section:

- *Understanding how to operate on saved X–Y data objects*
- *Understanding X–Y data interpolation and extrapolation*
- *Operating on saved X–Y data objects*
- *Using X–Y data operations*
- *Using addition on X–Y data objects*
- *Using negation or subtraction on X–Y data objects*
- *Using multiplication on X–Y data objects*
- *Using division on X–Y data objects*
- *Taking the absolute value of an X–Y data object*
- *Finding the average of two or more X–Y data objects*
- *Finding the current average of an X–Y data object*
- *Differentiating an X–Y data object*
- *Performing curve fitting on an X–Y data object*
- *Integrating an X–Y data object*
- *Interpolating an X–Y data object*
- *Linearizing an X–Y data object*
- *Normalizing an X–Y data object*
- *Taking the square root of an X–Y data object*
- *Taking the square root of the sum of the squares of two or more X–Y data objects*
- *Summing two or more X–Y data objects*
- *Calculating vector magnitudes*
- *Applying trigonometric functions to an X–Y data object*
- *Taking the exponential of an X–Y data object*
- *Applying logarithmic functions to an X–Y data object*
- *Raising an X–Y data object to a power*
- *Applying Butterworth filtering to an X–Y data object*
- *Applying Chebyshev Type I or Type II filtering to an X–Y data object*
- *Reducing the sample size of an X–Y data object*
- *Applying SAE filtering to an X–Y data object*
- *Applying sine-Butterworth filtering to an X–Y data object*
- *Smoothing an X–Y data object*
- *Finding the current maximum of an X–Y data object*
- *Finding the current minimum of an X–Y data object*
- *Finding the current range of an X–Y data object*
- *Finding the current maximum of two or more X–Y data objects*
- *Finding the current minimum of two or more X–Y data objects*
- *Finding the current range of two or more X–Y data objects*

- *Appending two or more X–Y data objects*
- *Combining two X–Y data objects*
- *Converting the angular units used for X–Y data objects*
- *Swapping the order of an X–Y data object*
- *Truncating data from the ends of an X–Y data object*

Understanding how to operate on saved X–Y data objects

You can derive new *X–Y* data by operating on *previously saved* *X–Y* data objects. You define your new data by building a mathematical expression. The expression can include the names of previously saved *X–Y* data objects, built-in functions, and mathematical operators. An example of an expression might be: `currentMax("XYData-1")+2.5`. Abaqus/CAE evaluates the expression to derive the new *X–Y* data.

To build your expression, you use the ***Operate on XY Data*** dialog box. This dialog box lists the data object names and operators available for your expression. You can click to select listed data objects and operators; type values in using the keyboard; and use standard mouse and keyboard editing techniques such as backspace, copy, and paste to configure your expression. The expression can contain any syntactically valid series of supported operations. The following syntax rules apply:

- Multiple arguments to a function must be separated by commas.
- Data object names must be surrounded by quotation marks.
- Parentheses must be used to group function arguments.

Parentheses can also be used to control mathematical evaluation or for visual clarity. Abaqus/CAE notifies you if your expression contains invalid syntax or cannot be evaluated for mathematical reasons; for example, if evaluation would require dividing by zero.



Tip: When you select multiple variables in the **History Output** dialog box and click **Save As**, you can choose to apply one of the more common operators to the *X–Y* data before saving it.

By default, Abaqus/CAE performs checks to validate the mathematical expressions you create. If your expression includes a large number of *X–Y* data points, this validation can be time consuming. You can toggle on **Skip checks** to skip this set of checking procedures.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Operating on saved X–Y data objects*
- *Using X–Y data operations*

Understanding X–Y data interpolation and extrapolation

Abaqus/CAE performs *X–Y* data interpolation and extrapolation as necessary when you combine multiple *X–Y* data objects. You can combine multiple *X–Y* data objects by using *X–Y* data operations or by requesting that multiple data objects appear in a single table within an *X–Y report*.

A complete list of *X–Y* data operations appears in *Using X–Y data operations*. Several of the *X–Y* data operations that accept two or more *previously saved*X–Y data objects as input arguments require that the data objects have matching *X*-coordinate values. Examples of such operations are the **combine** and **sum** functions.

Similarly, matching *X–Y* values are required for data objects appearing in the same *X–Y* report table. For information on combining *X–Y* data objects into a single report table, see *Controlling report layout, width, format, and coordinate system*.

If the *X*-coordinate values to be combined do not match, Abaqus/CAE computes additional data points to allow the objects to be aligned. For each unmatched *X*-coordinate value, Abaqus/CAE constructs a data pair by creating a matching *X*-coordinate value and computing a *Y*-coordinate value as follows:

- If the missing *X*-coordinate lies within the minimum and maximum available *X*-coordinates for the data object, a new *Y*-coordinate value is computed by linear interpolation.
- If the missing *X*-coordinate lies beyond the minimum or maximum available *X*-coordinates for the data object, a new *Y*-coordinate value is computed by assuming the *Y*-coordinate value remains constant beyond these extreme points.

These additional data points exist only to carry out the operation and are not saved as part of the input argument data objects.

Additional information

- *Using X–Y data operations*
- *Generating tabular data reports*

Operating on saved X–Y data objects

To define a new *X–Y* data object by operating on *previously saved X–Y* data objects, select **Tools->XY Data->Create** from the main menu bar; then choose **Operate on XY data** from the dialog box that appears.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. Build your expression:

   a. Click on data object names and built-in functions in the dialog box to select them. Click **Add to Expression** near the bottom of the dialog box to add data objects to the expression; the built-in functions appear in the expression automatically when you select them.

      Use drag-select for multiple contiguous data object name selection.

      Commas are inserted automatically between the data object names.

   b. Use standard mouse and keyboard editing techniques to position the cursor and configure your expression.

   c. Adjust the syntax of your expression if necessary: data object names must be surrounded by quotation marks, commas must separate function arguments, and parentheses may be needed.

3. To save your new *X–Y* data object, click **Save As**.

   The **Save XY Data As** dialog box appears. Provide a name for your data; then click **OK**.

4. To plot the expression you have configured, click **Plot Expression**.

   An *X–Y* plot appears in the current viewport. The plot represents the expression in the dialog box, which Abaqus considers temporary data, whether or not you have saved your new *X–Y* data object.

5. To plot your new, saved *X–Y* data object, select **Tools->XY Data->Plot** from the main menu bar and select the *X–Y* data object from the pull-right menu.

6. To clear (erase) the expression, click **Clear Expression**.

7. When you are done, click **Cancel** to close the dialog box.

Additional information

- *Understanding how to operate on saved X–Y data objects*
- *Using X–Y data operations*
- *Saving an X–Y data object*

Using X–Y data operations

You can derive new *X–Y* data by operating on *previously saved**X–Y* data objects. This overview lists the operations you can perform on *X–Y* data. You can also operate on history data during the save procedure. For more information, see *Saving an X–Y data object*.

The following keys are used to classify function arguments:

Text keys for X–Y operations:

A	The argument can be an *X–Y* data object, a floating point number, or an integer.
X	The argument must be an *X–Y* data object.
I	The argument must be an integer.
F	The argument must be a floating point number.

The following list indicates the section where you can find more information for each *X–Y* data operation.

Mathematical operations:

+	*Using addition on X–Y data objects*
–	*Using negation or subtraction on X–Y data objects*
*****	*Using multiplication on X–Y data objects*
/	*Using division on X–Y data objects*
1/A	*Using division on X–Y data objects*
abs(A)	*Taking the absolute value of an X–Y data object*
avg(X,X,...)	*Finding the average of two or more X–Y data objects*
currentAvg(X)	*Finding the current average of an X–Y data object*
differentiate(X)	*Differentiating an X–Y data object*
fit(X)	*Performing curve fitting on an X–Y data object*
integrate(X)	*Integrating an X–Y data object*
interpolate(X)	*Interpolating an X–Y data object*
linearize(X,CONSTANT)	*Linearizing an X–Y data object*
linearize(X,LINEAR)	*Linearizing an X–Y data object*
normalize(X)	*Normalizing an X–Y data object*
sqrt(A)	*Taking the square root of an X–Y data object*
srss(X,X,...)	*Taking the square root of the sum of the squares of two or more X–Y data objects*
sum(A,A,...)	*Summing two or more X–Y data objects*
vectorMagnitude(X,X,X)	*Calculating vector magnitudes*

Trigonometric operations:

For more information, see *Applying trigonometric functions to an X–Y data object*.

cos(A)	Take the cosine of an *X–Y* data object.
acos(A)	Take the arccosine of an *X–Y* data object.
cosh(A)	Take the hyperbolic cosine of an *X–Y* data object.

sin(A)	Take the sine of an *X–Y* data object.
asin(A)	Take the arcsine of an *X–Y* data object.
sinh(A)	Take the hyperbolic sine of an *X–Y* data object.
tan(A)	Take the tangent of an *X–Y* data object.
atan(A)	Take the arctangent of an *X–Y* data object.
tanh(A)	Take the hyperbolic tangent of an *X–Y* data object.

Logarithmic and exponential operations:

exp(A)	*Taking the exponential of an X–Y data object*
log(A)	*Applying logarithmic functions to an X–Y data object*
log10(A)	*Applying logarithmic functions to an X–Y data object*
power(A,A)	*Raising an X–Y data object to a power*

Filtering and smoothing operations:

butterworthFilter(X,F)	*Applying Butterworth filtering to an X–Y data object*
chebyshev1Filter(X,F,F)	*Applying Chebyshev Type I or Type II filtering to an X–Y data object*
chebyshev2Filter(X,F,F)	*Applying Chebyshev Type I or Type II filtering to an X–Y data object*
decimateFilter(X,I)	*Reducing the sample size of an X–Y data object*
saeGeneralFilter(X,F)	*Applying SAE filtering to an X–Y data object*
sae60Filter(X,F)	*Applying SAE filtering to an X–Y data object*
sae100Filter(X,F)	*Applying SAE filtering to an X–Y data object*
sae180Filter(X,F)	*Applying SAE filtering to an X–Y data object*
sae600Filter(X,F)	*Applying SAE filtering to an X–Y data object*
sae1000Filter(X,F)	*Applying SAE filtering to an X–Y data object*
sineButterworthFilter(X,F)	*Applying sine-Butterworth filtering to an X–Y data object*
smooth(X,I)	*Smoothing an X–Y data object*
smooth2(X,F)	*Smoothing an X–Y data object*

Range and magnitude related operations:

currentMax(X)	*Finding the current maximum of an X–Y data object*
currentMin(X)	*Finding the current minimum of an X–Y data object*
currentRng(X)	*Finding the current range of an X–Y data object*
maxEnvelope(A,A,...)	*Finding the current maximum of two or more X–Y data objects*
minEnvelope(A,A,...)	*Finding the current minimum of two or more X–Y data objects*
rng(A,A,...)	*Finding the current range of two or more X–Y data objects*

Other operations:

append(X,X,...)	*Appending two or more X–Y data objects*
combine(X,X)	*Combining two X–Y data objects*

degreeToRadian(A)	*Converting the angular units used for X–Y data objects*
radianToDegree(A)	*Converting the angular units used for X–Y data objects*
swap(X)	*Swapping the order of an X–Y data object*
truncate(X,F)	*Truncating data from the ends of an X–Y data object*

Additional information

- *Understanding how to operate on saved X–Y data objects*
- *Operating on saved X–Y data objects*

Using addition on X–Y data objects

You can use addition on previously saved X–Y data objects to produce a new X–Y data object.

Use the mathematical **+** symbol to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The **+** operator accepts two commutative arguments and can be applied in one of two ways:

To add a scalar to an X–Y data object:

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as the scalar added to each original Y-coordinate. For example, if

$$\mathbf{XYData} = [(1,1),(2,2),(3,3)],$$

then

$$(\mathbf{XYData} + 5) = [(1,6),(2,7),(3,8)].$$

To add an X–Y data object to another X–Y data object:

This method yields a new X–Y data object having as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the two objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the Y-coordinates of the first data object added to the Y-coordinates of the second data object. For example, let

$$\mathbf{XYData1} = [(1,1),(2,2),(3,3)],$$

and

$$\mathbf{XYData2} = [(4,4),(5,5)].$$

For alignment, Abaqus/CAE computes the values of the first and second data objects as:

$$\mathbf{XYData1\ extrapolated} = [(1,1),(2,2),(3,3),(4,3),(5,3)],$$

and

$$\mathbf{XYData2\ extrapolated} = [(1,4),(2,4),(3,4),(4,4),(5,5)];$$

then

$$(\mathbf{XYData1} + \mathbf{XYData2}) = [(1,5),(2,6),(3,7),(4,7),(5,8)].$$

Figure 1 illustrates the above example.

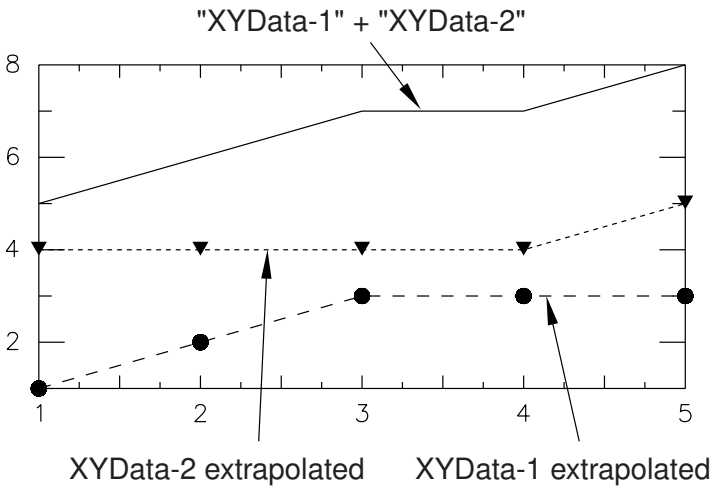


Figure 1: *X–Y* plot illustrating addition of data objects.

Note:

This application of the **+** symbol has the same behavior as the **sum** function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the expression window.

3. From the **Operators** listed, click **+**.

   The **+** symbol appears after the data object name in the expression window.

4. To specify the second argument, do one of the following:

   - Use your mouse and keyboard to enter a scalar as the second argument of the **+** operator in the expression window, or
   - From the **XY Data** choices, click the name of a data object argument for the **+** operator in the expression window and click **Add to Expression**.

5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

8. When you are finished, click **Cancel** to close the dialog box.

Additional information

• *Understanding X–Y data interpolation and extrapolation*

- *Summing two or more X–Y data objects*
- *Appending two or more X–Y data objects*
- *Combining two X–Y data objects*
- *Using X–Y data operations*

Using negation or subtraction on X–Y data objects

Use the mathematical **–** symbol to perform negation or subtraction on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object.

To perform negation, precede a single X–Y data object with the **–** operator. Negation yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as the negative of each original Y-coordinate.

To perform subtraction, use one of three methods:

To subtract a scalar from an X–Y data object:

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as the scalar subtracted from each original Y-coordinate. For example, if

$$\mathbf{XYData} = [(1,6),(2,7),(3,8)],$$

then

$$(\mathbf{XYData} - 5) = [(1,1),(2,2),(3,3)].$$

To subtract an X–Y data object from a scalar:

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as each original Y-coordinate subtracted from the scalar. For example, if

$$\mathbf{XYData} = [(1,6),(2,7),(3,8)],$$

then

$$(5 - \mathbf{XYData}) = [(1,-1),(2,-2),(3,-3)].$$

To subtract one X–Y data object from another X–Y data object:

This method yields a new X–Y data object having as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the two objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the Y-coordinates of the second data object subtracted from the Y-coordinates of the first data object. For example, let

$$\mathbf{XYData1} = [(4,4),(5,5)]$$

and

$$\mathbf{XYData2} = [(1,1),(2,2),(3,3)].$$

For alignment, Abaqus/CAE computes the values of the first and second data objects as:

$$\mathbf{XYData1\ extrapolated} = [(1,4),(2,4),(3,4),(4,4),(5,5)]$$

and

$$\mathbf{XYData2\ extrapolated} = [(1,1),(2,2),(3,3),(4,3),(5,3)];$$

then

$$(\mathbf{XYData1} - \mathbf{XYData2}) = [(1,3),(2,2),(3,1),(4,1),(5,2)].$$

Figure 1 shows an *X–Y* plot of the above example.

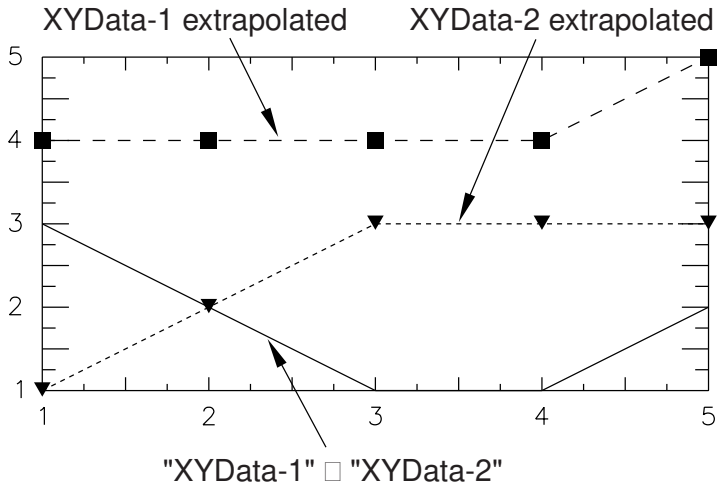


Figure 1: *X–Y* plot illustrating subtraction of data objects.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **–**.

   The **–** symbol appears in the expression window.

3. Use your mouse and keyboard to position the cursor in the expression window; then specify the necessary arguments. Available *X–Y* data objects include all those previously saved within this session (listed alphabetically in the **XY Data** field).

To use negation

From the **XY Data** choices, click the name of the *X–Y* data object to negate and click **Add to Expression**.

To use subtraction

- Use your keyboard to type in a scalar argument, or
- From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**.

The arguments appear within the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Using multiplication on X–Y data objects

Use the mathematical * symbol to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The * operator accepts two commutative arguments and can be applied in one of two ways:

To multiply a scalar and an X–Y data object:

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as the scalar multiplied by each original Y-coordinate. For example, if

$$\mathbf{XYData} = [(1,1),(2,2),(3,3)],$$

then

$$(\mathbf{XYData} * 5) = [(1,5),(2,10),(3,15)].$$

To multiply an X–Y data object with another X–Y data object:

This method yields a new X–Y data object having as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the two objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the Y-coordinates of the first data object multiplied by the Y-coordinates of the second data object. For example, let

$$\mathbf{XYData1} = [(1,1),(2,2),(3,3)]$$

and

$$\mathbf{XYData2} = [(4,4),(5,5)].$$

For alignment, Abaqus/CAE computes the values of the first and second data objects as:

$$\mathbf{XYData1\ extrapolated} = [(1,1),(2,2),(3,3),(4,3),(5,3)]$$

and

$$\mathbf{XYData2\ extrapolated} = [(1,4),(2,4),(3,4),(4,4),(5,5)];$$

then

$$(\mathbf{XYData1} * \mathbf{XYData2}) = [(1,4),(2,8),(3,12),(4,12),(5,15)].$$

Figure 1 shows an X–Y plot of the preceding example.

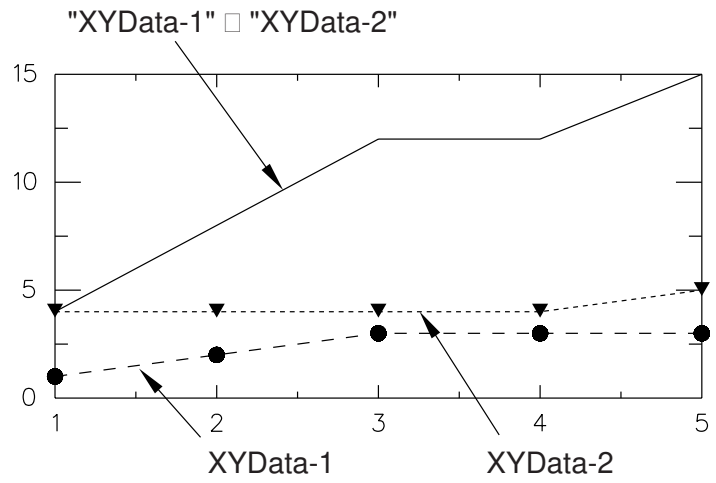


Figure 1: X–Y plot illustrating multiplication of data objects.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the expression window.

3. From the **Operators** listed, click *****.

   The * symbol appears after the data object name in the expression window.

4. To specify the second argument, do one of the following:

   - Use your mouse and keyboard to enter a scalar as the second argument of the ***** operator in the expression window, or
   - From the **XY Data** choices, click the name of a data object argument for the * operator in the expression window and click **Add to Expression**.

5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Using division on X–Y data objects

Use the mathematical **/** symbol to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The **/** operator accepts two non-commutative arguments and can be applied in one of three ways:

To divide a scalar by an X–Y data object:

This method yields a new *X–Y* data object having the same *X*-coordinate values as the original *X–Y* data object. Abaqus/CAE computes the new *Y*-coordinates as the scalar divided by each original *Y*-coordinate. For example, if

$$\mathbf{XYData} = [(1, 2), (2, 10), (3, 20)],$$

then

$$(2 \,/\, \mathbf{XYData}) = [(1, 1), (2, .2), (3, .1)].$$



Note:

As a convenience, the **Operate on XY Data** dialog box offers an inverse function, **1/A**.

To divide an X–Y data object by a scalar:

This method yields a new *X–Y* data object having the same *X*-coordinate values as the original *X–Y* data object. Abaqus/CAE computes the new *Y*-coordinates as each original *Y*-coordinate divided by the scalar. For example, if

$$\mathbf{XYData} = [(1, 2), (2, 10), (3, 20)],$$

then

$$(\mathbf{XYData} \,/\, 2) = [(1, 1), (2, 5), (3, 10)].$$

To divide one X–Y data object by another X–Y data object:

This method yields a new *X–Y* data object having as its *X*-coordinates all *X*-coordinates of the first data object and any additional *X*-coordinates needed for alignment of the two objects. Abaqus/CAE computes additional *X–Y* data pairs by *interpolation and extrapolation*. The new data object has as its *Y*-coordinates the *Y*-coordinates of the first data object divided by the *Y*-coordinates of the second data object. For example, let

$$\mathbf{XYData1} = [(4, 4), (5, 5)]$$

and

$$\mathbf{XYData2} = [(1, 1), (2, 2), (3, 3)].$$

For alignment, Abaqus/CAE computes the values of the first and second data objects as:

$$\mathbf{XYData1\ extrapolated} = [(1, 4), (2, 4), (3, 4), (4, 4), (5, 5)]$$

and

$$\mathbf{XYData2\ extrapolated} = [(1, 1), (2, 2), (3, 3), (4, 3), (5, 3)];$$

then

$$(\mathbf{XYData1} \,/\, \mathbf{XYData2}) = [(1, 4), (2, 2), (3, 1.33), (4, 1.33), (5, 1.66)].$$

Figure 1 shows an *X–Y* plot of the above example.

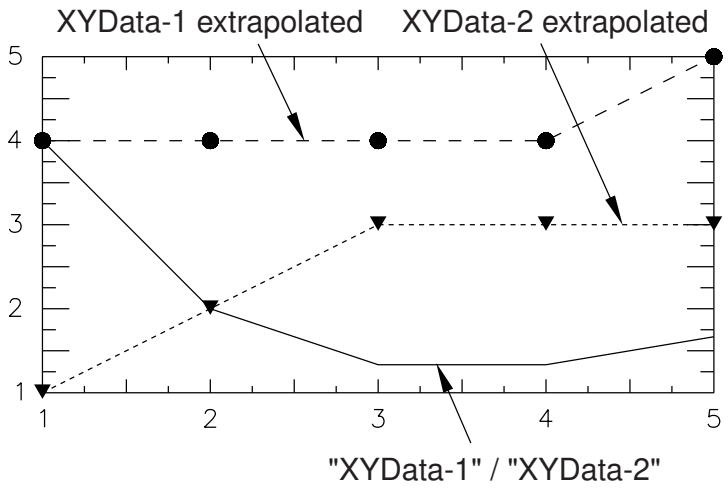


Figure 1: X–Y plot illustrating division of data objects.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **/**.

   The **/** symbol appears in the expression window.

3. To specify each of the two arguments for the **/** operator, use your mouse and keyboard to position the cursor in the expression window; then do one of the following:

   - Use your keyboard to type in a scalar argument, or
   - From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The arguments appear within the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Taking the absolute value of an X–Y data object

Use the **abs** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the absolute value of the original Y-coordinates.

Figure 1 shows an example of an X–Y plot produced using the **abs** function.

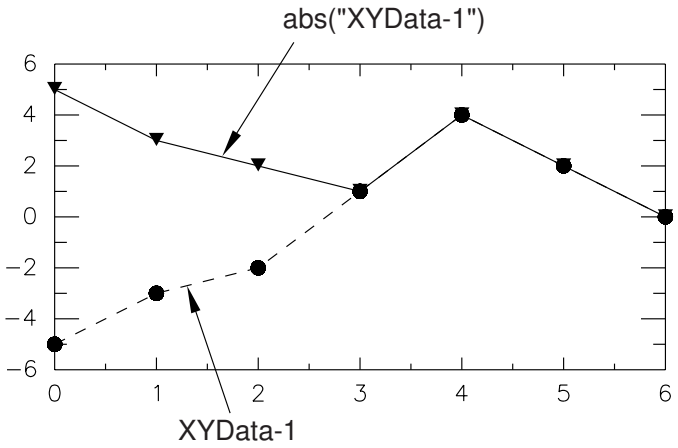


Figure 1: X–Y plot produced using the abs function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **abs(A)**.

   The **abs** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **abs** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*

Finding the average of two or more X–Y data objects

Use the **avg** function to operate on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has as its *X*-coordinates all *X*-coordinates of the input data objects and any additional *X*-coordinates needed for alignment of the input data objects. Abaqus/CAE computes additional *X–Y* data pairs by *interpolation and extrapolation*. Abaqus/CAE computes the new data object *Y*-coordinates as the average of all input *Y*-coordinates at the current *X*-coordinate. The arguments of this operation are commutative.

Figure 1 illustrates the type of *X–Y* plot that can be produced using the **avg** function.

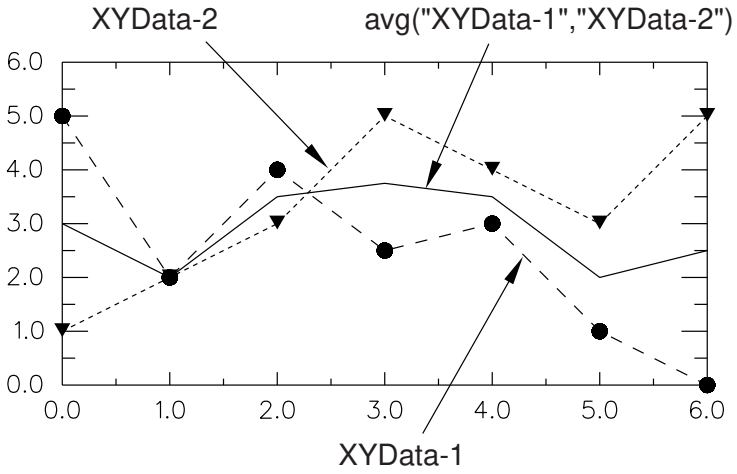


Figure 1: *X–Y* plot produced using the avg function.

For information on how to find the current average of a single *X–Y* data object, see *Finding the current average of an X–Y data object*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **avg(X,X,...)**.

   The **avg** function appears in the expression window.

3. From the **XY Data** choices, click the names of two or more *X–Y* data objects on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object names appear, separated by commas, within the **avg** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current average of an X–Y data object*
- *Using X–Y data operations*

Finding the current average of an X–Y data object

Use the **currentAvg** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has the same *X*-coordinate values as the original *X–Y* data object, but new *Y*-coordinates are computed as the average of all *Y*-coordinates up to and including the current *X*-coordinate.

Figure 1 shows an example of an *X–Y* plot produced using the **currentAvg** function.

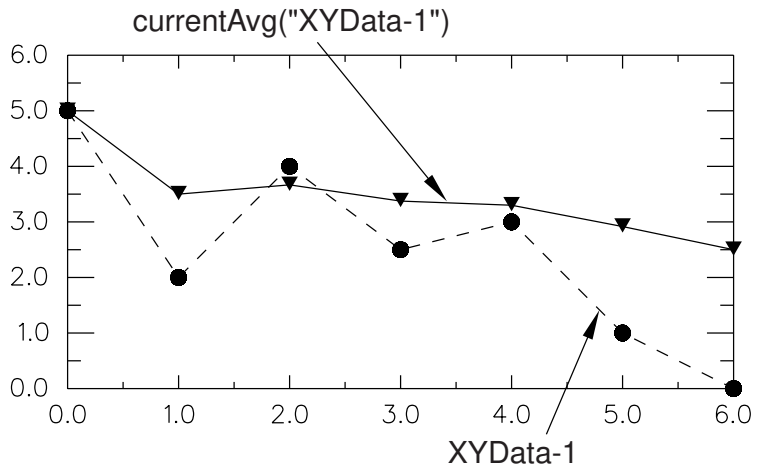


Figure 1: *X–Y* plot produced using the currentAvg function.

Note:

The last value of the new *X–Y* data object represents the overall average *Y*-coordinate of the original *X–Y* data object.

For information on how to find the current average of two or more *X–Y* data objects, see *Finding the average of two or more X–Y data objects*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **currentAvg(X)**.

   The **currentAvg** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **currentAvg** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the average of two or more X–Y data objects*
- *Using X–Y data operations*

Differentiating an X–Y data object

Use the **differentiate** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has the same *X*-coordinate values as the original *X–Y* data object, but new *Y*-coordinates are computed as the numerical derivative of the original *Y*-coordinates.

Abaqus/CAE computes the derivative using three-point parabolic segments of the original *X–Y* data object. The values at the first and last points are computed as the gradient at the start and at the end of the associated parabolas, respectively. The slope at each remaining point is computed using the midpoint of the parabola defined by two neighboring points. *Figure 1* shows an example of an *X–Y* plot produced using the **differentiate** function.

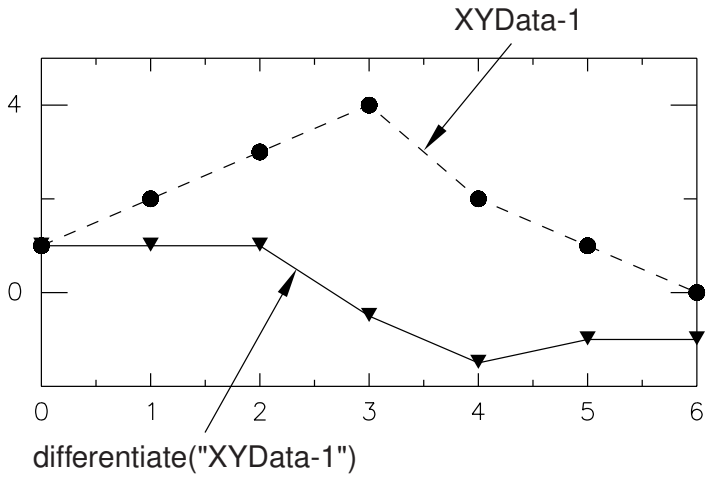


Figure 1: *X–Y* plot produced using the differentiate function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **differentiate(X)**.

   The **differentiate** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **differentiate** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Integrating an X–Y data object*

- *Using X–Y data operations*

Performing curve fitting on an X–Y data object

Use the **fit** function to operate on a *previously saved*$X–Y$ data object (a collection of ordered pairs) to produce a new $X–Y$ data object. The new $X–Y$ data object contains points that are spaced at regular intervals along the curve that best fits the previously saved $X–Y$ data object. The **fit** function provides two algorithms for curve fitting: a linear least squares fit and a spline interpolation.

Figure 1 and *Figure 2* show examples of $X–Y$ plots produced using the **fit** function using each curve fitting algorithm.

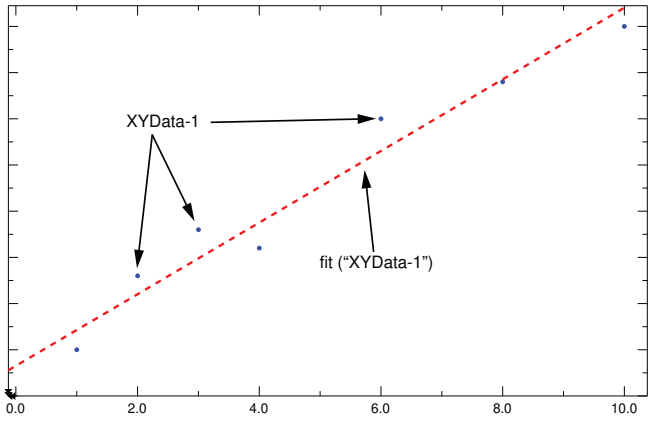


Figure 1: $X–Y$ plot produced using the fit function with the linear least squares option selected.

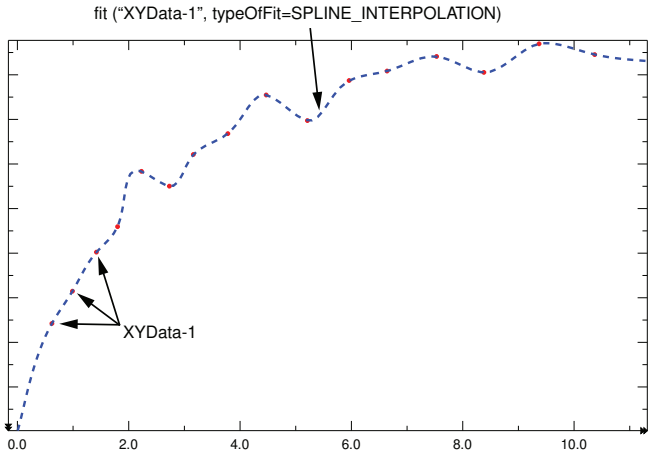


Figure 2: $X–Y$ plot produced using the fit function with the spline interpolation option selected.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **fit(X)**.

   The **fit** function appears in the expression window with the linear least squares fit option selected by default.

3. From the **XY Data** choices, click the name of the $X–Y$ data object on which to operate and click **Add to Expression**. You can choose from all $X–Y$ data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The $X–Y$ data object name appears within the **fit** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include. You can customize the **fit** function itself by including the following changes to its options:
   - If you want to perform a spline interpolation, set the `typeOfFit` option to `SPLINE_INTERPOLATION`.
   - If you want to specify a custom number of points to be created for the new *X–Y* data object, append `numFitPoints=`*number*. By default, Abaqus/CAE uses 2 points to define a linear least squares fit and 100 points to define a spline interpolation.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Integrating an X–Y data object*
- *Using X–Y data operations*

Integrating an X–Y data object

Use the **integrate** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has the same *X*-coordinate values as the original *X–Y* data object, but new *Y*-coordinates are computed as the numerical integral (using the trapezoidal rule) of the original *Y*-coordinates.

Figure 1 shows an example of an *X–Y* plot produced using the **integrate** function.

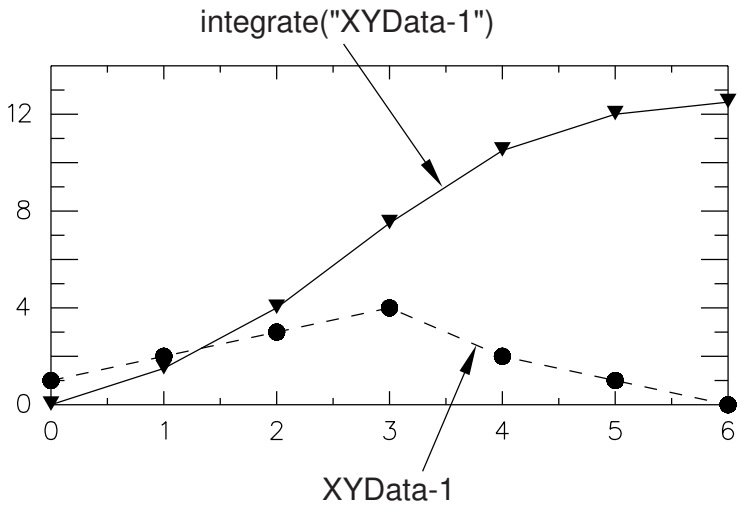


Figure 1: *X–Y* plot produced using the integrate function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **integrate(X)**.

   The **integrate** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **integrate** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Differentiating an X–Y data object*
- *Using X–Y data operations*

Interpolating an X–Y data object

Use the **interpolate** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. Interpolating an *X–Y* data object with nonuniform data produces an *X–Y* data object with new, interpolated *Y*-coordinate values occurring at regular increments on the *X*-axis. By default, the data are interpolated at increments equal to the smallest *X*-increment in the source data.

Figure 1 shows an example of an *X–Y* plot produced using the **interpolate** function, superimposed over the raw data.

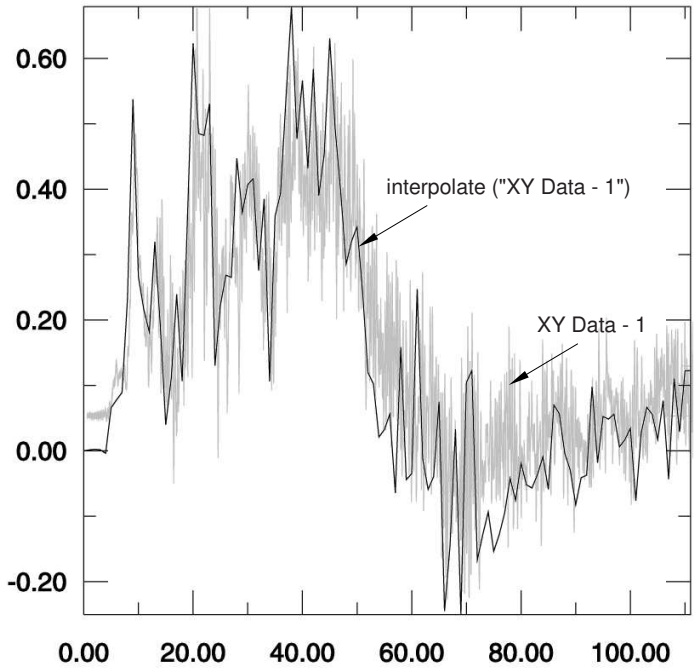


Figure 1: *X–Y* plot produced using the interpolate function.

The name of the *X–Y* data object is the only required argument for the **interpolate** function. A description of the optional arguments follows:

- The *X*-axis increment between interpolated data points (*xIncrement*). If you do not specify a value for this argument or if you specify a non-positive value, Abaqus/CAE uses the minimum *X*-increment of the raw data set.
- A symbolic constant that specifies the interpolation scheme (*interpolation*). Valid values for this argument are QUADRATIC, specifying a Lagrange second-order interpolation scheme; CUBIC_SPLINE, specifying a cubic spline interpolation scheme; and LINEAR, specifying a linear interpolation scheme. The default value is QUADRATIC.
- The slope of the raw data curve leading up to the first data point (*startslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- The slope of the raw data curve continuing past the final data point (*endslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **interpolate(X)**.

   The **interpolate** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **interpolate** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*

Linearizing an X–Y data object

Use the **linearize** function to separate the *Y*-values of a *previously saved*$X–Y$ data object (a collection of ordered pairs) into constant or linear components. The new $X–Y$ data object produced by the **linearize** function contains exactly two points.

Figure 1 shows an example of an $X–Y$ plot produced using the **linearize** function.

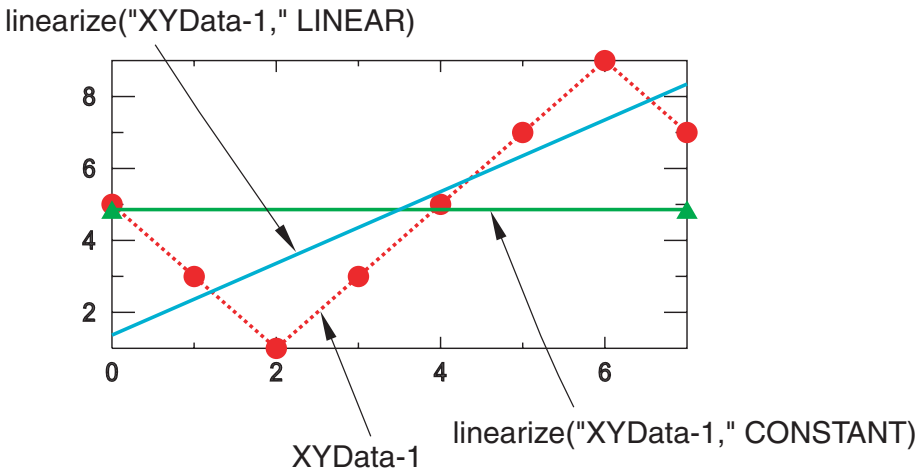


Figure 1: $X–Y$ plot produced using both the constant and linear forms of the linearize function.

Note:

The *X*-values of the $X–Y$ data object that you input to the **linearize** function must be equally spaced.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click either **linearize(X, CONSTANT)** or **linearize(X, LINEAR)** to apply linearization to create a constant Y term or to linearize the Y term, respectively.

   The **linearize** function appears in the expression window.

3. From the **XY Data** choices, click the name of the $X–Y$ data object on which to operate and click **Add to Expression**. You can choose from all $X–Y$ data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The $X–Y$ data object name appears within the **linearize** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and $X–Y$ data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new $X–Y$ data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in $X–Y$ plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

• *Using X–Y data operations*

Normalizing an X–Y data object

Use the **normalize** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. This operation is analogous to normalizing a vector. The new X–Y data object has the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates by dividing each original Y-coordinate by the root mean square (the square root of the sum of the squares) of all original Y-coordinate values.

Figure 1 illustrates the type of X–Y plot that can be produced by using the **normalize** function.

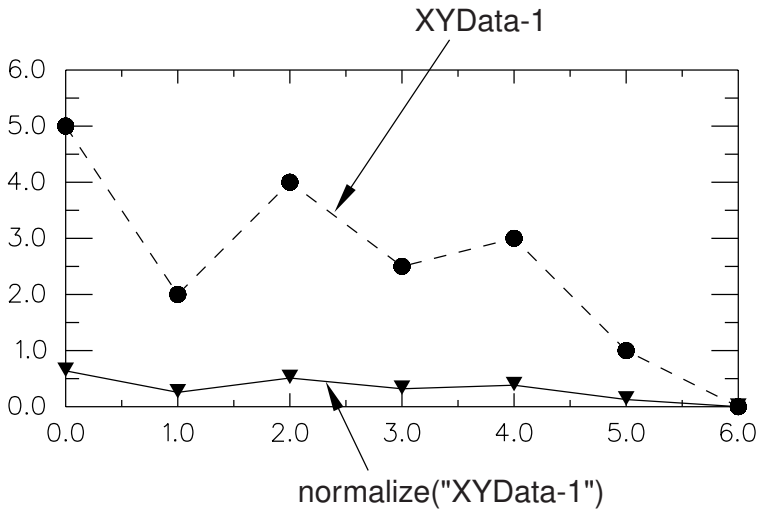


Figure 1: X–Y plot produced using the normalize function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **normalize(X)**.

   The **normalize** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **normalize** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Taking the square root of the sum of the squares of two or more X–Y data objects*
- *Using X–Y data operations*

Taking the square root of an X–Y data object

Use the **sqrt** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the square root of the original Y-coordinates.

Figure 1 provides an example of an X–Y plot produced using the **sqrt** function.

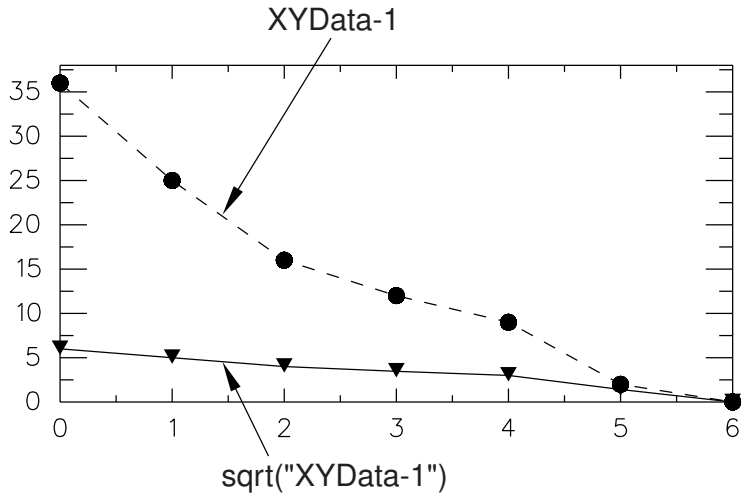


Figure 1: *X–Y* plot produced using the sqrt function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **sqrt(A)**.

   The **sqrt** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **sqrt** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Taking the square root of the sum of the squares of two or more X–Y data objects*
- *Using X–Y data operations*

Taking the square root of the sum of the squares of two or more X–Y data objects

Use the **srss** function to perform a square root of the sum of the squares operation on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new X–Y data object. Use this function, for example, to find the total magnitude of two or more X–Y data objects at matching X-coordinates.

The new X–Y data object has as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the remaining objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the square root of the sum of the squares of all input X–Y data object Y-coordinates at matching X-coordinates. The arguments to this function are commutative.

Figure 1 illustrates the type of X–Y plot that can be produced using the **srss** function.

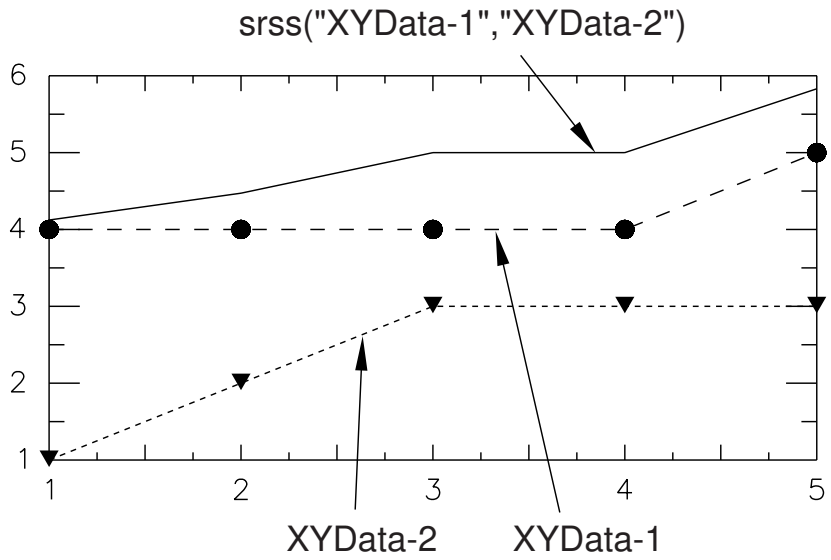


Figure 1: X–Y plot produced using the srss function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **srss(X,X,...)**.

   The **srss** function appears in the expression window.

3. From the **XY Data** choices, click the names of two or more X–Y data objects on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object names appear, separated by commas, within the **srss** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Taking the square root of an X–Y data object*
- *Normalizing an X–Y data object*
- *Using X–Y data operations*

Summing two or more X–Y data objects

Use the **sum** function to operate on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new X–Y data object. The **sum** function accepts two or more commutative arguments.



Note:

This function has the same behavior as the **+** symbol. For more information, see *Using addition on X–Y data objects*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.
2. From the **Operators** listed, click **sum(A,A,...)**.

   The **sum** function appears in the expression window.
3. From the **XY Data** choices, click the names of two or more X–Y data objects to sum together and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object names appear, separated by commas, within the **sum** function parentheses in the expression window.
4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.
7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using addition on X–Y data objects*
- *Appending two or more X–Y data objects*
- *Combining two X–Y data objects*
- *Using X–Y data operations*

Calculating vector magnitudes

Use the **vectorMagnitude** function to perform a square root of the sum of the squares operation on at most three *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new X–Y data object. This function is essentially identical to the **srss** function (see *Taking the square root of the sum of the squares of two or more X–Y data objects*) except that the operation can be performed on at most three data objects. You can use this function to plot the total magnitude of a vector by calculating the square root of the sums of the squares of the three vector components.

The new X–Y data object has as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the remaining objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the square root of the sum of the squares of all input X–Y data object Y-coordinates at matching X-coordinates. The arguments to this function are commutative.

Figure 1 illustrates the type of X–Y plot that can be produced using the **vectorMagnitude** function.

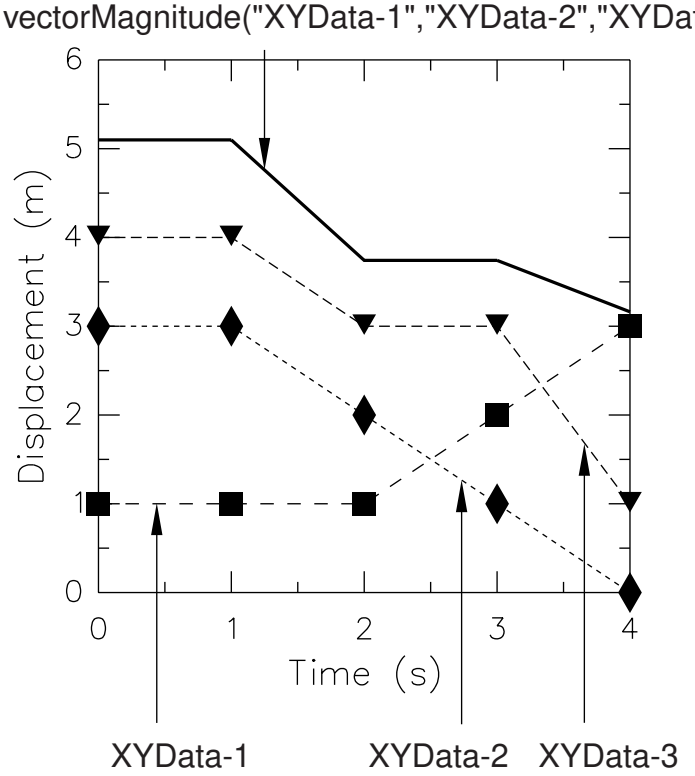


Figure 1: X–Y plot produced using the vectorMagnitude function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **vectorMagnitude(X,X,X)**.

   The **vectorMagnitude** function appears in the expression window.

3. From the **XY Data** choices, click the names of three X–Y data objects on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object names appear, separated by commas, within the **vectorMagnitude** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Taking the square root of an X–Y data object*
- *Normalizing an X–Y data object*
- *Using X–Y data operations*

Applying trigonometric functions to an X–Y data object

Use the trigonometric functions **cos**, **cosh**, **acos**, **sin**, **sinh**, **asin**, **tan**, **tanh**, and **atan** to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has the same *X*-coordinate values as the original *X–Y* data object, but new *Y*-coordinates are computed as the trigonometric function value of each original *Y*-coordinate. Arguments to trigonometric functions are assumed to be in radians.

For example, if

$$\mathbf{XYData} = [(0,0),(1,1.571),(2,3.14)],$$

then

$$\mathbf{sin(XYData)} = [(0,0),(1,1),(2,0.00159)].$$

Figure 1 shows an *X–Y* plot of the above example.

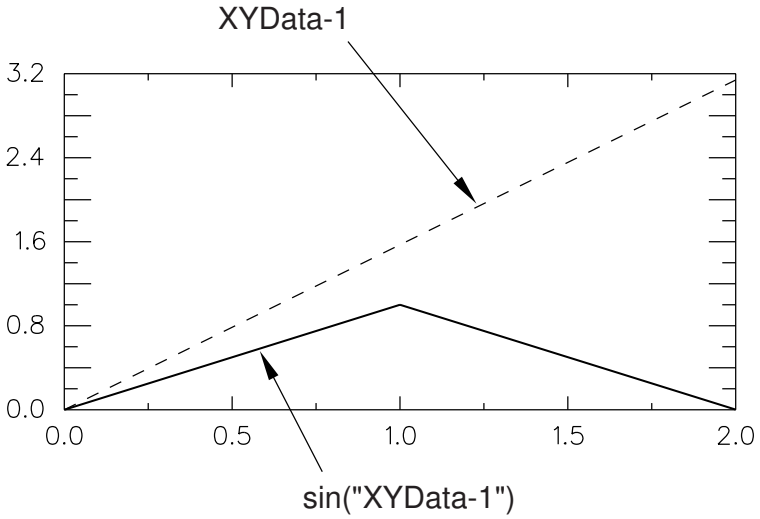


Figure 1: *X–Y* plot produced using the sin function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click one of the following trigonometric functions:

sin(A)	To compute the sine of your *X–Y* data object.
sinh(A)	To compute the hyperbolic sine of your *X–Y* data object.
asin(A)	To compute the arcsine of your *X–Y* data object. Your *Y*-data values must be in the range [–1, 1].
cos(A)	To compute the cosine of your *X–Y* data object.
cosh(A)	To compute the hyperbolic cosine of your *X–Y* data object.
acos(A)	To compute the arccosine of your *X–Y* data object. Your *Y*-data values must be in the range [–1, 1].
tan(A)	To compute the tangent of your *X–Y* data object.
tanh(A)	To compute the hyperbolic tangent of your *X–Y* data object.
atan(A)	To compute the arctangent of your *X–Y* data object.

The trigonometric function you choose appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the trigonometric function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Converting the angular units used for X–Y data objects*
- *Using X–Y data operations*

Taking the exponential of an X–Y data object

Use the **exp** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the exponential of the original Y-coordinates.

Figure 1 shows an example of an X–Y plot produced by using the **exp** function.

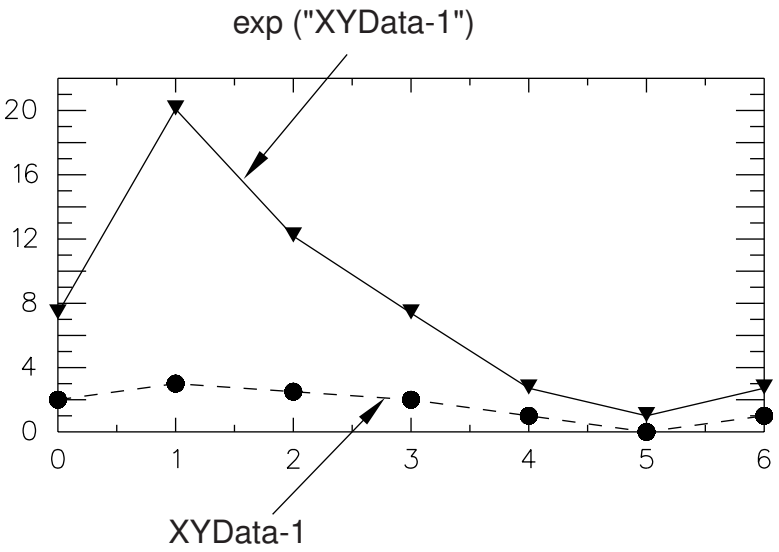


Figure 1: *X–Y* plot produced using the exp function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **exp(A)**.

   The **exp** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **exp** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Raising an X–Y data object to a power*
- *Using X–Y data operations*

Applying logarithmic functions to an X–Y data object

Use the logarithmic functions **log** and **log10** to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the logarithmic function value of each original Y-coordinate. For example, if

$$\mathbf{XYData} = [(1,1),(2,2.71),(3,7.39)],$$

then

$$\mathbf{log(XYData)} = [(1,0),(2,1.0),(3,2.0)].$$

Figure 1 illustrates the above example.

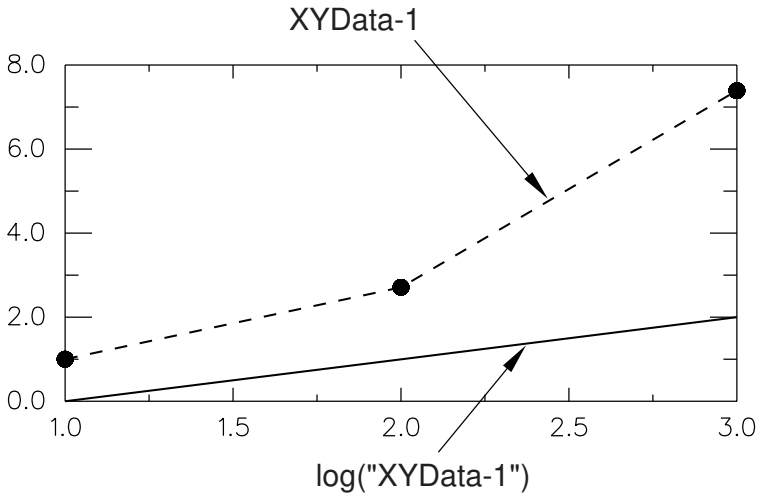


Figure 1: X–Y plot produced using the log function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click either **log(A)** or **log10(A)** to apply the natural log or base 10 log function, respectively.

   The logarithmic function you choose appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the logarithmic function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

• *Using X–Y data operations*

Raising an X–Y data object to a power

Use the **power** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The **power** function requires two arguments, at least one of which must be an X–Y data object. This function can be applied in three ways:

To raise a scalar to an X–Y data object: power(scalar, data object name).

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as the scalar raised to the power of each original Y-coordinate. For example, if

$$\mathbf{XYData} = [(1,1),(2,2),(3,3)],$$

then

$$\mathbf{power}(2, \mathbf{XYData}) = [(1,2),(2,4),(3,8)].$$

To raise an X–Y data object to a scalar: power(data object name, scalar).

This method yields a new X–Y data object having the same X-coordinate values as the original X–Y data object. Abaqus/CAE computes the new Y-coordinates as each original Y-coordinate raised to the scalar. For example, if

$$\mathbf{XYData} = [(2,-2),(3,3),(4,-4)],$$

then

$$\mathbf{power}(\mathbf{XYData}, 2) = [(2,4),(3,9),(4,16)].$$

To raise an X–Y data object to another X–Y data object: power(data object name, data object name).

This method yields a new X–Y data object having as its X-coordinates all X-coordinates of the first data object and any additional X-coordinates needed for alignment of the two objects. Abaqus/CAE computes additional X–Y data pairs by *interpolation and extrapolation*. The new data object has as its Y-coordinates the Y-coordinates of the first data object raised to the Y-coordinates of the second data object. For example, let

$$\mathbf{XYData1} = [(1,1),(2,2),(4,2),(5,1)],$$

and

$$\mathbf{XYData2} = [(0,1),(3,2),(5,6)].$$

For alignment, Abaqus/CAE computes the values of the first and second data objects as:

$$\mathbf{XYData1\ extrapolated} = [(0,1),(1,1),(2,2),(3,2),(4,2),(5,1)],$$

and

$$\mathbf{XYData2\ extrapolated} = [(0,1),(1,1.33),(2,1.66),(3,2),(4,4),(5,6)];$$

then

$$\mathbf{power}(\mathbf{XYData1}, \mathbf{XYData2}) = [(0,1),(1,1),(2,3.17),(3,4),(4,16),(5,1)].$$

Figure 1 shows a plot of the above example.

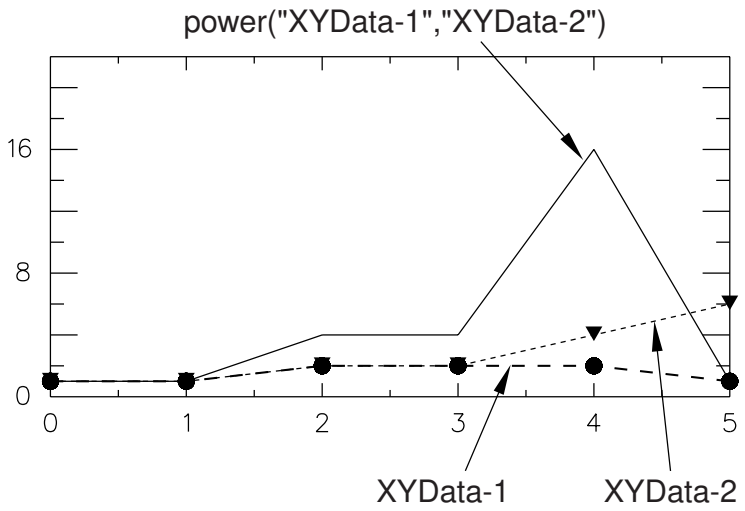


Figure 1: *X–Y* plot of one data object raised to another data object.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **power(A,A)**.

   The **power** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **power** function parentheses in the expression window.

4. To specify the next argument, do one of the following:

   - Use your mouse and keyboard to enter a scalar as either the first or second argument of the **power** function in the expression window, depending on the operation you want to perform, and to separate the two arguments with a comma, or
   - From the **XY Data** choices, click the name of a data object argument for the **power** function in the expression window and click **Add to Expression**.

5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Taking the exponential of an X–Y data object*
- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Applying Butterworth filtering to an X–Y data object

Use the **butterworthFilter** function to apply a Butterworth filtering operation to a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. This filtering operation can be used, for example, to remove high-frequency noise.

The transfer function of a Butterworth filter is shown in *Filtering Output and Operating on Output in Abaqus/Explicit*.

Figure 1 illustrates the type of *X–Y* plot that can be produced using the **butterworthFilter** operation.

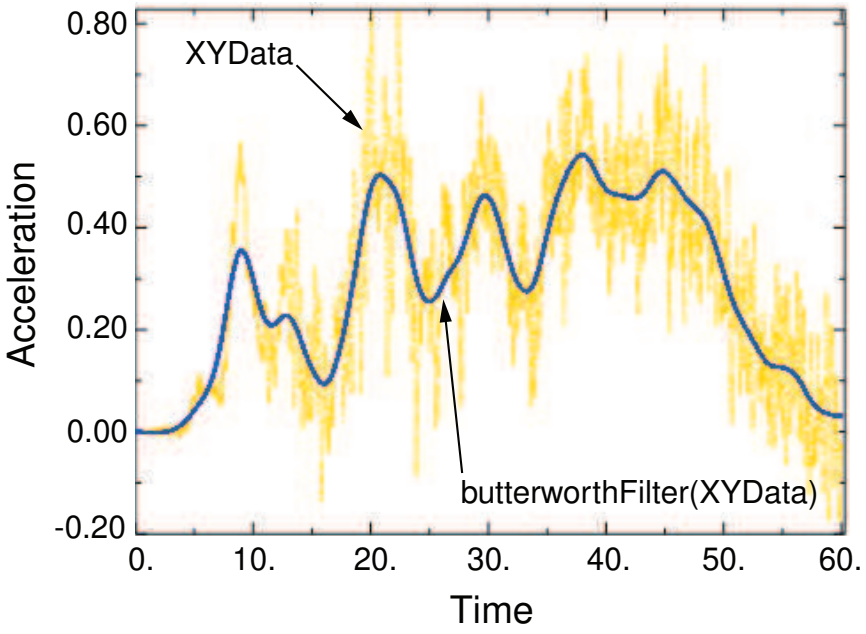


Figure 1: *X–Y* plot produced using the butterworthFilter operation.

The **butterworthFilter** function requires two arguments: the name of the *X–Y* data object (*name*) and the cutoff frequency (*cutoffFrequency*), which is the frequency above which the filter attenuates at least half of the input signal. A description of the optional arguments follows:

- The order of the filter you want to use (*filterOrder*). This argument must be a positive, even integer value; the default value is 2.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the start of the data signal (*startCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the first data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the first data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the first data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the first two data points. The default value is CONSTANT.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the end of the data signal (*endCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the last data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the last data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing

through the last data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the last two data points. The default value is CONSTANT.

- A symbolic constant that specifies the interpolation scheme (*interpolation*). Valid values for this argument are QUADRATIC, specifying a Lagrange second-order interpolation scheme; CUBIC_SPLINE, specifying a cubic spline interpolation scheme; and LINEAR, specifying a linear interpolation scheme. The default value is QUADRATIC.
- The slope of the raw data curve leading up to the first data point (*startslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- The slope of the raw data curve continuing past the final data point (*endslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- A Boolean specifying whether a backward pass (*backwardPass*) is to be performed on the filtered data. The default value for this argument is True. When this argument is set to False, the *endCondition* argument is ignored.

Your *X–Y* data object must have a constant time step for it to be filtered. If the time step is not constant, Abaqus/CAE computes additional points at constant intervals by *interpolation*. The constant time step for Butterworth filtering is defined by the smallest time step in the *X–Y* data object to be filtered.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **butterworthFilter(X,F)**.

   The **butterworthFilter** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **butterworthFilter** function parentheses in the expression window.

4. Position the cursor in the expression window before the second comma, and type in a value for the cutoff frequency.
5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*
- *Applying sine-Butterworth filtering to an X–Y data object*
- *Smoothing an X–Y data object*
- *Filtering Output and Operating on Output in Abaqus/Explicit*

Applying Chebyshev Type I or Type II filtering to an X–Y data object

Use the **chebyshev1Filter** or **chebyshev2Filter** function to apply a Chebyshev Type I or Type II filtering operation to a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. This filtering operation can be used, for example, to remove high-frequency noise.

The transfer functions of the Chebyshev filters are shown in *Filtering Output and Operating on Output in Abaqus/Explicit*.

Figure 1 illustrates the type of X–Y plot that can be produced using the **chebyshev1Filter** operation.

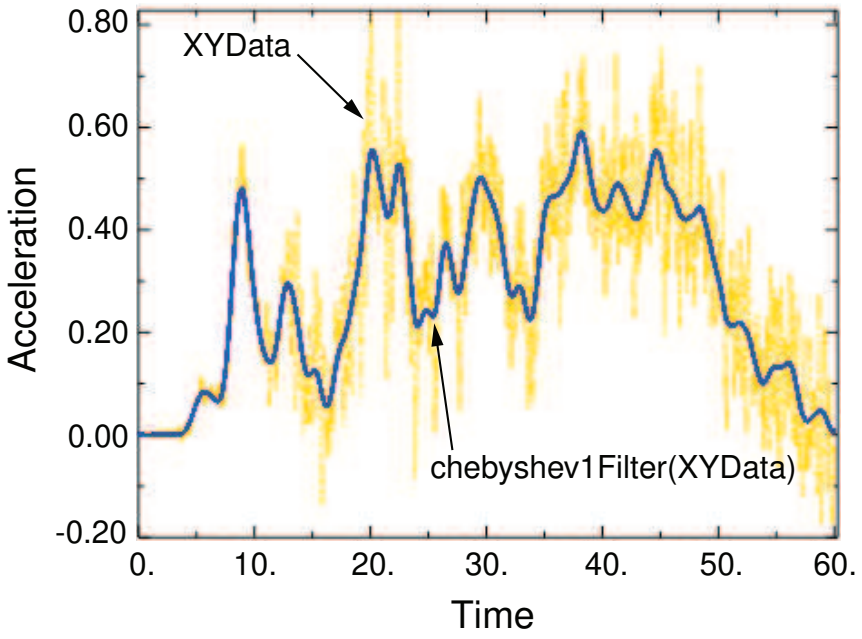


Figure 1: X–Y plot produced using the chebyshev1Filter operation.

The **chebyshev1Filter** and **chebyshev2Filter** functions use the same syntax and require the same set of arguments. The following arguments are required: the name of the X–Y data object (*name*); the cutoff frequency (*cutoffFrequency*), which is the frequency above which the filter attenuates at least half of the input signal; and the ripple factor (*rippleFactor*), which is a floating point number that indicates how much oscillation you will allow in exchange for an improved filter response. Both Chebyshev Type I and II filtering require a ripple factor greater than 0; in addition, Chebyshev Type II filtering requires the value to be less than 1. Chebyshev Type I filtering does not place any upper bound on the ripple factor value.

The two types of Chebyshev filters differ in where their response ripples occur and in their handling of the ripple factor value; see *Filtering Output and Operating on Output in Abaqus/Explicit* for a comparison of Type I and Type II Chebyshev output with typical Butterworth filter output.

A description of the optional arguments follows:

- The order of the filter you want to use (*filterOrder*). This argument must be a positive, even integer value; the default value is 2.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the start of the data signal (*startCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the first data point in the X–Y data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the X–Y data object about a vertical line passing through the first data point; REVERSE_MIRROR, which applies a projection

and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the first data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the first two data points. The default value is CONSTANT.

- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the end of the data signal (*endCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the last data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the last data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the last data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the last two data points. The default value is CONSTANT.
- A symbolic constant that specifies the interpolation scheme (*interpolation*). Valid values for this argument are QUADRATIC, specifying a Lagrange second-order interpolation scheme; CUBIC_SPLINE, specifying a cubic spline interpolation scheme; and LINEAR, specifying a linear interpolation scheme. The default value is QUADRATIC.
- The slope of the raw data curve leading up to the first data point (*startslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- The slope of the raw data curve continuing past the final data point (*endslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- A Boolean specifying whether a backward pass (*backwardPass*) is to be performed on the filtered data. The default value for this argument is True. When this argument is set to False, the *endCondition* argument is ignored.

Your *X–Y* data object must have a constant time step for it to be filtered. If the time step is not constant, Abaqus/CAE computes additional points at constant intervals by *interpolation*. The constant time step for Chebyshev Type I or II filtering is defined by the smallest time step in the *X–Y* data object to be filtered.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **chebyshev1Filter(X,F,F)** or **chebyshev2Filter(X,F,F)** for Chebyshev Type I or Type II filtering, respectively.

   The **chebyshev1Filter** or **chebyshev2Filter** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **chebyshev1Filter** or **chebyshev2Filter** function parentheses in the expression window.

4. Position the cursor in the expression window before the second comma, and type in a value for the cutoff frequency.
5. Position the cursor in the expression window before the third comma, and type in a positive value for the ripple factor. For the **chebyshev2Filter** function, this value must be less than 1.
6. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
7. To evaluate and display your expression, click **Plot Expression**.
8. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

9. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*
- *Applying sine-Butterworth filtering to an X–Y data object*
- *Smoothing an X–Y data object*
- *Filtering Output and Operating on Output in Abaqus/Explicit*

Reducing the sample size of an X–Y data object

Use the **decimateFilter** function to reduce a large, *previously saved*X–Y data object (a collection of ordered pairs) to a smaller, representative sample. The **decimateFilter** function is an anti-aliasing filter that selects points at a user-defined sampling frequency to produce a new *X–Y* data object.

The **decimateFilter** function requires two arguments: the name of the *X–Y* data object (*name*) and the decimation factor (*decimationFactor*), which is the factor by which the sampling rate is to be increased. The decimation factor must be an integer value greater than 1. The function also accepts an optional argument that specifies the ratio of the new sampling rate to the cutoff frequency used by the **decimateFilter** function prior to decimation (*cutoffFactor*). The cutoff factor must be an integer value of 2 or more. The default value is 3.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **decimateFilter(X,I)**.

   The **decimateFilter** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **decimateFilter** function parentheses in the expression window.

4. Position the cursor in the expression window before the second comma, and type in a value for the decimation factor.
5. If desired, position the cursor in the expression window before the third comma, and type in a value for the cutoff factor.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*
- *Smoothing an X–Y data object*

Applying SAE filtering to an X–Y data object

Use one of the SAE filtering functions to apply an SAE filtering operation to a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. For example, you can use these functions to remove high-frequency noise.

The SAE filtering operation performs two-pass, zero phase shift, second-order Butterworth filtering. The SAE filtering classes that are allowed in Abaqus/CAE are 60, 100, 180, 600, and 1000, as defined in SAE Standard J211 (OCT88). The Visualization module in Abaqus/CAE uses the Signal Analysis Software obtained from the National Highway Traffic Safety Administration (NHTSA). For more information, see the documentation for the Signal Analysis Software installation kit on the NHTSA website (search for NVS Software Applications on http://www.nhtsa.gov/).

Abaqus/CAE provides the following two types of SAE filtering functions:

- The general operator, **saeGeneralFilter**, uses a general filtering class and can filter data for a user-specified cutoff frequency.
- Class-specific operators filter for each of the five supported SAE filtering classes: **sae60Filter**, **sae100Filter**, **sae180Filter**, **sae600Filter**, and **sae1000Filter**. Because they use a specific filtering class, these operators do not support specification of a cutoff frequency.

Your X–Y data object must have a constant time step for it to be filtered. If the time step is not constant, Abaqus/CAE computes additional points at constant intervals by *interpolation*. The constant time step for SAE filtering is defined by the smallest time step in the X–Y data object to be filtered.

Figure 1 illustrates the type of X–Y plot that can be produced using the **saeGeneralFilter** operation.

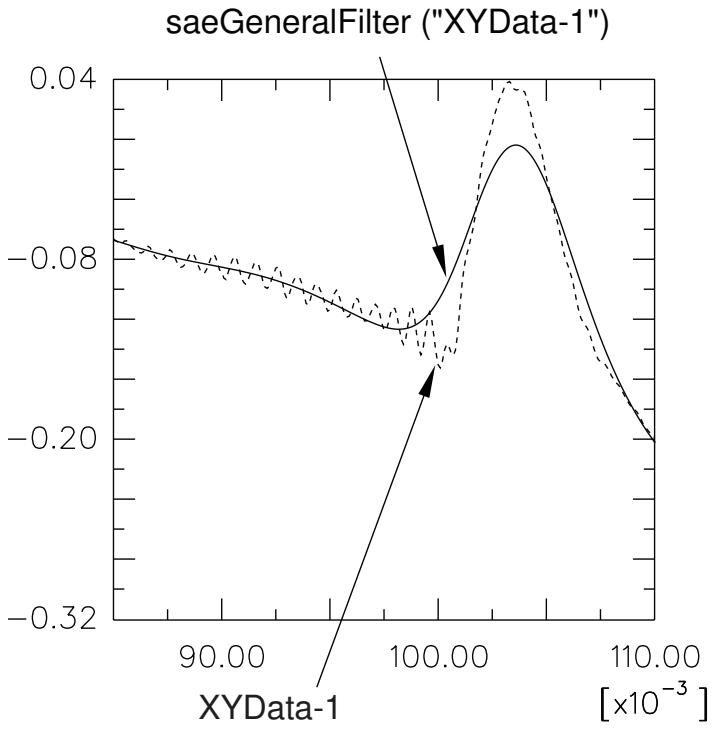


Figure 1: X–Y plot produced using the saeGeneralFilter operation.

The **saeGeneralFilter** function requires two arguments: the name of the X–Y data object (*name*) and the cutoff frequency (*cutoffFrequency*), which is the frequency above which the filter attenuates at least half of the input signal. The

class-specific filters also require two arguments: the name of the *X–Y* data object (*name*) and the time scale factor (*timeScaleFactor*), which is a multiplier for the *X*-values of the data set being filtered. For example, a value of 1 indicates a time scale of seconds, and a value of 0.001 indicates a time scale of milliseconds.

A description of the optional arguments follows:

- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the start of the data signal (*startCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the first data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the first data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the first data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the first two data points. The default value is CONSTANT.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the end of the data signal (*endCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the last data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the last data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the last data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the last two data points. The default value is CONSTANT.
- A symbolic constant that specifies the interpolation scheme (*interpolation*). Valid values for this argument are QUADRATIC, specifying a Lagrange second-order interpolation scheme; CUBIC_SPLINE, specifying a cubic spline interpolation scheme; and LINEAR, specifying a linear interpolation scheme. The default value is QUADRATIC.
- The slope of the raw data curve leading up to the first data point (*startslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- The slope of the raw data curve continuing past the final data point (*endslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- A Boolean specifying whether a backward pass (*backwardPass*) is to be performed on the filtered data. The default value for this argument is True. When this argument is set to False, the *endCondition* argument is ignored.

Abaqus/CAE refers to the values in the table below to determine the cutoff frequency multiplier (CM) for the class-specific SAE filter operators.

SAE class	Cutoff frequency multiplier (CM)
60	100
100	165
180	300
600	1000
1000	1650

Abaqus/CAE then calculates the cutoff frequency using the cutoff frequency multiplier and the user-specified time scale factor (TSF), as follows:

$$\text{cutoff frequency} = CM \times TSF$$

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **saeGeneralFilter(X,F)** for general SAE filtering or click one of the class-specific SAE filter functions.

   Your choice of filtering function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the function parentheses in the expression window.

4. If you selected **saeGeneralFilter(X,F)**, position the cursor in the expression window and enter a value for the cutoff frequency.
5. If you selected a class-specific SAE filter, position the cursor in the expression window and enter a value for the time scale factor.
6. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
7. To evaluate and display your expression, click **Plot Expression**.
8. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

9. When you are finished, click **Cancel** to close the dialog box.

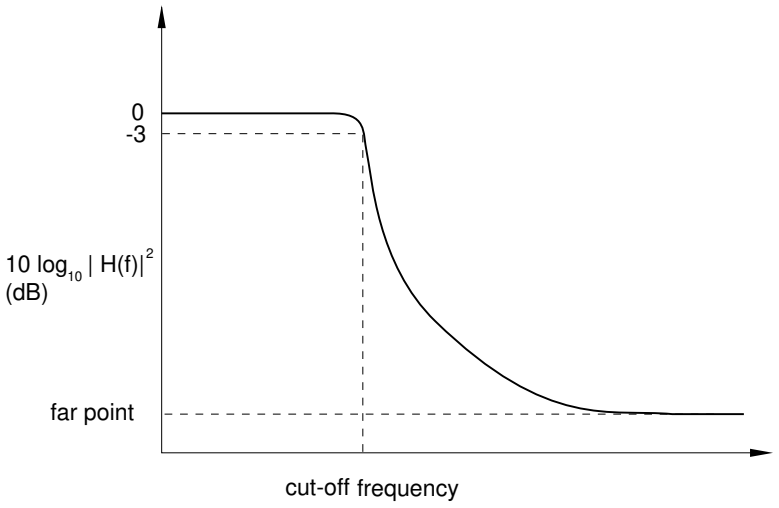
Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*
- *Applying sine-Butterworth filtering to an X–Y data object*
- *Smoothing an X–Y data object*

Applying sine-Butterworth filtering to an X–Y data object

Use the **sineButterworthFilter** function to apply a sine-Butterworth filtering operation to a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. This filtering operation can be used, for example, to remove high-frequency noise.

Abaqus/CAE uses a sine-Butterworth filter whose transfer function $|H(f)|^2$ has the following form:



For the filter to be applied to a curve, the curve must have data points at regularly spaced intervals in time. Therefore, a curve to be filtered is resampled at a given frequency (the sampling frequency). The default sampling frequency will be inadequate for curves containing data with a frequency content that is much higher than the cutoff frequency. Using a very high sampling frequency will create a curve with a very large number of data points.

The **sineButterworthFilter** requires two arguments: the name of the *X–Y* data object (*name*) and the cutoff frequency (*cutoffFrequency*), which is the frequency above which the filter attenuates at least half of the input signal. A description of the optional arguments follows:

- The order of the filter you want to use (*filterOrder*). This argument must be a positive, even integer value; the default value is 6.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the start of the data signal (*startCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the first data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the first data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the first data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the first two data points. The default value is CONSTANT.
- A symbolic constant specifying the method for computation of the projection and pre-charge to be applied at the end of the data signal (*endCondition*). Valid values for this argument are ZERO, which applies a constant projection and pre-charge of zero; CONSTANT, which applies a constant projection and pre-charge equal to the last data point in the *X–Y* data object; MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about a vertical line passing through the last data point; REVERSE_MIRROR, which applies a projection and pre-charge equivalent to reflecting the *X–Y* data object about both a vertical line and a horizontal line passing through the last data point; and TANGENTIAL, which applies a linear projection and pre-charge that is tangential to the last two data points. The default value is CONSTANT.

- A symbolic constant that specifies the interpolation scheme (*interpolation*). Valid values for this argument are QUADRATIC, specifying a Lagrange second-order interpolation scheme; CUBIC_SPLINE, specifying a cubic spline interpolation scheme; and LINEAR, specifying a linear interpolation scheme. The default value is QUADRATIC.
- The slope of the raw data curve leading up to the first data point (*startslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- The slope of the raw data curve continuing past the final data point (*endslope*). This argument's default value is 0.0 (for a level slope), and it is used only when interpolation=CUBIC_SPLINE.
- A Boolean specifying whether a backward pass (*backwardPass*) is to be performed on the filtered data. The default value for this argument is True. When this argument is set to False, the *endCondition* argument is ignored.

Your *X–Y* data object must have a constant time step for it to be filtered. If the time step is not constant, Abaqus/CAE computes additional points at constant intervals by *interpolation*. The constant time step is computed from the sampling frequency according to the following relationship:

$$\text{constant time step} = \frac{1}{2 \times \text{sampling frequency}}$$

Figure 1 illustrates the type of *X–Y* plot that can be produced using the **sineButterworthFilter** operation.

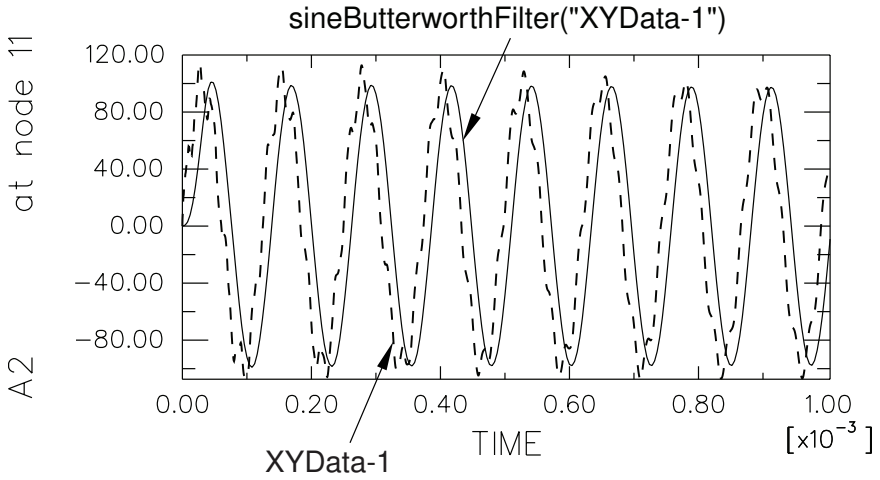


Figure 1: *X–Y* plot produced using the sineButterworthFilter operation.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **sineButterworthFilter(X,F)**.

   The **sineButterworthFilter** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **sineButterworthFilter** function parentheses in the expression window.

4. Position the cursor in the expression window before the second comma, and type in a value for the cutoff frequency.
5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.

6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*
- *Applying SAE filtering to an X–Y data object*
- *Smoothing an X–Y data object*

Smoothing an X–Y data object

Use the **smooth** function or the **smooth2** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object having a smoother curve. The new X–Y data object has the same X-coordinate values as the original X–Y data object. At each X-coordinate Abaqus/CAE computes the new Y-coordinate as the average of neighboring Y-coordinates of the original data object. This computation is known as a "moving average." The two functions use different algorithms to calculate this moving average value:

- The **smooth** function uses a simple moving average to calculate the new Y-coordinate values, which means that each new Y-coordinate is the unweighted mean of a number of neighboring points. You can specify the number of neighboring points to include in the average; a larger value produces a smoother curve. Abaqus/CAE interprets the value you specify as the number of points to use on either side of the current X-coordinate. The default is 2, which means Abaqus/CAE averages 5 values to compute each new Y-coordinate.
- The **smooth2** function uses an exponential moving average to calculate the new Y-coordinate values, which means that the neighboring points have an exponentially greater influence on the resulting Y-coordinate value. You can specify a smoothing factor between 0 and 1 for the curve; a smaller value produces a smoother curve. The default smoothing factor is 0.75.

Figure 1 illustrates the type of X–Y plot that can be produced using the **smooth** and **smooth2** functions.

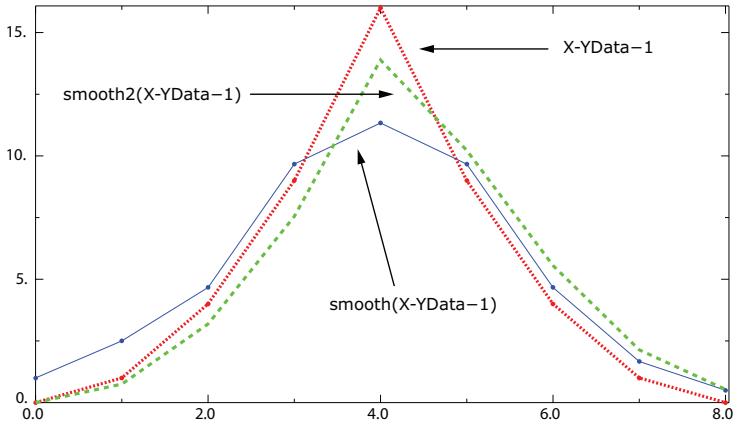


Figure 1: X–Y plots produced using the smooth and smooth2 functions.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **smooth(X, I)** or **smooth2(X, F)**.

   The **smooth** or **smooth2** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **smooth** or **smooth2** function parentheses in the expression window.

4. Position your cursor in the expression window before the second comma, and enter the following value:

- For the **smooth** function, enter an integer value for the number of points on either side of a given point to include in the average.
- For the **smooth2** function, enter a decimal value between 0 and 1 to determine the smoothing factor for the curve. A smaller value produces a smoother curve.

5. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
6. To evaluate and display your expression, click **Plot Expression**.
7. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
8. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Applying sine-Butterworth filtering to an X–Y data object*
- *Applying SAE filtering to an X–Y data object*
- *Using X–Y data operations*
- *Reducing the sample size of an X–Y data object*

Finding the current maximum of an X–Y data object

Use the **currentMax** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the maximum Y-coordinate at or before the current X-coordinate.

Figure 1 provides an example of an X–Y plot produced by using the **currentMax** function.

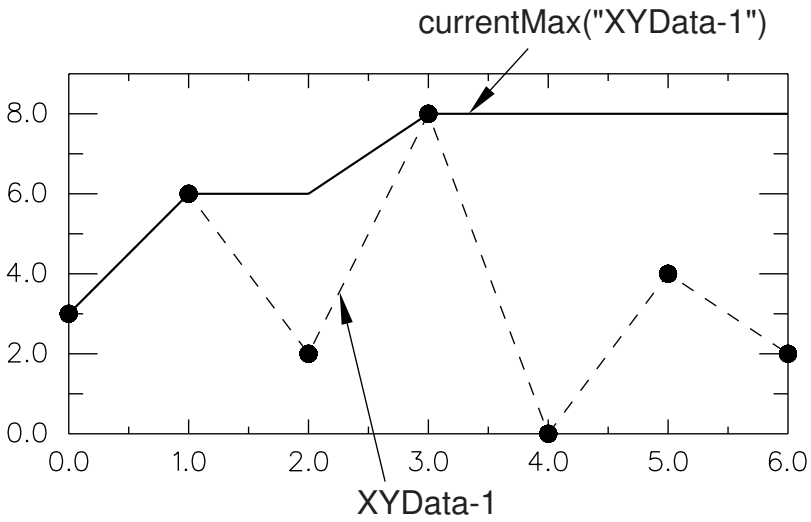


Figure 1: X–Y plot produced using the currentMax function.

Note:

The last value of the new X–Y data object represents the overall maximum Y-coordinate of the original X–Y data object.

For information on how to find the current maximum of two or more X–Y data objects, see *Finding the current maximum of two or more X–Y data objects*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **currentMax(X)**.

   The **currentMax** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **currentMax** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current maximum of two or more X–Y data objects*
- *Using X–Y data operations*

Finding the current minimum of an X–Y data object

Use the **currentMin** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the minimum Y-coordinate at or before the current X-coordinate.

Figure 1 shows an example of an X–Y plot produced by using the **currentMin** function.

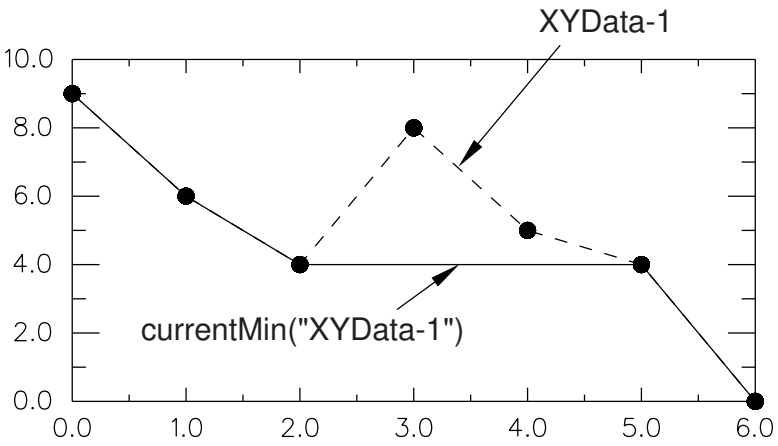


Figure 1: X–Y plot produced using the currentMin function.

Note:

The last value of the new X–Y data object represents the overall minimum Y-coordinate of the original X–Y data object.

For information on how to find the current minimum of two or more X–Y data objects, see *Finding the current minimum of two or more X–Y data objects*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **currentMin(X)**.

   The **currentMin** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **currentMin** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current minimum of two or more X–Y data objects*
- *Using X–Y data operations*

Finding the current range of an X–Y data object

Use the **currentRng** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object. The new X–Y data object has the same X-coordinate values as the original X–Y data object, but new Y-coordinates are computed as the difference between the minimum Y-coordinate and the maximum Y-coordinate at or before the current X-coordinate.

Figure 1 provides an example of an X–Y plot produced by using the **currentRng** function.

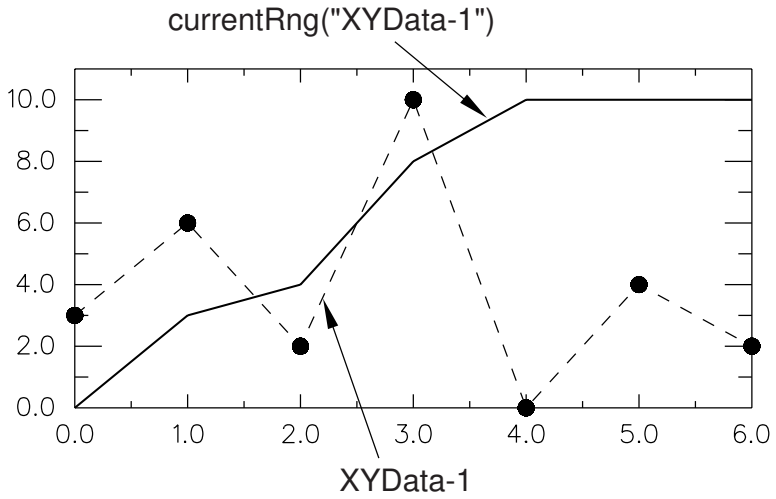


Figure 1: X–Y plot produced using the currentRng function.

Note:

The last value of the new X–Y data object represents the overall range of Y-coordinates of the original X–Y data object.

For information on how to find the current range of two or more X–Y data objects, see *Finding the current range of two or more X–Y data objects*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **currentRng(X)**.

   The **currentRng** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **currentRng** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current range of two or more X–Y data objects*
- *Using X–Y data operations*

Finding the current maximum of two or more X–Y data objects

Use the **maxEnvelope** function to operate on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has as its *X*-coordinates all *X*-coordinates of the input data objects and any additional *X*-coordinates needed for alignment of the input data objects. Abaqus/CAE computes additional *X–Y* data pairs by *interpolation and extrapolation*. The new data object has as its *Y*-coordinates the maximum *Y*-coordinate of all input *X–Y* data objects at the current *X*-coordinate. The arguments of this operation are commutative.

Figure 1 illustrates the type of *X–Y* plot that can be produced by using the **maxEnvelope** function.

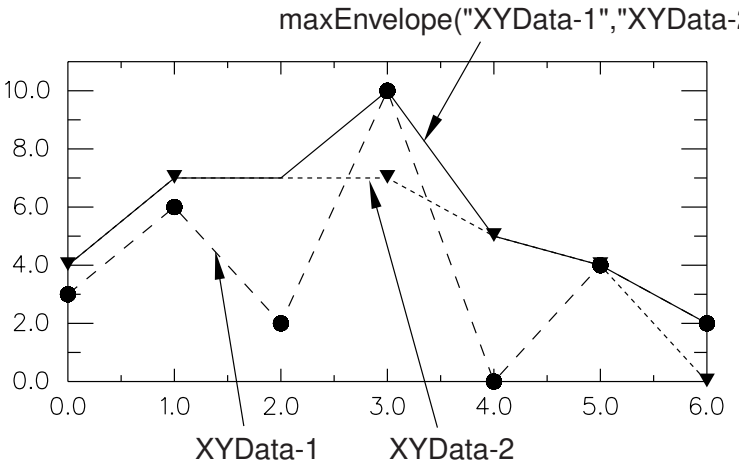


Figure 1: *X–Y* plot produced using the maxEnvelope function.

For information on how to find the current maximum of a single *X–Y* data object, see *Finding the current maximum of an X–Y data object*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **maxEnvelope(A,A,...)**.

   The **maxEnvelope** function appears in the expression window.

3. From the **XY Data** choices, click the names of two or more *X–Y* data objects on which to operate and click **Add to Expression**. Choices include all *X–Y* data objects previously saved within this session, listed alphabetically.

   The *X–Y* data object names appear, separated by commas, within the **maxEnvelope** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current maximum of an X–Y data object*
- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Finding the current minimum of two or more X–Y data objects

Use the **minEnvelope** function to operate on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has as its *X*-coordinates all *X*-coordinates of the input data objects and any additional *X*-coordinates needed for alignment of the input data objects. Abaqus/CAE computes additional *X–Y* data pairs by *interpolation and extrapolation*. The new data object has as its *Y*-coordinates the minimum *Y*-coordinate of all input *X–Y* data objects at the current *X*-coordinate. The arguments of this operation are commutative.

Figure 1 illustrates the type of *X–Y* plot that can be produced by using the **minEnvelope** function.

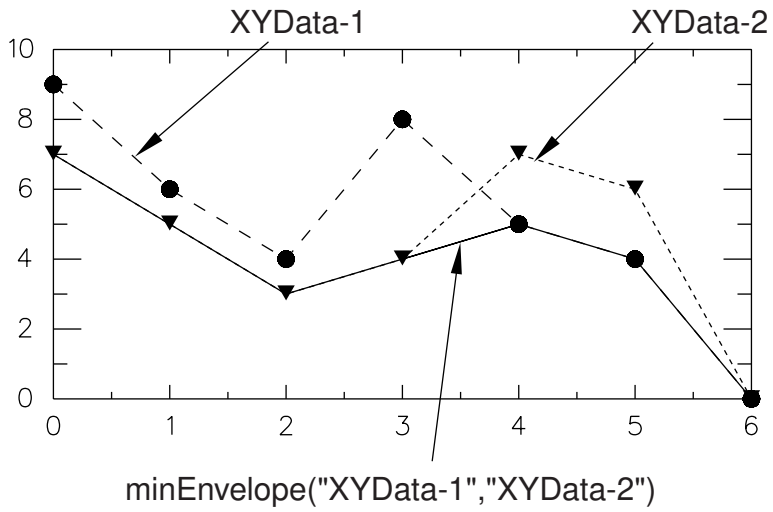


Figure 1: *X–Y* plot produced using the minEnvelope function.

For information on how to find the current minimum of a single *X–Y* data object, see *Finding the current minimum of an X–Y data object*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **minEnvelope(A,A,...)**.

   The **minEnvelope** function appears in the expression window.

3. From the **XY Data** choices, click the names of two or more *X–Y* data objects on which to operate and click **Add to Expression**. Choices include all previously saved *X–Y* data objects, which are listed alphabetically.

   The *X–Y* data object names appear, separated by commas, within the **minEnvelope** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current minimum of an X–Y data object*
- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Finding the current range of two or more X–Y data objects

Use the **rng** function to operate on two or more *previously saved*X–Y data objects (each object is a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has as its *X*-coordinates all *X*-coordinates of the input data objects and any additional *X*-coordinates needed for alignment of the input data objects. Abaqus/CAE computes additional *X–Y* data pairs by *interpolation and extrapolation*. Abaqus/CAE computes the new data object *Y*-coordinates as the difference between the minimum *Y*-coordinate and the maximum *Y*-coordinate at the current *X*-coordinate. The arguments of this operation are commutative.

Figure 1 illustrates the type of *X–Y* plot that can be produced by using the **rng** function.

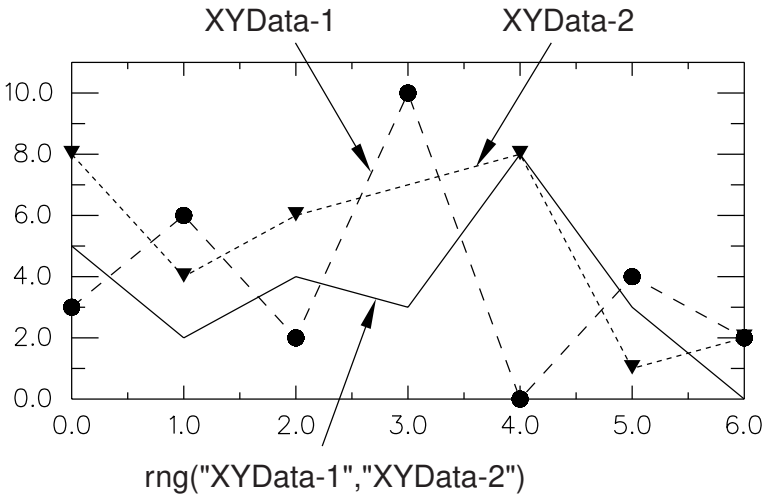


Figure 1: *X–Y* plot produced using the rng function.

For information on how to find the current range of a single *X–Y* data object, see *Finding the current range of an X–Y data object*.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **rng(A,A,...)**.

   The **rng** function appears in the expression window.

3. From the **XY Data** choices, click the names of two or more *X–Y* data objects on which to operate and click **Add to Expression**. Choices include all *X–Y* data objects previously saved within this session, listed alphabetically.

   The *X–Y* data object names appear, separated by commas, within the **rng** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Finding the current range of an X–Y data object*
- *Understanding X–Y data interpolation and extrapolation*
- *Using X–Y data operations*

Appending two or more X–Y data objects

Use the **append** function to link together two or more *previously saved*X–Y data objects (collections of ordered pairs) to produce a new X–Y data object. The new X–Y data object contains the data objects you specify concatenated in the order you specify.

Figure 1 shows an X–Y plot created by appending an X–Y data object with Y-values varying from 1 to −1 to an X–Y data object with Y-values varying from 0.5 to −0.5.

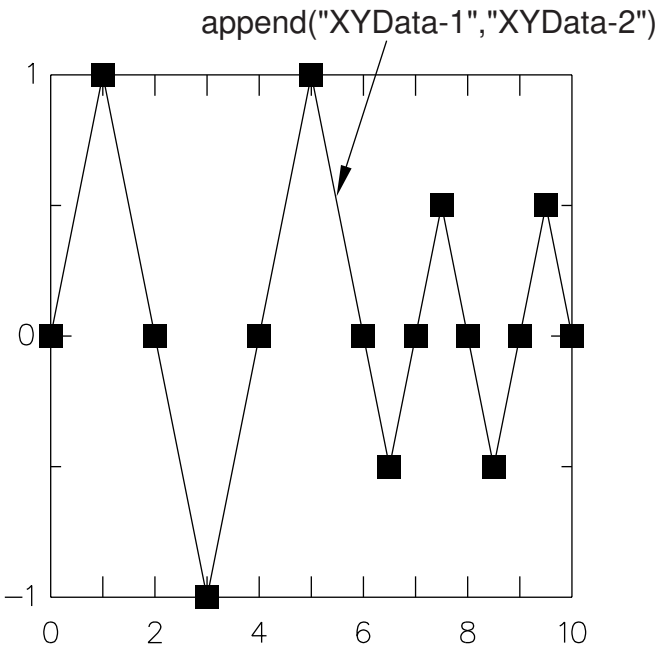


Figure 1: X–Y plot produced using the append function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **append(X,X,...)**.

   The **append** function appears in the expression window.

3. From the **XY Data** choices, click the names of the X–Y data objects to append in the order you want Abaqus to concatenate them and click **Add to Expression**. You can choose from X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object names appear, separated by commas, within the **append** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and X–Y data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new X–Y data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in X–Y plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Combining two X–Y data objects*
- *Using addition on X–Y data objects*
- *Summing two or more X–Y data objects*
- *Using X–Y data operations*

Combining two X–Y data objects

Use the **combine** function to combine two *previously saved*X–Y data objects (collections of ordered pairs) to produce a new X–Y data object. The new X–Y data object contains data pairs consisting of the Y-coordinate values of the first data object and the Y-coordinate values of the second data object, combined wherever X-coordinate values for the two data objects match. For example,

$$\mathbf{XYData1} = [(1,4),(2,4),(3,4),(4,5),(5,6)]$$

and

$$\mathbf{XYData2} = [(1,9),(2,8),(3,7),(4,6),(5,4)];$$

then

$$\mathbf{combine(XYData1, XYData2)} = [(4,9),(4,8),(4,7),(5,6),(6,4)].$$

The arguments of this operation are not commutative. If X-coordinate values for the two data objects do not match, Abaqus/CAE computes the missing data points to allow the X–Y data objects to be aligned. For more information on how missing data points are computed, see *Understanding X–Y data interpolation and extrapolation*.

A typical application of this function would be to combine a data object having X-values of time and Y-values of load with a data object having X-values of time and Y-values of displacement to produce a data object having X-values of displacement and Y-values of load. Another example would be to combine a data object having X-values of time and Y-values of stress with a data object having X-values of time and Y-values of strain to produce a data object having X-values of strain and Y-values of stress, as shown in *Figure 1*.

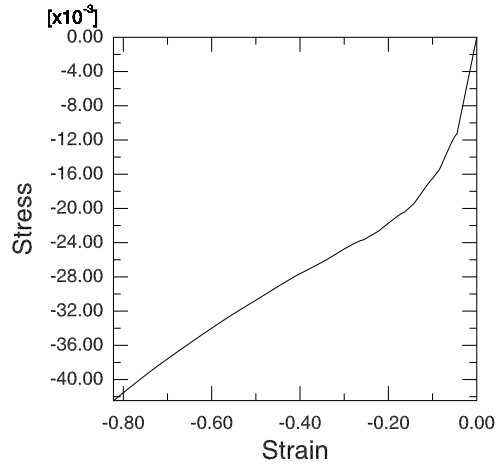


Figure 1: X–Y plot of stress versus strain.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **combine(X,X)**.

   The **combine** function appears in the expression window.

3. From the **XY Data** choices, click the names of the two X–Y data objects to combine in the order you want Abaqus to combine them and click **Add to Expression**. Choices include all X–Y data objects previously saved within this session, listed alphabetically.

The *X–Y* data object names appear, separated by commas, within the **combine** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding X–Y data interpolation and extrapolation*
- *Appending two or more X–Y data objects*
- *Using addition on X–Y data objects*
- *Summing two or more X–Y data objects*
- *Using X–Y data operations*

Converting the angular units used for X–Y data objects

Use the **degreeToRadian** or **radianToDegree** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has the same *X*-coordinate values as the original *X–Y* data object. Abaqus/CAE computes the new *Y*-coordinate values by converting each original *Y*-coordinate from degrees to radians or from radians to degrees, depending on the function you specify.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **degreeToRadian** to convert from degrees to radians or **radianToDegree** to convert from radians to degrees.

   The respective function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*

Swapping the order of an X–Y data object

Use the **swap** function to exchange the *X*- and *Y*-values of a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new *X–Y* data object. The new *X–Y* data object has as its *X*-coordinate values the *Y*-coordinate values of the original *X–Y* data object and has as its *Y*-coordinate values the *X*-coordinate values of the original data object.

Figure 1 illustrates the type of *X–Y* plot that can be produced by using the **swap** function.

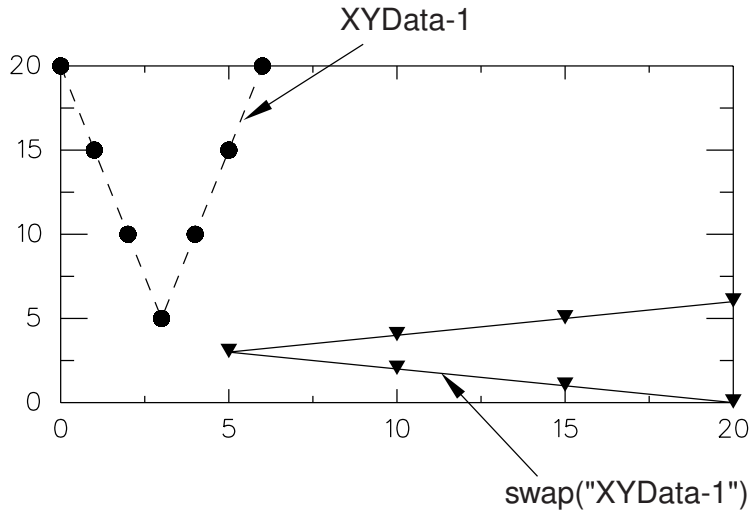


Figure 1: *X–Y* plot produced using the swap function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **swap(X)**.

   The **swap** function appears in the expression window.

3. From the **XY Data** choices, click the name of the *X–Y* data object on which to operate and click **Add to Expression**. You can choose from all *X–Y* data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The *X–Y* data object name appears within the **swap** function parentheses in the expression window.

4. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
5. To evaluate and display your expression, click **Plot Expression**.
6. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.

7. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*

Truncating data from the ends of an X–Y data object

Use the **truncate** function to operate on a *previously saved*X–Y data object (a collection of ordered pairs) to produce a new X–Y data object that excludes data points whose X–coordinate values are less than a user-specified minimum value or data points whose X–coordinate values are greater than a user-specified maximum value.

Figure 1 illustrates an X–Y plot that has been truncated at a maximum of 7 by using the **truncate** function.

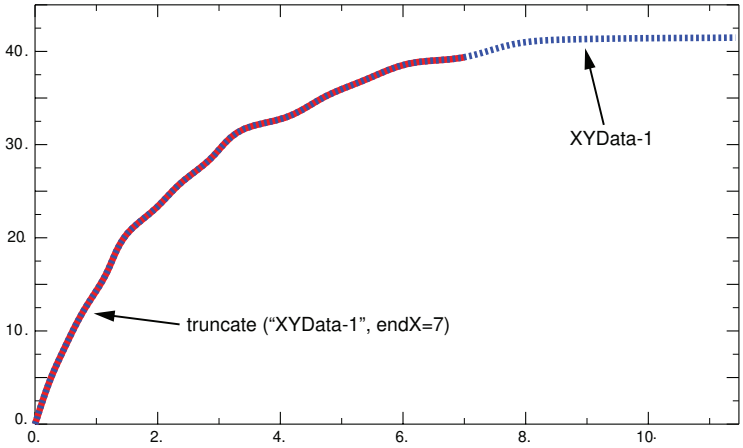


Figure 1: X–Y plot produced using the truncate function.

1. Locate the **Operate on XY Data** dialog box.

   From the main menu bar, select **Tools->XY Data->Create**. Click **Operate on XY data** in the dialog box that appears; then click **Continue**. The **Operate on XY Data** dialog box appears.

2. From the **Operators** listed, click **truncate(X,F)**.

   The **truncate** function appears in the expression window.

3. From the **XY Data** choices, click the name of the X–Y data object on which to operate and click **Add to Expression**. You can choose from all X–Y data objects previously saved within this session (listed alphabetically in the **XY Data** field).

   The X–Y data object name appears within the **truncate** function parentheses in the expression window.

4. Specify the X–coordinate values above which or below which data points should be removed from the X–Y data object. To specify a maximum value, position your cursor in the expression window before the second comma, and enter a number for the maximum value. To specify a minimum value, append `startX=`*value* to the expression in the expression window.

5. If desired, customize the truncation order value, which determines whether Abaqus/CAE should create new data points at the truncation boundary or boundaries you specify and, if so, how these new data points should be calculated. Select one of the following:

   - If `truncOrder=2` (the default value), Abaqus/CAE performs a piecewise, linear fit on the data to determine a new data point at each boundary and includes the new point or points in the new X–Y data object.
   - If `truncOrder=3`, Abaqus/CAE performs a spline interpolation on the data to determine a new data point at each boundary and includes the new point or points in the new X–Y data object.
   - If `truncOrder=1`, Abaqus/CAE does not create any new points at the boundary values.

6. If desired, customize the truncation tolerance value. If you elect to include interpolated values at the truncation boundaries, the truncation tolerance value overrides the creation of these new data points

if Abaqus/CAE finds existing data points that are already very close to the boundaries. By default, the truncation tolerance (`truncTol`) is 0.005, or one half of one percent of the average distance in the *X*-direction between adjacent data points in the *X–Y* data object. If Abaqus/CAE finds an existing data point closer to the truncation boundary than the truncation tolerance value, it does not create a new interpolated data point at that boundary.

7. To continue to build your expression, position the cursor in the expression window and type in or select the functions, operators, and *X–Y* data you want to include.
8. To evaluate and display your expression, click **Plot Expression**.
9. To *save* your new *X–Y* data object, click **Save As** and then provide a name in the dialog box that appears.

   Saving your data object makes it available for future operations within this session and for inclusion in *X–Y* plots containing multiple data objects.
10. When you are finished, click **Cancel** to close the dialog box.

Additional information

- *Using X–Y data operations*

Customizing X–Y plot axes

This section explains how you use the **Axis Options** dialog box to customize the appearance of the plot axes of an *X–Y* plot.

In this section:

- *Using X–Y plot axis options*
- *Plotting multiple X–Y data objects with different variables*
- *Customizing the X–Y plot axis scale*
- *Customizing the X–Y plot axis range*
- *Customizing the X–Y plot axis tick mode*
- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis titles*
- *Customizing the placement of X–Y plot axes*
- *Customizing the X–Y plot axis labels*
- *Customizing the X–Y plot axis color and style*

Using X–Y plot axis options

You can use the axis options to customize the appearance of the axes in an *X–Y* plot. *Figure 1* identifies the *X–Y* plot characteristics that you can customize.

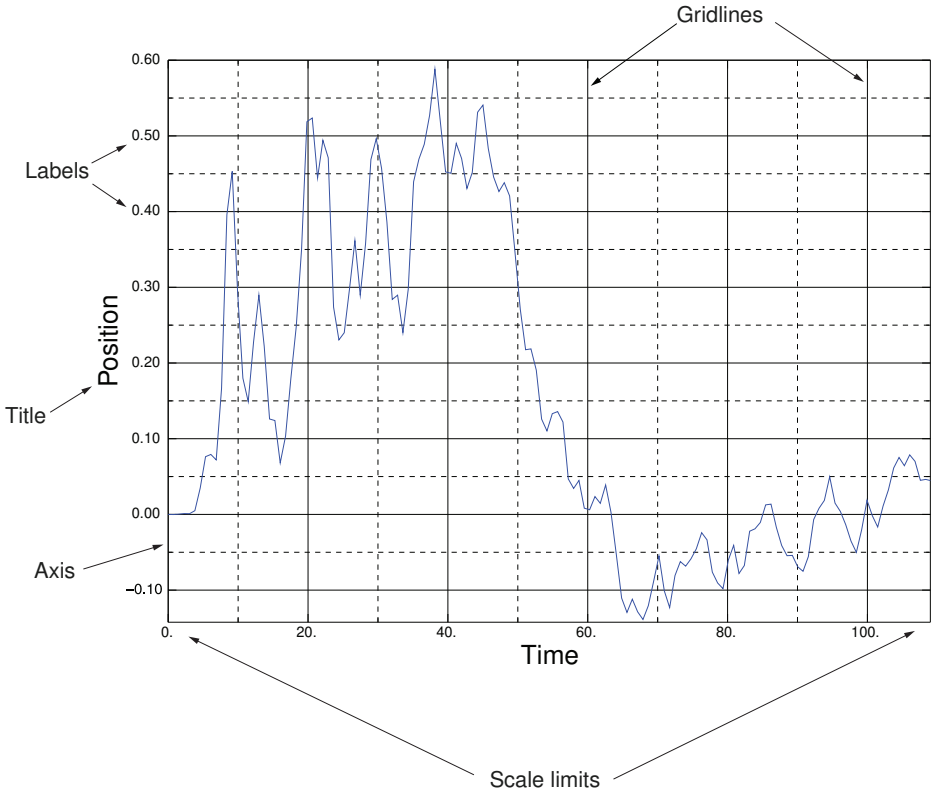


Figure 1: Characteristics of an X–Y plot.

Abaqus/CAE provides access to these options from the **Axis Options** dialog box, which you can open from the following locations in the Visualization module:

- From the main menu bar, select **Options->XY Options->Axis**.
- From the Visualization module toolbox, click .
- From the Results Tree, expand the **X Axes** or **Y Axes** container, highlight the axis or axes you want to modify, click mouse button 3, and select **Axis Options** from the list that appears.
- With an *X–Y* plot displayed in the viewport, double-click one of the axes in the plot.

  Items in an *X–Y* plot are selectable only when there is no active procedure in effect; you might have to cancel the current procedure to activate the components in an *X–Y* plot.

You can click the following tabs to customize the appearance of the selected axes:

- **Scale**: Choose axes scales and limits and customize the occurrence of tick marks.
- **Tick Marks**: Control the appearance and placement of tick marks.
- **Title**: Specify the content and appearance of axes titles.

- **Axes**: Choose axes placement, color, style, and thickness, and control numeric axes labels.

Additional information

- *Plotting multiple X–Y data objects with different variables*
- *Customizing the X–Y plot axis scale*
- *Customizing the X–Y plot axis range*
- *Customizing the X–Y plot axis tick mode*
- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis titles*
- *Customizing the placement of X–Y plot axes*
- *Customizing the X–Y plot axis labels*
- *Customizing the X–Y plot axis color and style*

Plotting multiple X–Y data objects with different variables

When you plot *X–Y* data objects with different variables on the same *X–Y* plot, Abaqus/CAE displays an axis for each quantity type used in these data objects. In the *X–Y* plot shown in *Figure 1*, three *X–Y* data objects are plotted: the **ALLKE** and **ALLSE** respectively show kinetic energy and strain energy over time, while **Displacement** shows displacement of a selected node over time. The resultant plot includes only a single axis, **Time**, in the *X*-direction because all three *X–Y* curves share this quantity type along this axis. In the *Y*-direction, where these *X–Y* data objects use different quantity types, the *X–Y* plot displays separate **Energy** and **Displacement** axes.

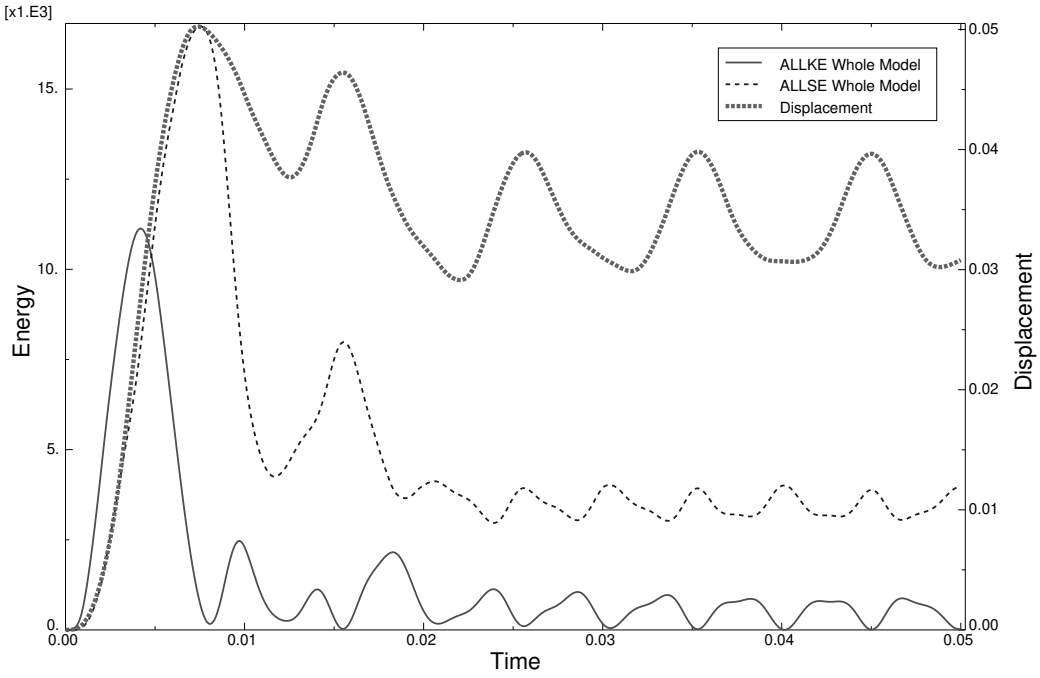


Figure 1: X–Y plot displaying two axes in the *Y*-direction.

When you plot multiple curves on the same *X–Y* plot, Abaqus/CAE sets the minimum and maximum values for the plot along both axes to the minimum and maximum values of the first *X–Y* data object you included in your plot. This *X–Y* data object is listed first in the *X–Y* plot legend and in the **Curves** area of the **Curve Options** dialog box.

When an *X–Y* plot displays multiple axes in the *X*- or *Y*-direction, you can align them on the same side of the *X–Y* plot, on opposite sides, or place one or both in the middle of the *X–Y* plot. You can also manipulate the scale or appearance of these *X–Y* plot axes independently to make their differences more apparent in the viewport.

Additional information

- *Customizing the X–Y plot axis scale*
- *Customizing the X–Y plot axis range*
- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis titles*
- *Customizing the X–Y plot axis labels*
- *Customizing the X–Y plot axis color and style*

Customizing the X–Y plot axis scale

Click the **Scale** tab to choose either a linear, logarithmic, or decibel-based scale for the *X*- and *Y*-axes of an *X–Y* plot. The scales of all *X*- and *Y*-axes that you display are independent. By default, Abaqus/CAE uses a linear scale for all axes.

Decibel-based scales

You can rescale data along a particular axis using either of two decibel-based scales. Each scale is suited to a particular type of wave data: the **10 dB** scale is recommended for displaying data about wave power or intensity; the **20 dB** scale is recommended for data about wave amplitude.

When you select a decibel-based axis scale, you can also adjust the decibel reference value. Adjusting this value enables you to rescale your data object according to the environmental conditions for the data in this *X–Y* data object and for the units in your model. The default value is `1`. You may want to adjust this value to display data at different environments or for waves traveling through a different medium, such as water.

1. Locate the **Scale** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Scale** tab in the dialog box that appears.
2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Select one of the following axis scale options:

Linear

The selected axis is displayed in a linear progression.

Log

The selected axis is displayed in a base 10 logarithmic progression.

10 dB

The selected axis is displayed in a 10 dB progression. The **dB Reference** field becomes available, and you can enter the decibel reference value for the 10 dB–based data along this axis.

20 dB

The selected axis is displayed in a 20 dB progression. The **dB Reference** field becomes available, and you can enter the decibel reference value for the 20 dB–based data along this axis.

4. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Using X–Y plot axis options*

Customizing the X–Y plot axis range

Click the **Scale** tab to customize the range of the *X*- and *Y*-axes of an *X*–*Y* plot. By default, Abaqus/CAE computes the minimum and maximum axis values based on the minimum and maximum data points included in the plot.

1. Locate the axis **Max** and **Min** options.

   Select **Options**->**XY Options**->**Axis**; or click , which is located with the *X*–*Y* plotting tools in the Visualization module toolbox. Click the **Scale** tab in the dialog box that appears.

   The **Max** and **Min** options appear in the middle of the **Scale** page.
2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Enter a value in the **Max** field, or toggle on **Auto-compute** next to this field to set the maximum value for the selected axis or axes to the highest available data point.

   Abaqus/CAE updates the selected axis or axes.
4. Enter a value in the **Min** field, or toggle on **Auto-compute** next to this field to set the minimum value for the selected axis or axes to the lowest available data point.

   Abaqus/CAE updates the selected axis or axes.
5. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Using X–Y plot axis options*

Customizing the X–Y plot axis tick mode

Click the **Scale** tab to customize the tick mode of the *X*- and *Y*-axes of an *X–Y* plot. The tick mode options enable you to control the number of major and minor tick marks that appear along either axis. By default, Abaqus/CAE computes the minimum and maximum axis values based on the minimum and maximum data points included in the plot.

1. Locate the **Tick Mode** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Scale** tab in the dialog box that appears.

   The **Tick Mode** options appear at the bottom of the **Scale** page.

2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Choose the method of computing the interval between major tick marks along the selected axis.

   From the **Major** options, choose one of the following:

   ### Automatic

   Abaqus computes the interval between major tick marks along the selected axis, rounding to provide reasonable intervals.

   ### By increment

   The **Tick increment** text field becomes available, and you can enter the increment between major tick marks along the selected axis.

   ### By count

   The **Number of ticks per decade** field becomes available, and you can increase or decrease the total number of major tick marks to appear between the minimum and maximum values along the selected axis.

4. Specify the number of minor tick marks.

   From the **Minor** ticks option, click the **Ticks per increment** arrows (for linear or decibel-based axes) or the **Ticks per decade** arrows (for logarithmic axes) to specify the number of minor tick marks between each major tick mark.

5. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Using X–Y plot axis options*

Customizing X–Y plot axis tick marks

You can customize X–Y plot axis tick marks.

Click the **Tick Marks** tab to control the length, thickness, style, and color of major and minor tick marks that appear along the *X*- and *Y*-axes of an *X*–*Y* plot. The number of major and minor tick marks is controlled separately; see *Customizing the X–Y plot axis tick mode*. You can control the *X*- and *Y*-axis tick marks independently. Examples of major and minor tick marks are shown in *Figure 1*.

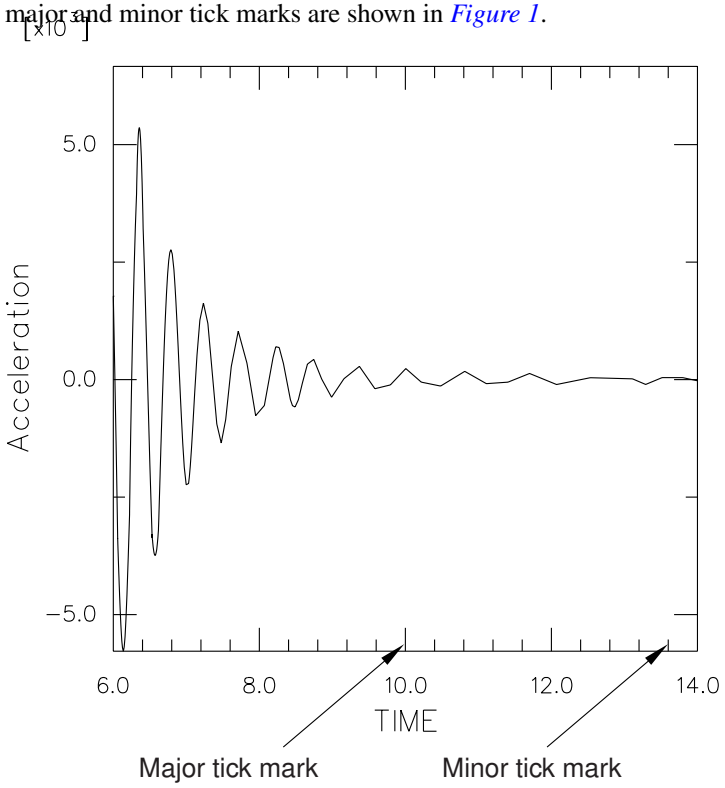


Figure 1: X–Y plot with major and minor tick marks indicated.

Major tick marks govern the placement of the *X*–*Y* plot major *grid* and the *numeric axis labels*. Minor tick marks are shorter, unlabeled marks governing the placement of the minor gridlines.

By default, Abaqus/CAE automatically computes the number of tick marks. You can specify major tick marks by giving the increment between them or the total number of tick marks. You can specify minor tick marks only by giving the total number between each major tick mark.

The color of the tick marks is governed by the color of the axes. For more information, see *Customizing the X–Y plot axis color and style*.

1. Locate the axis **Tick Marks** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X*–*Y* plotting tools in the Visualization module toolbox. Click the **Tick Marks** tab in the dialog box that appears.

2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Choose the location in which to place the axis tick marks.

   From the **Placement** options, choose one of the following:

None

Abaqus hides the tick marks completely for the selected axis.

Inside

Abaqus displays the tick marks on the same side of the axis as the *X–Y* curves.

Outside

Abaqus displays the tick marks on the side of the axis opposite the *X–Y* curves.

Across

Abaqus displays tick marks that straddle both sides of the selected axis.

4. Specify the length of the tick marks.

   Click the **Length** arrows to increase or decrease the length of the tick marks. Abaqus/CAE increases the length of the major and minor tick marks.

5. Choose the line style.

   From the **Style** list, click the desired line style (solid, dashed, etc.).

   The specified line style appears for the tick marks along the selected axis and in the **Style** list.

6. Choose the line thickness.

   From the **Thickness** list, click the desired line thickness.

   The specified line thickness appears for the tick marks along the selected axis and in the **Thickness** list.

7. Choose the line color:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The tick marks for the selected axis change to the selected color.

8. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Customizing the X–Y plot axis labels*
- *Using X–Y plot axis options*

Customizing the X–Y plot axis titles

Click the **Title** tab to customize the titles that appear along the *X*- and *Y*-axes of an *X–Y* plot. You can customize the following:

- The text that appears as each title.
- The color of each title.
- The font of each title.

1. Locate the axis **Title** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Title** tab in the dialog box that appears.
2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. To specify title text other than the default for the selected axis, enter the title in the field. You can revert back to the default title by toggling on **Use default**.
4. Select the font of the axis title.

   a. Click .

   Abaqus/CAE displays the **Select Font** dialog box.

   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.

   c. Click **Apply** to see the effect of your font selections.

   d. Click **OK** to close the **Select Font** dialog box.

   The X–Y plot title changes to the selected font, size, and style.
5. Select the color of the axis title.

   a. Click the color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

   The axis title for the selected axis changes to the selected color.
6. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Customizing the X–Y plot axis labels*
- *Using X–Y plot axis options*

Customizing the placement of X–Y plot axes

Click the **Axes** tab to customize the placement of the *X*- and *Y*-axes of an *X*–*Y* plot. This section describes the **Placement** options, which control the location of the axes themselves; to customize the location of the axis labels, see *Customizing the X–Y plot axis labels*

1. Locate the **Placement** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X*–*Y* plotting tools in the Visualization module toolbox. Click the **Axes** tab in the dialog box that appears. The **Placement** options for the axes appear at the top of the **Axes** page.
2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. From the **Placement** options at the top of the **Axes** page, select one of the following:

Min Edge

Abaqus/CAE places an axis on the minimum edge of the *X*–*Y* plot only. For *X*-axes, the minimum edge is the bottom of the viewport; for *Y*-axes, the minimum edge is the left side of the viewport.

Max Edge

Abaqus/CAE places an axis on the maximum edge of the *X*–*Y* plot only. For *X*-axes, the maximum edge is the top of the viewport; for *Y*-axes, the maximum edge is the right side of the viewport.

Min/Max Edge

Abaqus/CAE places the selected axis on both the maximum and minimum edges of the *X*–*Y* plot. For *X*-axes, Abaqus/CAE places the selected axis on the top and bottom of the viewport; for *Y*-axes, Abaqus/CAE the selected axis on the left and right sides of the viewport.

Center

Abaqus/CAE places the axis in the center of the *X*–*Y* plot.

4. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis titles*
- *Using X–Y plot axis options*

Customizing the X–Y plot axis labels

You can customize the X–Y plot axis labels.

Click the **Axes** tab to customize the numeric labels that appear with the major *tick marks* along the *X*- and *Y*-axes of an *X–Y* plot. Examples of axis labels appear in *Figure 1*.

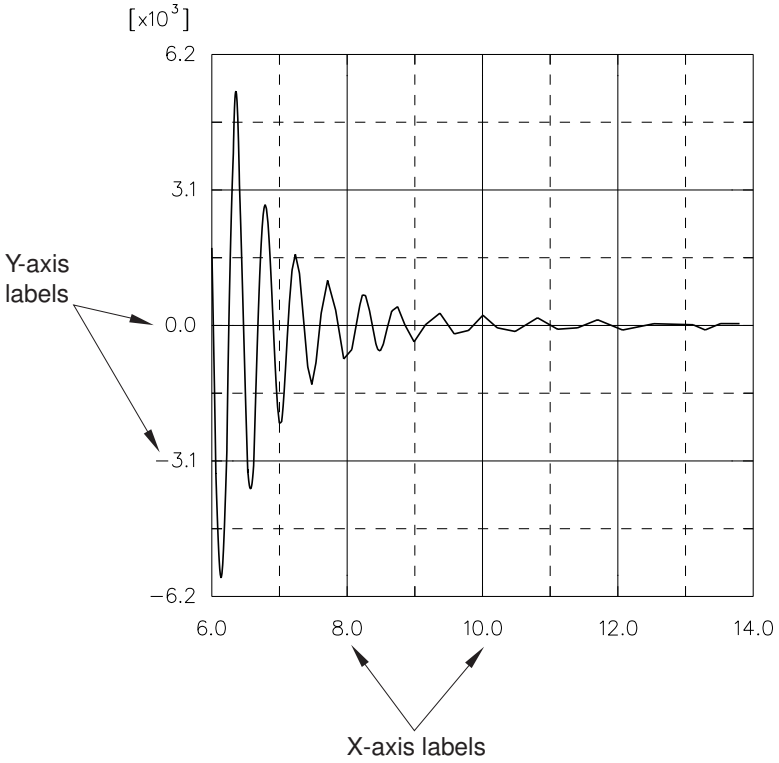


Figure 1: *X–Y* plot with axis labels indicated.

You can customize the following:

- The format of the labels (automatic, decimal, engineering, or scientific).
- The number of decimal places or significant digits in the labels.
- The frequency of the labels in relation to the major tick marks.
- The font of the labels.

The color of the labels is governed by the color of the axes. See *Customizing the X–Y plot axis color and style*, for more information.

1. Locate the **Labels** options.

   Select **Options->XY Options->Axis**; or click ⟼, which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Axes** tab in the dialog box that appears.

   The **Labels** options for the axes appear in the middle of the **Axes** page.

2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Specify where you would like axis labels to appear.

   From the **Placement** list in the **Labels** options, choose one of the following:

None

Abaqus/CAE hides axis labels for the selected axis.

Inside

Abaqus/CAE displays axis labels on the same side of the axis as the *X–Y* curves.

Outside

Abaqus/CAE displays axis labels on the side of the axis opposite the *X–Y* curves.

4. Click the **Frequency** arrows to specify how often labels are printed relative to each major tick mark. For example, if you select a frequency of two, labels will appear at every second major tick mark along the axis. If you select a frequency of zero, no labels will be displayed along the axis. You can request a label frequency of at most one greater than the number of major tick marks.
5. From the **Labels** options, select the format of the selected labels.
   a. Click the **Format** button to reveal the *X*-axis label format options.
   b. Choose one of the following:

Automatic

Very large values and very small values are expressed in scientific format. All remaining values are expressed with no exponent and using the specified number of significant digits.

Decimal

All values are expressed with no exponent and using the specified number of significant digits.

Engineering

Values greater than or equal to 1000 (or values less than or equal to 0.001) are expressed as a number ranging from 1 to 999 that is multiplied by 10 to the *n*th power, where *n* is a product of 3 (for example, `20.5E+03` or `17.76E+06`). All remaining values are expressed using the specified number of significant digits.

Scientific

All values are expressed as a number between 1 and 10 that is multiplied by an appropriate power of 10 (for example, `2.05E+04` or `1.776E+07`).

6. Click the **Precision** arrows to specify the desired number of significant digits (for non-decimal formats) or decimal places (for decimal format) in each label.
7. Select the font of the axis labels.
   a. Click .
   Abaqus/CAE displays the **Select Font** dialog box.
   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.
   c. Click **Apply** to see the effect of your font selections.

d. Click **OK** to close the **Select Font** dialog box.

The axis labels change to the selected font, size, and style.

8. Select the color of the axis labels.

a. Click the color sample .

Abaqus/CAE displays the **Select Color** dialog box.

b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

c. Click **OK** to close the **Select Color** dialog box.

The axis labels change to the selected color.

9. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis titles*
- *Using X–Y plot axis options*

Customizing the X–Y plot axis color and style

Click the **Axes** tab to customize the color and thickness of the graph axes of an *X–Y* plot. Your selections apply to both the *X*- and *Y*-axes; it is not possible to customize the axes individually.

1. Locate the **Style** options.

   Select **Options->XY Options->Axis**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Axes** tab in the dialog box that appears.

   The **Style** options for the axes appear at the bottom of the **Axes** page.
2. From the **X Axis** or **Y Axis** fields, highlight one or more axes.
3. Choose the line style.

   From the **Style** list, click the desired line style (solid, dashed, etc.).

   The specified line style appears for the selected axis and in the **Style** list.
4. Choose the line thickness.

   From the **Thickness** list, click the desired line thickness.

   The specified line thickness appears for the selected axis and in the **Thickness** list.
5. Choose the line color:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The axis line changes to the selected color.
6. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

- *Customizing X–Y plot axis tick marks*
- *Customizing the X–Y plot axis labels*
- *Using X–Y plot axis options*

Customizing X–Y curve appearance

This section explains how you use the **Curve Options** dialog box to customize the appearance of a curve representing a selected *X–Y* data object.
Only curves that appear in the most recent *X–Y* plot are available for customization.

In this section:

- *Using options to customize the appearance of X–Y curves*
- *Selecting one or more X–Y curves to customize*
- *Customizing the X–Y plot legend*
- *Customizing the appearance of an X–Y curve*
- *Customizing the symbols used on an X–Y curve*
- *Editing the list of auto-colors for X–Y plots*

Using options to customize the appearance of X–Y curves

To customize the appearance of *X–Y* curves, expand the **Curves** container in the Results Tree, highlight the curves you want to modify, then click mouse button 3 and select **Curve Options** from the list that appears.
You can customize the line style, symbol style, and legend text of each curve.

Only those curves that appear in the most recent *X–Y* plot are available for customization. You can customize the following:

- The text that appears in the legend describing the data. For more information, see *Customizing the X–Y plot legend*.
- The appearance of the line representing the data (the curve). For more information, see *Customizing the appearance of an X–Y curve*.
- The appearance of the symbols along the curve. For more information, see *Customizing the symbols used on an X–Y curve*.

1. Locate the **Curve Options** dialog box. You can open this dialog box using any of the following methods:
   - From the main menu bar, select **Options->XY Options->Curve**.
   - From the Visualization module toolbox, click .
   - From the Results Tree, expand the **Charts** container, highlight the plot or plots you want to modify, click mouse button 3, and select **Chart Options** from the list that appears.
2. Use the curve options to select and then customize the appearance of the *X–Y* curves in the current viewport.
3. When you have finished customizing the appearance of your *X–Y* curves, click **Dismiss** to close the **Curve Options** dialog box.

Additional information

- *Selecting one or more X–Y curves to customize*
- *Customizing the X–Y plot legend*
- *Customizing the appearance of an X–Y curve*
- *Customizing the symbols used on an X–Y curve*

Selecting one or more X–Y curves to customize

To customize a curve, you first select it and then choose customization options. Use the **Curves** field in the **Curve Options** dialog box to select one or more *X–Y* curves to customize. The **Curves** field lists the displayed *X–Y* curves, identifying each curve by name. For information on curve naming conventions, see *Understanding "Temp" and other X–Y data object names*.

After you have selected one or more *X–Y* curves to customize, you then use the options in the **Attributes** frame to control the line and the symbol representing the data and to enter the text that will appear in the legend.

1. Locate the **Curves** field.

   The **Curves** field is at the top of the dialog box.

2. Select one or more *X–Y* curves to customize. For more information on selecting multiple items in dialog boxes, see *Selecting multiple items from lists and tables*.

   Note:

   To make an *X–Y* curve available for selection, you must first plot it.

   If you have selected one curve, the **Attributes** frame displays the current attributes of that curve.

   If you have selected multiple curves, Abaqus/CAE displays the **Legend text** field as blank. The display in the **Attributes** frame varies depending on whether the selected curves have the same attributes or not. If the attributes are the same, Abaqus/CAE displays them; if the attributes differ, Abaqus/CAE displays the following:

   - The **Show** and **Show symbol** checkboxes are gray.
   - The line **Color** and symbol **Color** text fields display the term **As is**.
   - The **Style**, **Thickness**, **Symbol**, and **Size** buttons are blank.
   - The **Frequency** text field is blank.

3. From the **Attributes** field, you can control the following:

   - The text that appears in the legend describing the data. For more information, see *Customizing the X–Y plot legend*.
   - The appearance of the line representing the data (the curve). For more information, see *Customizing the appearance of an X–Y curve*.
   - The appearance of the symbols along the curve. For more information, see *Customizing the symbols used on an X–Y curve*.

Customizing the X–Y plot legend

The legend of an *X–Y* plot provides a key to associate each curve included in the plot with the *X–Y* data object that it represents. Each legend entry contains a line segment in the color and style shown in the plot. The legend entry also includes text stating, by default, the name of the *X–Y* data represented by the curve. You can customize this text using the **Curve Options**.

The remaining characteristics of the legend are controlled by options in the **Viewport Annotation Options** dialog box. For information on customizing the legend and on including in the legend a report of the minimum and maximum *X–Y* values, see *Customizing the legend*.

1. Locate the **Attributes** options for X–Y curves.

   The **Attributes** frame is at the bottom of the **Curve Options** dialog box.
2. In the **Curves** options, select one or more curves whose legend text you want to modify.
3. In the **Legend Text** field, enter the custom legend text. You can also revert to the default legend text by toggling on **Use default**.
4. Click **Dismiss** to close the **Curve Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the appearance of an X–Y curve

You can customize the appearance of an X–Y curve.

Use the **Curve Options** dialog box to customize the color, style, and thickness of the line used to represent an *X–Y* data object in an *X–Y* plot. *Figure 1* illustrates some of the ways in which *X–Y* data curves can be customized.

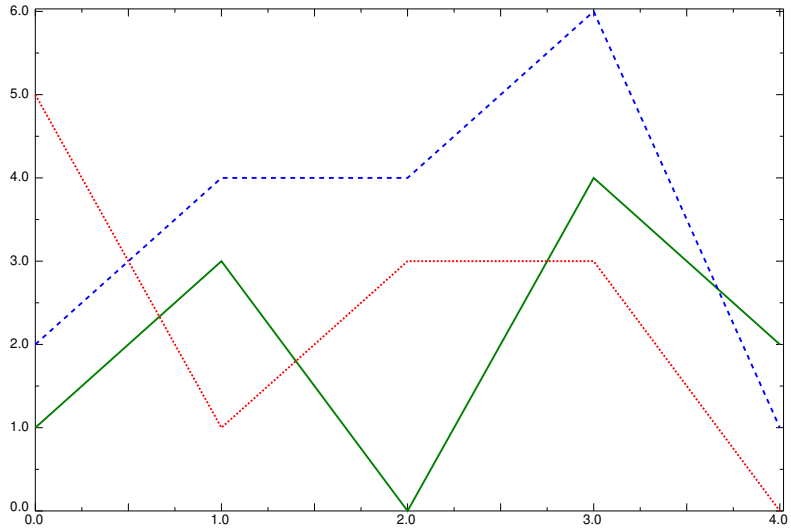


Figure 1: *X–Y* plot with customized curves.

The color, style, and thickness that you select appear along the curve and in the *legend*. The customization options are available only when **Show** is toggled on.

By default, Abaqus/CAE colors the *X–Y* curves in your plot by referring to the list of *X–Y* auto colors. You can modify this list from the **Edit XY-Auto-Colors** dialog box; see *Editing the list of auto-colors for X–Y plots*.

1. Locate the **Show** options.

   The **Show** options appear in the lower right corner of the **Curve Options** dialog box.

2. From the **Curves** field, *select one or more X–Y curves* whose attributes you wish to customize.

   Note:

   To make an *X–Y* curve available for selection, you must first plot it.

   The customization options you choose will apply to all of the curves you have selected; if you want to vary the appearance of multiple curves, you must customize them one at a time.

3. Toggle **Show** to display or suppress the line representing each selected *X–Y* curve.

   When **Show** is on, the curve is displayed and the line attribute options are enabled.

4. Choose the line color:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The selected *X–Y* curve changes to the selected color.

5. Choose the line style:
   a. Click the **Style** button to reveal the line style options (solid, dashed, etc.).
   b. From the style list, click the desired line style.

      The selected *X–Y* curve changes to the selected style.

6. Choose the line thickness:
   a. Click the **Thickness** button to reveal the line thickness options.
   b. From the thickness list, click the desired line thickness.

      The selected *X–Y* curve changes to the specified thickness.

7. Click **Dismiss** to close the **Axis Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the symbols used on an X–Y curve

You can customize the symbols used on an X–Y curve.

Use the **Curve Options** dialog box to customize the appearance of the symbols used to represent data points on an *X–Y* curve. For example, in *Figure 1* the plot on the left uses the default symbols, while the plot on the right uses customized symbols.

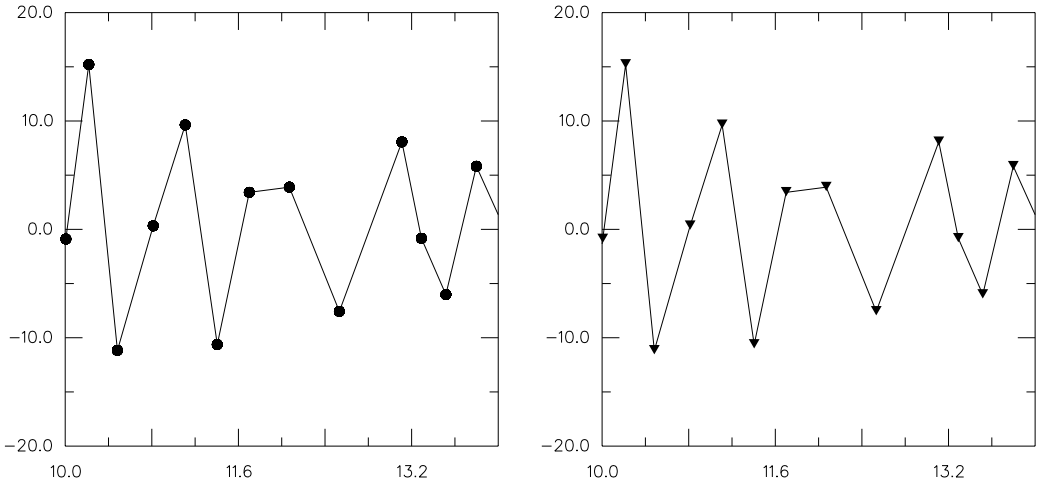


Figure 1: X–Y plots with different data point symbols.

The symbol that you select appears along the curve and in the *legend*. The customization options are available only when **Show symbol** is toggled on.

1. Locate the **Attributes** options.

   The **Attributes** options appear in the lower half of the **Curve Options** dialog box.

2. From the **Curves** field, *select one or more X–Y curves* whose symbols you wish to customize.

   Note:

   To make an *X–Y* curve available for selection, you must first plot it.

   The customization options you choose will apply to all of the curves you have selected; if you want to vary the symbols on multiple curves, you must customize them one at a time.

3. Toggle **Show symbol** to display or suppress the symbols representing each selected *X–Y* curve's data points.

   When **Show symbol** is on, symbols are displayed and the symbol attributes are enabled.

4. Choose the symbol color:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **XY Plot Options** dialog box.

5. Choose the symbol:

a. Click the **Symbol** button to reveal the choice of symbols.

b. From the list of symbols, click the desired symbol.

The specified symbol appears on the selected *X–Y* curve.

6. Choose the symbol size:

a. Click the **Size** button to reveal the symbol size options.

b. From the symbol size options, click the desired symbol size.

The specified symbol size appears on the selected *X–Y* curve.

7. Select the frequency of the symbols.

Type an integer into the **Frequency** text field to choose how often symbols appear along the curve. For example, if you enter a frequency of two, Abaqus displays symbols at every second data point. The value zero is not allowed; to suppress the display of symbols, toggle off **Show symbol**.

8. Click **Dismiss** to close the **Curve Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Editing the list of auto-colors for X–Y plots

Abaqus/CAE assigns colors to *X–Y* curves automatically by referring to the color definitions in the *X–Y* auto colors list. This list provides colors for *X–Y* curves in the same way that the general **Auto Colors** list is used to color geometry and mesh elements, but these two color lists are separate entities.

This section describes how to edit the *X–Y* auto colors list to modify the set of default color assignments for *X–Y* curves. You can also change the color assignment for an individual *X–Y* curve from the **Curve Options** dialog box. See *Customizing the appearance of an X–Y curve*.

1. In the Results Tree, click mouse button 3 on the **XY Plots** container, then select **Edit XY-Auto-Colors** from the list that appears.

   Abaqus/CAE displays the **Edit XY-Auto-Colors** dialog box.

2. To insert a new color into the *X–Y* auto colors list:

   1. Highlight the color in the *X–Y* auto colors list that you want to precede or follow the color that you add.
   2. Click **Insert Before** or **Insert After** to add a new color in the indicated location in the *X–Y* auto colors list.

      The **Select Color** dialog box opens.
   3. Choose a color by following the steps described in *Customizing colors*; then click **OK** to close the **Select Color** dialog box.

   Abaqus/CAE adds the new color in the selected location.

3. To change one of the colors in the *X–Y* auto colors list, double-click the color and select a new color from the **Select Color** dialog box that appears.
4. To move a color within the *X–Y* auto colors list, highlight the color and click either **Move Up** or **Move Down** to move the color one step in the selected direction.
5. To delete a color from the *X–Y* auto colors list, highlight the color and click **Delete**.
6. Continue adding, changing, moving, and deleting colors until the *X–Y* auto colors list contains the colors you want in the desired order.
7. Click **OK**.

   Abaqus/CAE uses the revised *X–Y* auto colors list for any subsequent color coding of *X–Y* curves.

Customizing X–Y plot appearance

This section explains how you use the **Chart Options** dialog box to customize the display of gridlines and to adjust the general shape, size, and position of an *X–Y* plot.

In this section:

- *Customizing the gridlines for an X–Y plot*
- *Customizing the border and fill color for an X–Y plot*
- *Customizing X–Y plot size and shape*
- *Customizing the position of an X–Y plot*

Customizing the gridlines for an X–Y plot

You can customize the gridlines for an X–Y plot.

Click the **Grid Display** tab to control the appearance of the major and minor gridlines of an *X–Y* plot. Examples of major and minor gridlines are shown in *Figure 1*.

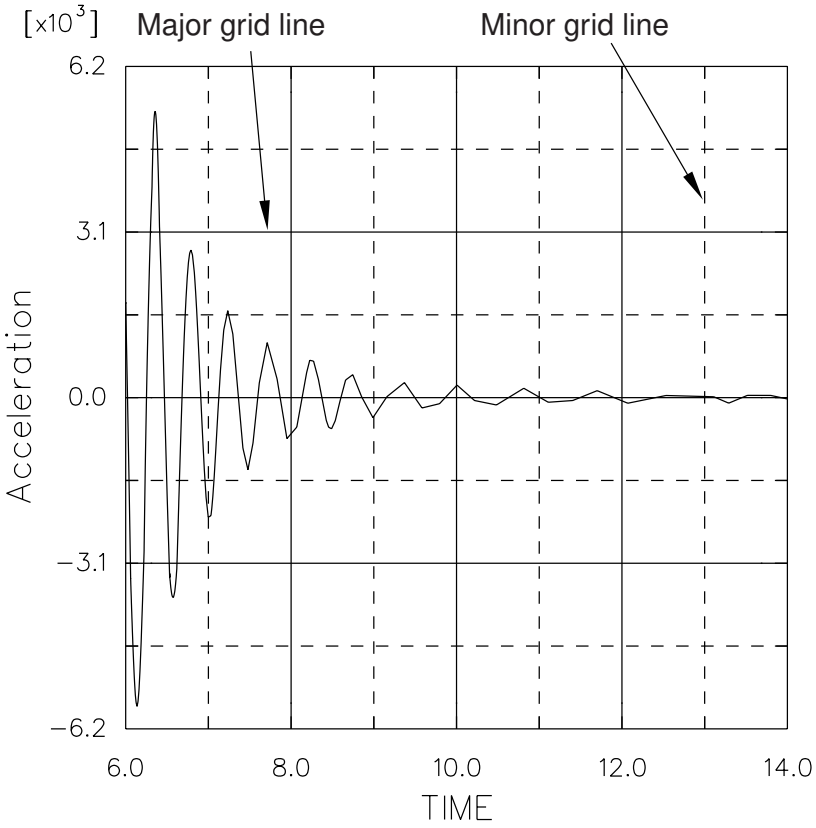


Figure 1: *X–Y* plot with major and minor gridlines indicated.

If your *X–Y* plot uses multiple axes in the same direction, Abaqus/CAE draws gridlines only for the axes of the first curve in the plot.

The **Grid Display** options provide independent control over each of the four sets of gridlines; that is, independent control over the major and minor gridlines in either direction. You can display or hide any of the four sets, and you can modify the color, style, or thickness of the lines for any of the four sets of gridlines.

Note:

Major gridlines originate from major tick marks, and minor gridlines originate from minor tick marks. To control the interval between gridlines, you must adjust the *spacing of the tick marks*.

1. Locate the axis **Grid Display** options.

   Select **Options->XY Options->Chart**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Grid Display** tab in the dialog box that appears.

2. Select one or more *X–Y* plots to customize from the **Charts** field.

3. From the **X Gridlines** options, toggle on the **Major** or **Minor** options to display major or minor vertical gridlines, respectively.

   When either **Major** or **Minor** is on, gridline style, color, and thickness options become available for that type of gridline.

4. From the **Y Gridlines** options, toggle on **Major** or **Minor** to display major or minor horizontal gridlines, respectively.

5. Repeat the following steps to customize the horizontal major and minor gridlines and the vertical major and minor gridlines:

   a. Choose the color for the gridlines:

      1. Click the color sample .

         Abaqus/CAE displays the **Select Color** dialog box.

      2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

      3. Click **OK** to close the **Select Color** dialog box.

         The selected gridlines change to the specified color.

   b. Choose the line style:

      1. Expand the style list to reveal the line style options (solid, dashed, etc.). The style list is the upper of the two unlabeled list options.

      2. From the style list, click the desired line style.

         The selected gridlines change to the specified style.

   c. Choose the line thickness:

      1. Expand the thickness list to reveal the line style options (solid, dashed, etc.). The style list is the lower of the two unlabeled list options.

      2. From the thickness list, click the desired line thickness.

         The selected gridlines change to the specified thickness.

6. Click **Dismiss** to close the **Chart Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the border and fill color for an X–Y plot

Click the **Grid Display** tab to access the **Grid Area** options for an *X–Y* plot. These options enable you to customize the grid in an *X–Y* plot, which is the rectangular area where the curves are plotted. You can wrap a border around the grid or fill the grid with a background color.

Note:

You can also control border and background color settings for the entire *X–Y* plot, which includes the portion of the viewport covered by an *X–Y* plot's grid and all of its axes. Abaqus/CAE displays the background color for the grid on top of the background for the entire *X–Y* plot. For more information, see *Customizing border and fill colors for an X–Y plot*.

1. Locate the **Grid Area** options.

   Select **Options->XY Options->Chart**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Grid Display** tab in the dialog box that appears.

   The **Grid Area** options are available at the bottom of the **Grid Display** page.
2. Select one or more *X–Y* plots from the top part of the window.
3. Toggle on **Show border** to display a border around the entire grid area.
4. If desired, change the color of the border by performing the following steps:
   a. Click the color sample immediately to the right of the **Show border** label .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample and the grid border change to the selected color.
5. Toggle on **Fill** to fill the grid area with the color displayed to the right of the **Fill** label.
6. If desired, change the grid area fill color by performing the following steps:
   a. Click the color sample immediately to the right of the **Fill** label .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The grid area changes to the selected color.
7. Click **Dismiss** to close the **Chart Options** dialog box.

Additional information

- *Customizing border and fill colors for an X–Y plot*

Customizing X–Y plot size and shape

Use the **Size** options in the **Chart Options** dialog box to customize the size and shape of the *X–Y* plot.

1. Locate the **Chart Options**.

   Select **Options**->**XY Options**->**Chart**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Grid Position** tab in the dialog box that appears.
2. From the **Charts** field, select one or more *X–Y* plots whose sizes or shapes you wish to customize.
3. Select one of the following *X–Y* plot sizing and shaping options:

   ### Fit to chart

   Abaqus/CAE fills the entire viewport with the selected *X–Y* plot, adjusting the shape of the plot to match the viewport dimensions.

   ### Square

   The **Size** slider becomes available, which you can drag to increase or decrease the size of the square-shaped *X–Y* plot.

   ### Manual

   The **X** and **Y** sliders become available, which you can drag to increase or decrease the size of the *X–Y* plot along either axis.
4. Click **Dismiss** to close the **Chart Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the position of an X–Y plot

Use the **Position** options in the **Chart Options** dialog box to customize the position of an *X–Y* plot in the viewport.

1. Locate the **Position** options.

   Select **Options**->**XY Options**->**Chart**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Grid Position** tab in the dialog box that appears.

2. From the **Charts** field, select one or more *X–Y* plots whose positions or shapes you wish to customize.
3. Select one of the following *X–Y* plot positioning options:

Auto-align

This option enables you to align the selected *X–Y* plot in one of nine positions in the viewport: its center, any of the four corners, or centered along any of the four edges. Expand the **Alignment** list, then select one of the options depicted graphically in the list.

Manual

This option enables you to position the *X–Y* plot origin at a specific *X–Y* location in the viewport. Drag the **X** and **Y** sliders to move the lower left corner of the *X–Y* plot horizontally or vertically in the viewport. When these sliders' values are set to `0`, the *X–Y* plot origin is placed in the lower left corner of the viewport.

4. Click **Dismiss** to close the **Chart Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the X–Y plot title

This section explains how you use the **Plot Title Options** dialog box to create a title for an X–Y plot and to customize the title's appearance and location in the viewport.

In this section:

- *Customizing the contents of the X–Y plot title*
- *Customizing the location of the X–Y plot title*
- *Adding a bounding box and fill color for the X–Y plot title*

Customizing the contents of the X–Y plot title

Use the **Text** options in the **Plot Title Options** to create and display a custom title for an X–Y plot.

1. Locate the **Text** options for plot titles.

   Select **Options->XY Options->Plot Title**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

2. Click the **Title** tab, if it is not already selected.
3. Select a title for the X–Y plot using one of the following methods:

   - To display the default name of the X–Y plot, toggle on **Use default**.
   - To display a custom title, toggle off **Use default**, enter a title in the **Title** field, and press `Enter`.

4. Select the font of the plot title.

   a. Click .

   Abaqus/CAE displays the **Select Font** dialog box.

   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.
   c. Click **Apply** to see the effect of your font selections.
   d. Click **OK** to close the **Select Font** dialog box.

   The X–Y plot title changes to the selected font, size, and style.

5. Select a color for the X–Y plot title.

   a. Click the color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

   The plot title text changes to the selected color.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the location of the X–Y plot title

Use the **Position** options in the **Plot Title Options** to place the X–Y plot title in a particular location in the viewport.

1. Locate the **Position** options.

   Select **Options**->**XY Options**->**Plot Title**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Area** tab from the dialog box that appears.
2. Toggle on **Inset** to display the plot title legend within the *X–Y* plot. If **Inset** is off, Abaqus/CAE keeps the plot title and the *X–Y* plot in separate bands of the viewport.
3. Choose one of the following two methods for positioning the plot title:

Auto-align

This option enables you to position the plot title in one of nine positions in the viewport: its center, any of the four corners, or centered along any of the four edges. Expand the **Alignment** list, then select one of the options depicted graphically in the list.

Manual

This option enables you to position the plot title in a specific *X–Y* location in the viewport. Drag the **X** and **Y** sliders to move the title horizontally or vertically in the viewport. When both of these values equal `0`, the plot title is positioned in the lower left corner of the viewport; when both values equal `100`, the plot title is positioned in the upper right corner.

Additional information

• *Selecting one or more X–Y curves to customize*

Adding a bounding box and fill color for the X–Y plot title

Use the **Bounding Box** options in the **Plot Title Options** to draw a rectangular box around the X–Y plot title or to fill the area behind the plot title with a background color.

1. Locate the **Bounding Box** options.

   Select **Options->XY Options->Plot Title**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox. Click the **Area** tab from the dialog box that appears. The **Bounding Box** options appear at the bottom of the dialog box.
2. Select one or more *X–Y* plots from the **Charts** field at the top of the dialog box.
3. Perform the following steps to display and modify the bounding box:
   a. Toggle on **Show border** to display a border around the legend.
   b. Select a color for the bounding box line.

      1. Click the color sample .

         Abaqus/CAE displays the **Select Color** dialog box.
      2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
      3. Click **OK** to close the **Select Color** dialog box.

         The bounding box line changes to the selected color.
4. Control the bounding box fill color:
   a. Toggle on **Fill** to display a background color in the bounding box.
   b. Select a fill color to display in the background of the bounding box.

      1. Click the color sample .

         Abaqus/CAE displays the **Select Color** dialog box.
      2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
      3. Click **OK** to close the **Select Color** dialog box.

         The bounding box background changes to the selected color.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the appearance of the X–Y plot legend

This section explains how you use the **Chart Legend Options** dialog box to customize the appearance and contents of the legend for an *X–Y* plot. You can also display a title on the *X–Y* plot legend, and you can include on the legend a report of the minimum and maximum *X–Y* values.

In this section:

- *Displaying or hiding the X–Y plot legend*
- *Customizing the title of the X–Y plot legend*
- *Adding minimum and maximum values to the X–Y plot legend*
- *Customizing the location of the legend*
- *Customizing the border and fill for the X–Y plot legend*

Displaying or hiding the X–Y plot legend

The *X–Y* plot legend provides a key to associate each curve included in the plot with the *X–Y* data object that it represents. Each legend entry contains a line segment in the color and style shown in the plot. The legend entry also includes text stating, by default, the description of the *X–Y* data object represented by the curve. You can display or hide the legend using the **Chart Legend Options**.

1. Locate the **Chart Legend Options**.

   Select **Options->XY Options->Chart Legend**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   Tip: You can also open this dialog box directly by double-clicking the legend in the viewport when an *X–Y* plot is displayed. You cannot select a component in an *X–Y* plot directly from the viewport when an active procedure is in effect; cancel the current procedure before selecting items from an *X–Y* plot.

2. Click the **Contents** tab, if it is not already selected.
3. Toggle on **Show legend** to display the legend, or toggle it off to hide the legend.
4. Click **Dismiss** to close the **Chart Legend Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the title of the X–Y plot legend

Use the **Title** options to display or hide a title for the *X–Y* plot legend and to customize its contents.

1. Locate the **Text** options.

   Select **Options->XY Options->Chart Legend**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   Tip: You can also open this dialog box directly by double-clicking the legend in the viewport when an *X–Y* plot is displayed. You cannot select a component in an *X–Y* plot directly from the viewport when an active procedure is in effect; cancel the current procedure before selecting items from an *X–Y* plot.

   The **Text** options for the legend appear in the lower half of the **Contents** page.

2. From the **Charts** field, select one or more *X–Y* plots whose legends you wish to customize.

   Note:

   To make an *X–Y* curve available for selection, you must first plot it.

3. Click the **Contents** tab, if it is not already selected.
4. If desired, enter a title for the *X–Y* plot legend in the **Title** field.
5. Select the font of the *X–Y* plot legend title.

   a. Click .

   Abaqus/CAE displays the **Select Font** dialog box.

   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.
   c. Click **Apply** to see the effect of your font selections.
   d. Click **OK** to close the **Select Font** dialog box.

   The X–Y plot legend title changes to the selected font, size, and style.

6. Select the color of the *X–Y* plot legend title.

   a. Click the color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

   The X–Y plot legend title text changes to the selected color.

7. Click **Dismiss** to close the **Chart Legend Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Adding minimum and maximum values to the X–Y plot legend

You can include on the plot legend a report of the minimum and maximum *X–Y* values in the plot, and you can customize the format and precision of these values.

1. Locate the **Chart Legend Options**.

   Select **Options->XY Options->Chart Legend**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   Tip: You can also open this dialog box directly by double-clicking the legend in the viewport when an *X–Y* plot is displayed. You cannot select a component in an *X–Y* plot directly from the viewport when an active procedure is in effect; cancel the current procedure before selecting items from an *X–Y* plot.

2. From the **Charts** field, select one or more *X–Y* plots whose legends you wish to customize.
3. Click the **Contents** tab, if it is not already selected.
4. Toggle on **Show min/max** to report the minimum and maximum *X–Y* values in the legend.
5. Select one of the following number formats for the minimum and maximum values displayed in the legend:

   ### Automatic

   Very large values and very small values are expressed in scientific format. All remaining values are expressed with no exponent and using the specified number of significant digits.

   ### Decimal

   All values are expressed with no exponent and using the specified number of significant digits.

   ### Engineering

   Values greater than or equal to 1000 (or values less than or equal to 0.001) are expressed as a number ranging from 1 to 999 that is multiplied by 10 to the *n*th power, where *n* is a product of 3 (for example, `20.5E+03` or `17.76E+06`). All remaining values are expressed using the specified number of significant digits.

   ### Scientific

   All values are expressed as a number between 1 and 10 that is multiplied by an appropriate power of 10 (for example, `2.05E+04` or `1.776E+07`).

6. Click the **Precision** arrows to specify the desired number of significant digits (for non-decimal formats) or decimal places (for decimal format) in each label.
7. Click **Dismiss** to close the **Chart Legend Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the location of the legend

You can move the *X–Y* plot legend by dragging it around the viewport or by specifying a viewport location from the **Position** options in the **Chart Legend Options** dialog box. You can drag the legend only when Abaqus/CAE does not have a procedure active; if you cannot drag the legend or select any other *X–Y* plot components from the viewport, cancel the current procedure.

1. Locate the **Chart Legend Options**.

   Select **Options->XY Options->Chart Legend**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   Tip: You can also open this dialog box directly by double-clicking the legend in the viewport when an *X–Y* plot is displayed. You cannot select a component in an *X–Y* plot directly from the viewport when an active procedure is in effect; cancel the current procedure before selecting items from an *X–Y* plot.

2. From the **Charts** field, select one or more *X–Y* plots whose legends you wish to move.
3. Click the **Area** tab, if it is not already selected.
4. Toggle on **Inset** to display the *X–Y* plot legend within the *X–Y* plot grid. If **Inset** is off, Abaqus/CAE keeps the legend and the *X–Y* plot in separate bands of the viewport.
5. Select one of the following positioning options:

Auto-align

This option enables you to align the legend in one of nine positions in the viewport: its center, any of the four corners, or centered along any of the four edges. Expand the **Alignment** list, then select one of the options depicted graphically in the list.

Manual

This option enables you to position the legend at a specific *X–Y* location in the viewport. Drag the **X** and **Y** sliders to move the legend horizontally or vertically in the viewport. When these sliders' values are set to `0`, the legend is placed in the lower left corner of the viewport.

6. Click **Dismiss** to close the **Chart Legend Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing the border and fill for the X–Y plot legend

Use the **Bounding Box** options in the **Chart Legend Options** dialog box to customize the appearance of the legend by wrapping it with a border or filling it with a specified color.

1. Locate the **Chart Legend Options**.

   Select **Options->XY Options->Chart Legend**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   Tip: You can also open this dialog box directly by double-clicking the legend in the viewport when an *X–Y* plot is displayed. You cannot select a component in an *X–Y* plot directly from the viewport when an active procedure is in effect; cancel the current procedure before selecting items from an *X–Y* plot.

2. From the **Charts** field, select one or more *X–Y* plots whose legends you wish to customize.

   Note:

   To make an *X–Y* curve available for selection, you must first plot it.

3. Click the **Area** tab, if it is not already selected.
4. Perform the following steps to display and modify the bounding box:
   a. Toggle on **Show border** to display a border around the legend.
   b. Select a color for the bounding box line.

      1. Click the color sample .

         Abaqus/CAE displays the **Select Color** dialog box.

      2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
      3. Click **OK** to close the **Select Color** dialog box.

         The bounding box line changes to the selected color.

5. Control the bounding box fill color:
   a. Toggle on **Fill** to display a background color in the bounding box.
   b. Select a fill color to display in the background of the bounding box.

      1. Click the color sample .

         Abaqus/CAE displays the **Select Color** dialog box.

      2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
      3. Click **OK** to close the **Select Color** dialog box.

         The bounding box background changes to the selected color.

6. Click **Dismiss** to close the **Chart Legend Options** dialog box.

Additional information

• *Selecting one or more X–Y curves to customize*

Customizing border and fill colors for an X–Y plot

Use the options in the **Plot Options** dialog box to create a custom border color and a custom background fill color for one or more *X–Y* plots in your session. By default, *X–Y* plots have no border and they are filled with a light grey background color. You might want to adjust the border color to distinguish between different *X–Y* plots used in an overlay plot; you might want to adjust the fill color if the *X–Y* curves used in your plot do not stand out clearly against the default background.

1. Select **Options->XY Options->Plot Options**; or click , which is located with the *X–Y* plotting tools in the Visualization module toolbox.

   The **Plot Options** dialog box opens.
2. From the **Plots** frame, highlight one or more *X–Y* plots that you want to modify. This frame lists plot names and the viewports in which they are displayed.
3. Toggle on **Show border** to display a border around the *X–Y* plot.
4. If desired, change the color of the *X–Y* plot border by performing the following steps:
   a. Click the color sample immediately to the right of the **Show border** label .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample and the *X–Y* plot border change to the selected color.
5. Toggle on **Fill** to fill the *X–Y* plot with the color displayed to the right of the **Fill** label.
6. If desired, change the *X–Y* plot fill color by performing the following steps:
   a. Click the color sample immediately to the right of the **Fill** label .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The *X–Y* plot fill color changes to the selected color.

Viewing results along a path

A path is a series of connected lines that you define by specifying a series of points or edges through your model. You can view results along the path in the form of an *X–Y* plot. This chapter explains how to view results along a path.

In this section:

- *Understanding results along a path*
- *Creating a path through your model*
- *Obtaining X–Y data along a path*

Understanding results along a path

To view results along a path through your model, you first select **Tools->Path** to specify the path and then select **Tools->XY Data** to obtain *X*–*Y* data along the path.

In this section:

- *Path specification*
- *How Abaqus/CAE obtains results along a path*

Path specification

A path is a line you define by specifying a series of points or line segments through your model. The points can be nodes or coordinate locations; the line segments can be element edges or lines joining the ends of element edges. Paths can cross more than one part instance. Abaqus/CAE offers four distinct types of paths:

Node list

The points making up the path consist of nodal locations only. You specify the points using node labels and node label ranges. A node label is a persistent means of referring to a given node; that is, node labels do not change as your model deforms. Because of this, the node labels that comprise a node list path are equally applicable to the undeformed or the deformed model shape. However, node labels are part instance-specific. In other words, you can have the same node label for multiple part instances; therefore, you must specify to which part instance you are referring when you use node labels. For more information, see *Creating or editing a node list path*.

Point list

The points making up the path consist of coordinate locations within your model. These locations may or may not coincide with nodal locations. Point list coordinates remain fixed in space and are independent of your model. For example, coordinates that coincide with a nodal location on the undeformed shape may not coincide with any location on the deformed shape. By the same logic, point list coordinates are independent of specific part instances. For more information, see *Creating or editing a point list path*.

Edge list

The points making up the path consist of edges that connect nodes within your model. You specify the edges by selecting individual element edges from the viewport or by selecting the starting edge and direction and allowing Abaqus/CAE to automatically complete the path to the end of the feature edge or to an endpoint that you select. Edges are defined by the elements in the model. Because of this, the edges that comprise an edge list path are equally applicable to the undeformed or the deformed model shape. However, element labels are part instance-specific. In other words, you can have the same element label for multiple part instances; therefore, you must specify to which part instance you are referring when you use edges. For more information, see *Creating or editing an edge list path*.

Circular

The points making up the path consist of coordinate locations within your model. You can create two types of circular paths in Abaqus/CAE: circumferential and radial. A circumferential path lies along the edge of a circle or an arc; a radial path lies along a radius. You specify either type of path by selecting coordinates that define a circle or circular arc and then defining the starting point, ending point, and number of points along the path. Once you have defined the original circle or arc, you can select a new radius length to create any path that shares the same center point as the original. You can use circumferential paths to obtain cross-sectional results in a curved portion of a model, as shown in *Figure 1*.

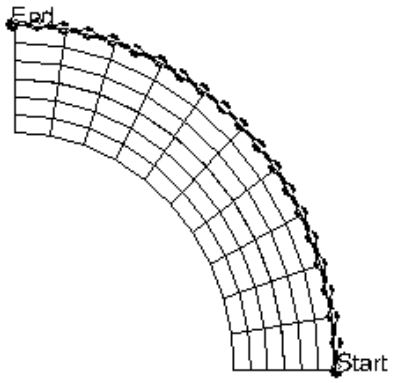


Figure 1: A circumferential path.

You can use radial paths to obtain results spanning from the inner to the outer face of a curve, as shown in *Figure 2*; radial paths are ideal for use in stress linearization (for more information on stress linearization, see *Calculating linearized stresses*).

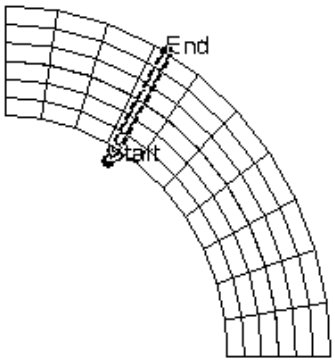


Figure 2: A radial path.

You can select coordinates by picking nodes from the viewport or by entering values in the prompt area; in either case, the coordinates remain fixed in space and are independent of your model. For example, coordinates that coincide with a nodal location on the undeformed shape may not coincide with any location on the deformed shape. By the same logic, circular path coordinate definitions are independent of specific part instances. For more information, see *Creating or editing a circular path*.

Abaqus/CAE forms the path by connecting the nodes, points, or edges that you specify in the order you have given.

After you create a path, you can select **Tools->Path** from the main menu bar to edit, copy, rename, delete, or plot it. Plotting the path itself is a means to verify visually that you have specified the intended line; to view results along the path, you must form *X–Y* data pairs and produce an *X–Y* plot. For more information on managing paths, see *Managing objects using manager dialog boxes*; for more information on producing an *X–Y* plot of path data, see *Obtaining X–Y data along a path*.

How Abaqus/CAE obtains results along a path

Abaqus/CAE obtains results along a path in the form of *X*–*Y* data pairs.
Abaqus determines the data pair *X*-values based on the points that make up the path. These points define model locations at which to obtain data. Abaqus determines the data pair *Y*-values based on analysis results at those model locations. Abaqus/CAE considers only the entities in the current display group when calculating the data pairs.

Path data X-values

You can choose to form *X*-values based only on the points you have specified for the path or to additionally include all locations at which the path intersects the model. (An intersection occurs where the path crosses an element face, element edge, surface face, or surface edge. You can omit individual intersections from the path definition if the intersection lies within a user-specified distance of an actual data point. You can set this intersection tolerance value using Abaqus Scripting Interface commands; for more information, see *Session object*.)

You can also choose whether Abaqus/CAE interprets the points that make up the path as locations on the undeformed or the deformed model shape. When the deformed model shape is chosen, Abaqus/CAE computes the deformed coordinates at a node as the sum of the base coordinates and the deformation at that node multiplied by the viewport-specific deformation scale factor. Node list labels and edge lists indicate model locations and are equally applicable to the undeformed or the deformed model shape. Point list and circular path coordinates are fixed locations independent of the model geometry. Abaqus does not form data pairs for points that do not coincide with the model shape you have chosen.

The node labels or point coordinates that comprise a path are usually not in a form suitable for direct use as *X*-values. Abaqus/CAE offers several options for you to convert this series of points to useful *X*-values and subsequent *X*–*Y* plot axis labels. You can choose one of the following options to convert path points to *X*-values:

- **True distance**: *X*-values correspond to each point's actual distance along the path in model space coordinates, starting with zero.
- **Normalized distance**: *X*-values correspond to each point's distance along the path as a fraction of the total length of the path.
- **Sequence ID**: *X*-values correspond to the order in which each point occurs in the path result list.
- **X, Y, or Z distance**: *X*-values correspond to each point's actual distance along the path in the single coordinate direction you specify, starting with zero. This option is particularly useful for generating a plot of results versus radius in an axisymmetric model.
- **X, Y, or Z coordinate**: *X*-values correspond to each point's distance from the origin along the axis you specify.

Path data Y-values

You can control data pair *Y*-values by choosing the results step, frame, and field output variable for which Abaqus/CAE obtains results and by controlling how Abaqus/CAE computes certain types of results, as follows:

- For node-based field output variables such as displacement to be obtained at nodal locations, Abaqus/CAE reads results directly from the output database with no further computations.
- For element-based field output variables such as stress or strain to be obtained at nodal locations, Abaqus/CAE reads results from the output database, extrapolates those values to the nodes, then conditionally averages multiple contributions according to options you select.
- For both node-based and element-based variables to be obtained at path points that do not coincide with nodal locations, Abaqus/CAE computes values by interpolating from the nodes to the requested location using a geometric approximation of the element's shape. You cannot control this computation.

The averaging options for element-based variables and the complex form options for complex number values are located in the **Result Options** dialog box. Averaging reduces multiple contributions into a single value. When you select result options to partially or fully suppress averaging, path points receiving multiple contributions produce multiple data pairs. Such data pairs share the same *X*-value but have a separate *Y*-value for each contribution. Depending on your path and on the characteristics of your model, the following techniques may be necessary to avoid multiple data pairs sharing the same *X*-value:

- Set result options to fully enable averaging.
- Set result options to ignore region boundaries when computing values.
- Avoid paths that traverse region discontinuities.
- Avoid paths along a line separating discontinuous regions.
- Use display groups to isolate individual regions prior to obtaining results along a path.
- Avoid point list paths that traverse one or more spatial points that lie visually outside of highly deformed elements and cylindrical elements that subtend a large angle but, due to isoparametric mapping, lie mathematically inside the elements. Such spatial points tend to be shared by two or more elements, leading to multiple *Y*-values.

For more information on value averaging, see *Understanding result value averaging*.

The *Y*-value of the data pairs is affected by the current complex form if your selected field output variable contains complex number results. An abbreviation of the complex form is appended to the *Y*-axis title when you plot the path. For example, if `S-Mises` is the selected field output variable and `Magnitude` is the complex form, the *Y*-axis title is `S-Mises CPX:Mg`. Other complex forms are similarly abbreviated. For more information on complex forms, see *Controlling the form of complex results*.

Abaqus/CAE does not form data pairs for points that do not have results for the specified step, frame, or field output variable.

Creating a path through your model

A path is a line you define by specifying a series of points or line segments through your model; the points can be nodes or coordinate locations, the line segments can be element edges or lines joining the ends of element edges.

In this section:

- *Creating or editing a node list path*
- *Creating or editing a point list path*
- *Creating or editing an edge list path*
- *Creating or editing a circular path*

Creating or editing a node list path

A path is a line you define by specifying a series of points through your model. In a node list path all of the points are nodal locations.

For information on including locations other than nodes in a path, see *Creating a path through your model*.

You create a node list path by entering part instance names, node labels, and node label ranges into a table. You can use the editing techniques described in the following procedures to create a path by selecting nodes directly from the viewport or from node sets. Alternatively, you can create a node list path by entering values directly into the table. If you do not know the part instance names in the model, use the Query toolset to query nodes in the viewport. To help determine node labels of interest, you may want to produce a model plot with node symbols and node numbers visible prior to creating the path. For more information, see *Customizing model labels*.

1. Locate the path creation and editing options.

   Use the following techniques to create or edit a node list path:

   Create a new path

   a. From the main menu bar, select **Tools->Path->Create**. The **Create Path** dialog box appears.
   b. Abaqus/CAE displays a default name for the path in the **Name** text field. To provide a more meaningful name, replace this default with the name of your choice (including blank spaces if you wish).
   c. In the **Create Path** dialog box, click **Node list**; then click **Continue**.

   Edit an existing path

   a. From the main menu bar, select **Tools->Path->Edit**. From the menu that appears, select the path you want to edit.

      Tip: You can also use the **Path Manager** to edit a path. From the main menu bar, select **Tools->Path->Manager** to display the manager. Select the path you want to edit, and click **Edit** from the buttons on the right side of the manager.

   The **Edit Node List Path** dialog box appears.

2. Specify the nodes along your path.

   Use any of the following techniques:

   Select the nodes directly from the viewport

   a. Click **Add Before** or **Add After** to begin making **Viewport selections**.

      Abaqus/CAE prompts you to **Select nodes to be inserted into the path** and closes the **Edit Node List Path** dialog box.

   b. Select one or more nodes from the viewport (for more information, see *Selecting objects within the viewport*). If a desired node exists in more than one part instance, toggle off the **Select the Entity Closest to the Screen** tool in the **Selection** toolbar. If you make an ambiguous selection, the part instance name and node label will appear in the lower left corner of the viewport in the form `Highlighting Node Instance.Node`. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

      Abaqus/CAE highlights the nodes as you select them and displays the node labels and the path connectivity.

      Note:

      You can only select nodes that are in the current display group.

   c. Click **Done** in the prompt area when you have finished.

Abaqus/CAE displays the **Edit Node List Path** dialog box; the part instances and nodes in your path are listed in the path definition table.

d. If desired, repeat the preceding steps to add additional nodes.

The **Add Before** and **Add After** buttons determine where your selections will be added in relation to the currently highlighted line in the path definition table.

Select the nodes from a node set

a. Click **Add Before** or **Add After** to begin making **Node set selections**.

The **Select Node Sets** dialog box displays the node sets available in the current output database in the table on the left side.

b. To limit the list for easier selection, type a filter pattern into the **Filter** field, and press **[Enter]** to apply the filter.

Abaqus/CAE lists only those node sets that meet your specification.

c. From the list of node sets, select one or more node sets to include in the path.

- Use the top two arrows in the middle of the dialog box to move the selected node sets into or out of the table on the right side of the dialog box.
- Use the bottom two arrows in the middle of the dialog box to move all of the node sets into or out of the table on the right side of the dialog box.

d. Click **OK** when you have completed your selection.

Abaqus/CAE displays the **Edit Node List Path** dialog box; the part instances and nodes in your path are listed in the path definition table.

e. If desired, repeat the preceding steps to add additional nodes.

The **Add Before** and **Add After** buttons determine where your selections will be added in relation to the currently highlighted line in the path definition table.

Enter the node labels directly into the table

a. In the **Path Definition** table, use standard keyboard and mouse editing techniques to insert, modify, or delete part instance names and node label expressions. If you do not know the part instance names in the model, use the Query toolset to query nodes in the viewport. For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.) A node label expression is either of the following:

- A single node label; for example, 5.
- A range of nodes that you specify with a starting and ending node label optionally followed by an increment; for example, 5:10 or 5:10:2.



Note:

If you read node labels from an ASCII file, you must read the table values starting with the first column, and each row must include a part instance name and a node label expression. The part instance name is case sensitive and must be enclosed in quotation marks as follows:`"INSTANCE_NAME"`

b. After each node label expression, press **[Enter]**.

Abaqus highlights all nodes included in the path and displays the node labels and the path connectivity.

3. When you are done selecting nodes, click **OK**.

If you created a new path, Abaqus/CAE adds the path name to the **Path Manager** list. By default, Abaqus/CAE saves the path you have specified for the duration of the session. If you want to retain a path for use in subsequent sessions, save it to an XML file, to the model database, or to an output database; for more information, see *Managing session objects and session options*.

Additional information

- *Understanding results along a path*
- *Obtaining X–Y data along a path*

Creating or editing a point list path

A path is a line you define by specifying a series of points through your model. In a point list path all of the points are model coordinate locations. These locations may or may not coincide with nodes.
For information on alternative forms of path specification, see *Creating a path through your model*.

You create a point list path by entering the coordinates of points into a table. For points that coincide with nodal locations, you can specify the node label or select the node in the viewport and Abaqus will enter the undeformed coordinates of the node into the table. You can also select a node list path to enter nodal coordinates into the table. To help determine which node labels are of interest, you may want to produce a model plot with node symbols and node numbers visible prior to creating the path. For more information, see *Customizing model labels*.

1. Locate the path creation and editing options.

   Use the following techniques to create or edit a point list path:

   Create a new path

   a. From the main menu bar, select **Tools->Path->Create**. The **Create Path** dialog box appears.
   b. Abaqus displays a default name for the path in the **Name** text field. To provide a more meaningful name, replace this default with the name of your choice (including blank spaces if you wish).
   c. In the **Create Path** dialog box, click **Point list**; then click **Continue**.

   Edit an existing path

   a. From the main menu bar, select **Tools->Path->Edit**. From the menu that appears, select the path you want to edit.

      Tip: You can also use the **Path Manager** to edit a path. From the main menu bar, select **Tools->Path->Manager** to display the manager. Select the path you want to edit, and click **Edit** from the buttons on the right side of the manager.

   The **Edit Point List Path** dialog box appears.

2. In the **Point Coordinates** table, use standard keyboard and mouse editing techniques to enter, modify, or delete coordinates.

   To enter coordinates, use any of the following techniques:

   Type in the coordinate values of a point

   a. Specify the values as *X*-, *Y*-, *Z*-coordinates separated by spaces or commas; for example, 1.0, 3.4, 2.0.
   b. Press **[Enter]**.

   Include the coordinates of a nodal location by querying the node

   a. Select **Node query**.
   b. Choose the part instance for which you want to determine nodal coordinates. Click the arrow next to the **Part instance** field to see a list of the available part instances.

      To obtain the coordinates of a node, do one of the following:

      - Type the node label into the **Node label** field, and press **[Enter]** or click **Query**.
      - Click **Select from Viewport** , and select a node.

      Abaqus displays the undeformed *X*-, *Y*-, and *Z*-coordinates of the node.

   c. To enter these coordinates into the path definition table before, after, or at the currently highlighted row, click **Add Before**, **Add After**, or **Replace**, respectively.

      For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.)

Include the coordinates of nodes from a node list path

a. Select **Node list path**.

b. Choose the node path list from which you want to enter the nodal coordinates. Click the arrow next to the **Node list path** field to see a list of the available paths.

c. To enter these coordinates into the path definition table before, after, or at the currently highlighted row, click **Add Before**, **Add After**, or **Replace**, respectively.

On the model plot in the current viewport, Abaqus highlights all points included in the path and displays the path connectivity.

3. When you are done, click **OK**.

If you created a new path, Abaqus/CAE adds the path name to the **Path Manager** list. By default, Abaqus/CAE saves the path you have specified for the duration of the session. If you want to retain a path for use in subsequent sessions, save it to an XML file, to the model database, or to an output database; for more information, see *Managing session objects and session options*.

Additional information

- *Understanding results along a path*
- *Obtaining X–Y data along a path*

Creating or editing an edge list path

A path is a line you define by specifying a series of line segments in your model. In an edge list path the segments are defined by element edges and lines joining the ends of element edges.
For information on including locations other than edges in a path, see *Creating a path through your model*.

You create an edge list path by entering part instance names and edge labels into a table. Each edge label consists of an element number, a face number, an edge number, and a direction along the edge. Select edges from the viewport for entry into the table using a combination of three methods:

- Select individual element edges.
- Select an element edge and a direction along a feature edge and let Abaqus/CAE automatically add the remaining edges up to one end of the feature.
- Select an element edge and a node and let Abaqus/CAE choose a shortest distance path along the element edges between them.

You can also edit the table directly, using the viewport to track your progress. However, some components of the edge labels depend on the element type, orientation, and desired connectivity, which are more easily determined by making selections from the viewport.

1. Locate the path creation and editing options.
   To create a new path:
   a. From the main menu bar, select **Tools->Path->Create**. The **Create Path** dialog box appears.
   b. Abaqus displays a default name for the path in the **Name** text field. To provide a more meaningful name, replace this default with the name of your choice (including blank spaces if you wish).
   c. In the **Create Path** dialog box, click **Edge list**; then click **Continue**.

   To edit an existing path:
   a. From the main menu bar, select **Tools->Path->Edit**. From the menu that appears, select the path you want to edit.

   Tip: You can also use the **Path Manager** to edit a path. From the main menu bar, select **Tools->Path->Manager** to display the manager. Select the path you want to edit, and click **Edit** from the buttons on the right side of the manager.

   The **Edit Edge List Path** dialog box appears.
2. To create or edit a path by selecting edges in the viewport, click **Add Before** or **Add After** to begin making selections.
   Abaqus/CAE prompts you to **Select edges to be inserted into the path** and closes the **Edit Edge List Path** dialog box.
3. Use the selection methods in the prompt area to select edges. Three edge selection methods are available:

Individually

Select individual element edges in the order that they should appear in the path; the selected edges need not be adjacent. If necessary, click **Flip** in the prompt area to reverse the connectivity of the last edge that you selected.

By feature edge

Select a starting edge that lies along a feature edge; Abaqus/CAE will automatically select all the edges between your selection and the end of the feature edge. Click **Flip** to reverse the connectivity of the edge that you selected and the direction used for automatic selection.

By shortest distance

Select a starting edge and an endpoint (node); Abaqus/CAE will automatically select the element edges that define the shortest path between your selections. Click **Flip** to reverse the connectivity of the edge that you selected and the direction used for automatic edge selection. If multiple routes exist with the same distance, Abaqus/CAE selects one of them.



Note:

The edge and node that you select must be on the same part instance.

If a desired edge exists in more than one part instance, toggle off the **Select the Entity Closest to the Screen** tool in the **Selection** toolbar. If you make an ambiguous selection, the part instance name and edge label (without the direction value) will appear in the lower-left corner of the viewport in the form `Highlighting Element Edge Element.Face.Edge`. Use the **Next** and **Previous** buttons to cycle through the possible selections, and click **OK** to confirm your selection.

Abaqus highlights the edges as you select them and displays the node labels and the path connectivity. In addition, Abaqus labels the first node in the path `Start` and the last node in the path `End`.

4. Click **Done** in the prompt area when you have finished.

   Abaqus/CAE displays the **Edit Edge List Path** dialog box; the part instances and edges in your path are listed in the path definition table.

5. If desired, repeat the preceding steps to add additional edges.

   The **Add Before** and **Add After** buttons determine where your selections will be added in relation to the currently highlighted line in the path definition table.

6. If desired, you can use standard keyboard and mouse editing techniques to insert, modify, or delete part instance names and edge labels in the **Path Definition** table. Edge labels are of the form (`element number, face number, edge number, direction`).

   Tip: Use the Query toolset to query nodes in the viewport to obtain part instance names. Display element and face labels in the viewport using the plot state-independent options. (For more information, see *Customizing model labels*.) Edges are numbered counterclockwise around a face as viewed from outside the element. The direction (−1 or 1) depends on your desired path connectivity.

   For special table editing options or to read data from an ASCII file, press mouse button 3. (For more information, see *Entering tabular data*.)

7. After editing the **Path Definition** table, press **[Enter]**.

   Abaqus updates the path displayed in the viewport.

8. When you are done selecting edges, click **OK**.

   If you created a new path, Abaqus/CAE adds the path name to the **Path Manager** list. By default, Abaqus/CAE saves the path you have specified for the duration of the session. If you want to retain a

path for use in subsequent sessions, save it to an XML file, to the model database, or to an output database; for more information, see *Managing session objects and session options*.

Additional information

- *Understanding results along a path*
- *Obtaining X–Y data along a path*

Creating or editing a circular path

A path is a line you define by specifying a series of points through your model. In a circular path, all of the points are model coordinate locations that lie on the circumference or along a radius of a circle. For information on alternative forms of path specification, see *Creating a path through your model*.

You create a circular path by selecting nodes or entering coordinates to define the circle. After you define the circle, you use the circle's edge or a radius to define either a curved circumferential path or a straight radial path.

1. Locate the path creation and editing options.
   To create a new path:
   a. From the main menu bar, select **Tools->Path->Create**. The **Create Path** dialog box appears.
   b. Abaqus displays a default name for the path in the **Name** text field. To provide a more meaningful name, replace this default with the name of your choice (including blank spaces if you wish).
   c. In the **Create Path** dialog box, click **Circular**; then click **Continue**.

   To edit an existing path:

   a. From the main menu bar, select **Tools->Path->Edit**. From the menu that appears, select the path you want to edit.

   Tip: You can also use the **Path Manager** to edit a path. From the main menu bar, select **Tools->Path->Manager** to display the manager. Select the path you want to edit, and click **Edit** from the buttons on the right side of the manager.

   The **Edit Circular Path** dialog box appears.

2. Select the path type, **Circumferential** or **Radial**.

   Abaqus/CAE updates the dialog box to reflect the type of path you are creating. A **Circumferential Path** lies along the edge of the circle and requires a radius, a start angle, and an end angle to define its position and length. A **Radial Path** lies along a radius of the circle and requires a radial angle, a start value, and an end value to define its position and length.

3. In the **Circle Definition** portion of the dialog box, select **Origin and axis** or **3 Points on Arc** as the method to define the circle.

   The origin and axis method uses two points to define the axis of rotation and one point in the plane of the circle to define the default radius length and the starting point of rotation. The three points on arc method fits a circle to three edge points.

4. Click **Select from Viewport** to select nodes or enter coordinates to define the circle.

   Abaqus/CAE closes the **Edit Circular Path** dialog box and displays prompts in the prompt area to guide you through the procedure.

5. Select nodes from the viewport or use the text field in the prompt area to specify each point as *X*-, *Y*-, *Z*-coordinates separated by spaces or commas; for example, 1.0, 3.4, 2.0.

   Abaqus/CAE opens the **Edit Circular Path** dialog box; the coordinates that you selected appear in the circle definition table. If you selected nodes, the coordinates used correspond to the current deformation state of the model.

   On the model plot in the current viewport, Abaqus highlights all points included in the path and displays the path connectivity.

6. You can edit a single point in the **Circle Definition** as follows:
   a. Position the cursor over the coordinates of the point.
   b. Click mouse button 3, and choose **Edit selection**.
   c. Select a node in the viewport, or edit the coordinates in the prompt area.

When you make your selection or enter the new coordinates, Abaqus/CAE returns to the **Edit Circular Path** dialog box and displays the redefined circle in the viewport.

7. Select the number of segments to include in the path.

   Abaqus/CAE modifies the path in the viewport to show the selected number of equal length segments.

8. Complete the editing or creation of the circular path as follows:

Circumferential paths

Select the **Radius**, and select the **Start angle** and **End angle** of rotation. By default, Abaqus uses the radius of the circle that you defined and rotates a full 360° starting at the first point that you selected on the circle—the only point on the circle if you chose the origin and axis method.

Radial paths

Select the **Radial Angle**, **Start Radius**, and **End Radius**. The default radial angle (0°) is from the center to the first point that you selected on the circumference of the circle—the only point on the circle if you chose the origin and axis method. The default starting radius is `0.0` (the center of the circle), and the default end radius is equal to the radius of the circle.

9. When you are done, click **OK**.

   If you created a new path, Abaqus/CAE adds the path name to the **Path Manager** list. By default, Abaqus/CAE saves the path you specified for the duration of the session. If you want to retain a path for use in subsequent sessions, save it to an XML file, to the model database, or to an output database; for more information, see *Managing session objects and session options*.

Additional information

- *Understanding results along a path*
- *Obtaining X–Y data along a path*

Obtaining X–Y data along a path

This section explains how to obtain *X*–*Y* data along a path.

In this section:

- *Choosing the path locations at which to obtain data*
- *Controlling data pair X-values*
- *Controlling data pair Y-values*

Choosing the path locations at which to obtain data

Abaqus/CAE obtains results along a path in the form of *X*–*Y* data pairs. The data pair *X*-values define the model locations at which to obtain data; the *Y*-values are analysis results at those locations.
Abaqus considers only the entities in the current display group when calculating the data pairs.

Abaqus calculates the data pair *X*-values based on the points you have specified for the path. You can choose to form *X*-values based only on the points you have specified or to additionally include all locations at which the path intersects the model. (An intersection occurs where the path crosses an element face, element edge, surface face, or surface edge.)

You can also choose whether Abaqus interprets the points that make up the path as locations on the undeformed or the deformed model shape. For node list paths, the node labels that constitute the path are equally applicable to undeformed and deformed model shapes. However, point list path coordinates remain fixed in space and are independent of your model. For example, coordinates that coincide with a nodal location on the undeformed shape may not coincide with any location on the deformed shape. Values obtained using point list path coordinates and the deformed model are affected by the deformation scale factor. Abaqus does not form data pairs for coordinate locations that do not coincide with your model.

1. Locate the **XY Data from Path** options.

   From the main menu bar, select **Tools->XY Data->Create**. From the dialog box that appears, select **Path**; then click **Continue**. The **XY Data from Path** dialog box appears.

2. Click the **Path** arrow to select the path for which to obtain data.

   The model plot in the current viewport changes to highlight the path you have selected.

3. Click the **Undeformed** or **Deformed** to choose whether Abaqus interprets the points that make up the path as locations on the undeformed or the deformed model shape, respectively.

4. Select **Project onto mesh** and specify a tolerance to obtain data from point locations that do not lie on or inside the mesh.

   Path points or interpolated locations that lie between path points are projected onto an element face in the mesh. If the projected distance is less than the specified tolerance, data are extracted from the projected location. This option has no effect on one-dimensional elements (such as beams) and surface variables (such as CPRESS).

5. From the **Point Locations** options, choose either of the following:

   - Select **Path points** to obtain data only at the points that make up the path. You can obtain *X*–*Y* data at locations where the path intersects the model as well as at the points that make up the path by toggling on **Include intersections**.
   - Select **Uniform spacing** to obtain data at regular, interpolated intervals along the path, but not necessarily at the points that comprise the path definition. You can change the value specified in the **Number of intervals** field to increase or decrease the number of locations for which interpolated data are obtained along the path.

6. To evaluate and display the data, click **Plot**.

   An *X*–*Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.

7. To save the data you have configured, click **Save As**.

   Note:

   To plot your saved *X*–*Y* data object, select **Tools->XY Data->Plot** from the main menu bar and choose the *X*–*Y* data object from the pull-right menu.

8. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding results along a path*
- *Creating a path through your model*
- *Controlling data pair X-values*
- *Controlling data pair Y-values*
- *X–Y plotting*

Controlling data pair *X*-values

Abaqus obtains results along a path in the form of *X*–*Y* data pairs, calculating the data pair *X*-values based on the points you have specified for the path and optionally on additional points where your path intersects the model. For more information, see *Choosing the path locations at which to obtain data*.

The node labels or point coordinates that comprise a path are usually not in a form suitable for direct use as *X*-values. Abaqus offers several options for you to convert this series of points to useful *X*-values and subsequent *X*–*Y* plot axis labels.

1. Locate the **XY Data from Path** options.

   From the main menu bar, select **Tools->XY Data->Create**. From the dialog box that appears, select **Path**; then click **Continue**. The **XY Data from Path** dialog box appears.

2. At the top of the dialog box, click the **Path** arrow to select the path for which to obtain data.

   The model plot in the current viewport changes to highlight the path you have selected.

3. In the **X-Values** field, choose one of the following options to convert path points to *X*-values:

   - **True distance**: *X*-values correspond to each point's actual distance along the path in model space coordinates, starting with zero.
   - **Normalized distance**: *X*-values correspond to each point's distance along the path as a fraction of the total length of the path.
   - **Sequence ID**: *X*-values correspond to the order in which each point occurs in the path result list.
   - **X, Y, or Z distance**: *X*-values correspond to each point's actual distance along the path in the single coordinate direction you specify, starting with zero. This option is particularly useful for generating a plot of results versus radius in an axisymmetric model.
   - **X, Y, or Z coordinate**: *X*-values correspond to each point's distance from the origin along the axis you specify.

4. To evaluate and display the data, click **Plot**.

   An *X*–*Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.

5. To save the *X*–*Y* data pairs you have configured as an *X*–*Y* data object , click **Save As**.

   Note:

   To plot your saved *X*–*Y* data object, select **Tools->XY Data->Plot** from the main menu bar and choose the *X*–*Y* data object from the pull-right menu.

6. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding results along a path*
- *Controlling data pair Y-values*
- *X–Y plotting*

Controlling data pair *Y*-values

Abaqus obtains results along a path in the form of *X*–*Y* data pairs. The data pair *X*-values define the model locations at which to obtain data; the *Y*-values are based on analysis results at those locations. You can control these *Y*-values by choosing the results step, frame, and field output variable for which Abaqus obtains results; by controlling how Abaqus averages certain types of results; and by selecting the display form of complex number results (if applicable).

1. Locate the **XY Data from Path** options.

   From the main menu bar, select **Tools->XY Data->Create**. From the dialog box that appears, select **Path**; then click **Continue**. The **XY Data from Path** dialog box appears.

2. At the top of the dialog box, click the **Path** arrow to select the path for which to obtain data.

   The model plot in the current viewport changes to highlight the path you have selected.

3. The **Y-Values** field identifies the step, frame, and field output variable for which Abaqus will obtain results.

   - To change the step or frame, click **Step/Frame**. For more information, see *Selecting the results step and frame*.
   - To change the field output variable, click **Field Output**. For more information, see *Selecting the primary field output variable*.
   - To remove duplicate XY data pairs, toggle on **Remove duplicate XY pairs**.
   - When extracting results from a node list path, toggle on **Include all elements** to include values from all the elements attached to the selected nodes irrespective of whether it is along the direction of path or not.
   - The averaging of element-based field output values is controlled by the result options settings for the current viewport. To view or change these settings, select **Result->Options** from the main menu bar. For more information, see *Controlling result averaging*.
   - The form of values for complex numbers is controlled by the result options settings for the current viewport. To view or change these settings, select **Result->Options** from the main menu bar. For more information, see *Controlling the form of complex results*.

   Abaqus refreshes the **Y-Values** field and the deformed, contour, or symbol plot in the current viewport according to any changes you apply to the step, frame, field output variable, or result options.

4. To evaluate and display the data, click **Plot**.

   An *X*–*Y* plot appears in the current viewport. The plot represents the data you have configured in the dialog box, which Abaqus considers temporary data whether or not you have clicked **Save As** to save it.

5. To save the *X*–*Y* data pairs you have configured as an *X*–*Y* data object, click **Save As**.

   Note:

   To plot your saved *X*–*Y* data object, select **Tools->XY Data->Plot** from the main menu bar and choose the *X*–*Y* data object from the pull-right menu.

6. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Understanding results along a path*
- *Controlling data pair X-values*

- *X–Y plotting*

Animating plots

An animation is a sequence of images that Abaqus/CAE displays in rapid succession, resulting in a movie-like effect.

In this section:

- *Understanding animation*
- *Object-based animation*
- *Saving an animation file*
- *Controlling animation playback*

Understanding animation

Abaqus/CAE offers two means of animation: object-based and image-based.

Object-based animation

Object-based animation is the display of a sequence of deformed shape plots, contour plots, symbol plots, or material orientation plots. Abaqus/CAE can produce three different types of sequences of these plots. These three sequence types are called “time history animation,”“scale factor animation,” and “harmonic animation.”

Time history animation produces a sequence of deformed shape, contour, symbol, or material orientation plots that vary over time according to actual analysis results. Scale factor animation produces a sequence of images that vary only in the scaling of a single deformed shape, contour, or symbol plot. Harmonic animation produces a sequence of images that represents complex result values varying according to applied angles. To better understand these three types of animation, see *Time history animation*; *Scale factor animation*; and *Harmonic animation*; respectively.

While an object-based animation is playing, you can dynamically change display characteristics such as the view, any viewport annotations, and plot state-dependent customization options.

Image-based animation

Image-based animation is the playback of an animation file. You create an animation file in Abaqus/CAE by first playing object-based animations in one or more viewports and then selecting **Animate->Save As** from the main menu bar. Once saved, your animation can be played external to Abaqus/CAE using industry-standard animation software. You can choose to save your image-based animation file in various formats, such as MP4 or QuickTime.

From within Abaqus/CAE you can also display an image-based animation by selecting the animation file as a background movie. When active, the background movie is displayed in a viewport in the Visualization module. For more information, see *Displaying and customizing a background movie*.



Note: Abaqus/CAE also enables you to save an animation in VRML format, which creates a three-dimensional rendition of the animation. Because these files are three-dimensional, they are not strictly image-based; however, you can play and distribute animation files in VRML format as you would files in another format like QuickTime.

In general, animation playback from a file provides better performance than object-based animation, particularly for large models. While an image-based animation is playing, you cannot change its display characteristics.

In this section:

- *Time history animation*
- *Scale factor animation*
- *Harmonic animation*

Time history animation

In a time history animation Abaqus/CAE creates the images in the sequence by redrawing the deformed shape, contour, symbol, or material orientation plot for every specified frame found in the output database for the active steps and frames. Each image in a time history animation displays actual analysis results. The deformation scale factor remains constant for each image in a time history animation. Result frames can be drawn sequentially or selected from the active frames based on time intervals.

If you are animating a contour plot, the display of the field output values depends on options you select. The default behavior is for the values within the legend to be fixed based on the minimum and maximum values for the first and last animation frame. To learn how to set contour limits, see *Setting contour limits*.

Additional information

- *Customizing the display of steps and frames in the results*
- *Producing a time history animation*
- *Activating and deactivating steps and frames*

Scale factor animation

In a scale factor animation Abaqus/CAE creates all of the deformed shape, contour, or symbol plot images in the sequence from a single step and frame of the output database. Abaqus/CAE generates animation scale factors ranging from 0 to 1 or from −1 to 1, according to your specification. Individual images are formed by applying the range of animation scale factors to the field output values displayed in the plot. This gives the appearance of the field output evolving over time, although all images are produced from a single step and frame of your results. (To view the actual evolution of results over time, you must produce a time history animation.)

If you are animating a contour plot, the display of the field output values depends on options you select. The default behavior is for the values within the legend to be fixed based on the scaled minimum and maximum values for the selected results step and frame. To learn how to set contour limits, see *Setting contour limits*. To learn how to customize the legend, see *Customizing the legend*.

Additional information

- *Producing a scale factor animation*
- *Customizing a scale factor or harmonic animation*

Harmonic animation

Harmonic animation is valid only for field output containing complex numbers. In a harmonic animation Abaqus/CAE creates all of the deformed shape, contour, or symbol plot images in the sequence from a single step and frame of the output database. Abaqus/CAE generates angles ranging from 0 to 180° or from −180° to 180°, according to your specification, and displays the value of the complex field output at each of these angles. This gives the appearance of the field output evolving over time, although all images are produced from a single step and frame of your results. (To view the actual evolution of results over time, you must produce a time history animation.) Harmonic animation is useful for simulating the vibration modes computed by an eigenvalue analysis.

If you are animating a contour plot, the display of the field output values depends on options you select. The default behavior is for the values within the legend to be fixed based on the range of the real component. To learn how to set contour limits, see *Setting contour limits*. To learn how to customize the legend, see *Customizing the legend*. To learn more about complex results, see *Understanding complex results*.

Additional information

• *Producing a harmonic animation*

Object-based animation

This section explains how to produce and customize Abaqus/CAE object-based time history, scale factor, and harmonic animations.

In this section:

- *Producing and customizing an object-based animation*
- *Producing a time history animation*
- *Producing a scale factor animation*
- *Producing a harmonic animation*
- *Customizing a scale factor or harmonic animation*
- *Customizing the underlying plot of an animation*

Producing and customizing an object-based animation

In a time history animation you can customize the analysis steps and frames that Abaqus/CAE includes in your animation. In a scale factor or a harmonic animation you can customize the apparent motion of your model.
For all three types of animation you can customize the underlying deformed shape, contour, symbol, or material orientation plot.

At any time before or during object-based animation playback you can customize the deformed shape, contour, symbol, or material orientation plot that underlies your animation. For example, you can toggle the legend, adjust the deformation scale factor, or choose a different field output variable. Use the **View**, **Result**, and **Options** menus to customize the underlying plot of your animation.

Additional information

- *Producing a time history animation*
- *Producing a scale factor animation*
- *Producing a harmonic animation*

Producing a time history animation

Animating by time history produces a sequence of plots that vary over time according to actual analysis results. You can choose to animate a deformed shape, contour, symbol, or material orientation plot.

1. Use the **File** menu to open the *output database* containing your analysis results.
2. To plot variables other than those Abaqus selects by default, use the **Result** menu.
   - For animation of a deformed or material orientation plot, select the *deformed field output variable* for which Abaqus obtains values over time.
   - For animation of a contour or symbol plot, select the *primary field output variable* or *symbol field output variable* to display. If you want to display your animating contour or symbol plot on the deformed shape, select the *deformed field variable*. For more information on field output variable selection, see *Selecting field output variables*.
3. To customize the steps and frames that Abaqus/CAE includes in the animation, locate the **Active Steps/Frames** dialog box.

   From the main menu bar, select **Result->Active Steps/Frames**. For more information about activating steps and frames, see *Customizing the display of steps and frames in the results*.
4. Use the **Options** and **View** menus to select the *customization options* that you want. These may include deformed, contour, symbol, orientation, and view options.
5. To initiate the animation, select **Animate->Time History** from the main menu bar.

    **Tip:** You can also produce a time history animation using the  tool in the toolbox.

   Animation begins in all animated viewports, and the animation controls appear in the right side of the context bar. Abaqus is now in the time history animation plot state. In this state Abaqus automatically refreshes your plot each time you click **Apply** or **OK** in the **Results**, **Options**, or **View** menu dialog boxes. All animated viewports remain in the time history animation state until you produce a plot in some other state, such as the undeformed or deformed state. To learn how to control the animation, see *Controlling animation playback*.

Additional information

- *Customizing the display of steps and frames in the results*
- *Controlling animation playback*

Producing a scale factor animation

Animating by scale factor produces a sequence of images created from a single step and frame of the output database. These images can consist of either deformed shape, contour, or symbol plots. For deformed shape plots and for contour and symbol plots on the deformed shape, the images vary in the scale factors applied to the deformations. For contour and symbol plots, the images also vary in the scale factors applied to the field output.

1. Use the **File** menu to *open the output database* containing your analysis results.
2. Use the **Result** menu to select the analysis results to be plotted, including the step and frame, the deformed field output variable, and, for contour or symbol plot animation, the *primary field output variable* or *symbol field output variable* to be plotted. For more information on field output variable selection, see *Selecting field output variables*.
3. Use the **Options** and **View** menus to select the *customization options* that you want.
4. To initiate the animation, select **Animate->Scale Factor** from the main menu bar.

    **Tip:** You can also produce a scale factor animation using the  tool in the toolbox.

   Animation begins in all animated viewports, and the animation controls appear in the right side of the context bar. Abaqus is now in the scale factor animation plot state. In this state Abaqus automatically refreshes your plot each time you click **Apply** or **OK** in the **Results**, **Options**, or **View** menu dialog boxes. All animated viewports will remain in the scale factor animation state until you produce a plot in some other state, such as the undeformed or deformed state. For more information on customizing the underlying model shape, see *Customizing model shape*.

Additional information

- *Scale factor animation*
- *Controlling animation playback*
- *Customizing a scale factor or harmonic animation*

Producing a harmonic animation

Harmonic animation produces a sequence of images created from a single step and frame of the output database. These images can consist of either deformed shape, contour, or symbol plots. For deformed shape plots and for contour and symbol plots on the deformed shape, the images vary in the angle applied to the value of the deformations. For contour and symbol plots, the images vary in the angle applied to the field output; both the real and the imaginary components of the results are used.

By default, Abaqus/CAE displays a frame counter during harmonic animation. The frame counter indicates the current angle when complex results are viewed.

1. Use the **File** menu to *open the output database* containing your analysis results.
2. Use the **Result** menu to select the analysis results to be plotted, including the step and frame, the deformed field output variable, and, for contour or symbol plot animation, the *primary field output variable* or *symbol field output variable* to be plotted. For more information on field output variable selection, see *Selecting field output variables*.
3. Use the **Options** and **View** menus to select the *customization options* that you want.
4. To initiate the animation, select **Animate->Harmonic** from the main menu bar.

    **Tip:** You can also produce a harmonic animation using the  tool in the toolbox.

   Animation begins in all animated viewports, and the animation controls appear in the prompt area. Abaqus is now in the harmonic animation plot state. In this state Abaqus automatically refreshes your plot each time you click **Apply** or **OK** in the **Results**, **Options**, or **View** menu dialog boxes. All animated viewports will remain in the harmonic animation state until you produce a plot in some other (for example, undeformed, deformed) state. For more information on customizing the underlying model shape, see *Customizing model shape*.

Additional information

- *Harmonic animation*
- *Controlling animation playback*
- *Customizing a scale factor or harmonic animation*

Customizing a scale factor or harmonic animation

You can customize the apparent motion of your animated model by choosing between the scale factor range **Half cycle** and the extended range **Full cycle**. The extended range is useful, for instance, to visualize a vibration mode obtained by modal analysis. You can also vary the number of frames (separate images) in the animation. A greater number of frames produces a smoother animation.

In a scale factor animation Abaqus/CAE divides the scale factor range by the number of frames to compute the animation scale factor values. These values are applied to the *deformation scale factors* in each coordinate direction and to the *field output variable values* to produce each image in the animation sequence.

In a harmonic animation Abaqus/CAE divides the angle range by the number of frames to compute the animation angle values. These values are applied to compute the value of the field output variable at that angle and to produce each image in the animation sequence.

1. Locate the **Scale Factor/Harmonic** options.

   From the main menu bar, select **Options->Animation**; then click the **Scale Factor/Harmonic** tab in the dialog box that appears. The **Scale Factor/Harmonic** options appear.

2. To control the range, click the **Relative Scaling** that you want:

   - **Full cycle**: for scale factor animation (real numbers) this selection corresponds to the range −1 to 1; for harmonic animation (complex numbers having real and imaginary components) this selection corresponds to the range −180° to 180°.
   - **Half cycle**: for scale factor animation (real numbers) this selection corresponds to the range 0 to 1; for harmonic animation (complex numbers having real and imaginary components) this selection corresponds to the range 0 to 180°.

3. To choose the number of separate images in your animation, click the **Frames** arrow.

   The specified number of frames appears in the **Frames** box.

4. Click **Apply** to implement your changes.

   The scale factor or harmonic animation in all animated viewports changes to reflect the relative scaling and the number of frames you have specified.

   Your changes are saved for the duration of the session and will affect all subsequent plots in the harmonic or scale factor animation state.

Additional information

- *Scale factor animation*
- *Producing a scale factor animation*
- *Harmonic animation*
- *Producing a harmonic animation*
- *Customizing the underlying plot of an animation*
- *Scaling deformations*

Customizing the underlying plot of an animation

Time history, scale factor, and harmonic animations all display a series of deformed shape, contour, symbol, or (for time history animations only) material orientation plots in rapid succession. At any time before or during animation playback you can customize the deformed shape, contour, symbol, or material orientation plot that underlies your animation. For example, you can toggle the legend, adjust the deformation scale factor, or choose a different field output variable. Use the **View**, **Result**, and **Options** menus to customize the underlying plot of your animation.

1. Locate the customization options applicable to the current plot state.
   - For contour plot animation, select **Options->Contour** from the main menu bar.
   - For symbol plot animation, select **Options->Symbol** from the main menu bar.
   - For material orientation plot animation, select **Options->Material Orientation** from the main menu bar.
2. From the dialog box that appears, select the plot state–dependent customization options that you want. For more information, see *Customizing your plots*.
3. Set the plot state–independent customization options that you want by selecting the **Options->Common**, **Options->Superimpose**, **Viewport->Viewport Annotation Options**, **View->Graphics Options**, or **View->ODB Display Options**.
4. Choose the *results variables* to be animated by selecting **Result->Field Output**.
5. Select the results steps and frames to be animated:
   - For scale factor or harmonic animation, choose the single step and frame to be animated by selecting **Result->Step/Frame**.
   - For time history animation, choose the steps and frames to be animated by selecting **Result->Active Steps/Frames**.

Saving an animation file

You can save animations playing in one or more viewports to a file. Once saved, your animation can be played using industry-standard animation software. This section explains how to save animations to a file and how to control the format of that file.

In this section:

- *Saving animations*
- *Choosing the animation file format*

Saving animations

You save animations to a file by first playing animations in one or more viewports and then selecting **Animate->Save As** from the main menu bar.

You can choose to save all or just selected viewports. Viewports that you save can be in any plot state (for example, deformed, *X–Y*, or animation). For viewports currently in the animation plot state Abaqus/CAE saves all frames of the animation, starting with the first frame; it does not matter whether the animation is playing at the time you perform the save. If you save multiple viewports containing animations, Abaqus/CAE saves them all as a single image-based animation. Abaqus/CAE synchronizes multiple animations of the file by repeating the last frame of any animations that have fewer frames than the others.

1. Produce one or more time history, scale factor, or harmonic animations.

   For more information, see *Object-based animation*.

2. From the main menu bar, select **Animate->Save As**.

   The **Save Image Animation** dialog box appears.

3. For MP4, AVI, QuickTime, or GIF format, use the **Selection** field to choose the viewports to save:

   a. Click the arrow next to the **Capture** field to save **All Viewports** (the default) or the **Current Viewport**.

   b. Toggle **Capture viewport decorations** to control whether Abaqus/CAE saves visible viewport decorations.

   For VRML or Compressed VRML format, the saved animation will display the current viewport only and omit viewport decorations.

4. Use the **Settings** field to specify the name and format of your animation file:

   a. Click  to filter and browse file names, or enter the file name of your choice in the **File name** field. For more information, see *Using file selection dialog boxes*.

   b. From the **Format** options, select **MP4** (the default), **AVI**, **QuickTime**, **GIF**, **VRML**, or **Compressed VRML** format for your file. To learn more about these format options, see *Choosing the animation file format*.

5. For MP4, AVI, QuickTime, or GIF format, use the **Frame Rate** field to specify the number of frames that display per second in the saved animation.

   You can specify any positive integer as the frame rate by entering a value in the **Rate** field or drag the slider to select a frame rate between 1 and 50 frames/second. By default, the frame rate matches the frame rate currently selected on the **Player** page.

6. Click **Apply** to save your animation.

   Abaqus/CAE captures canvas viewports to a file according to your specifications.

Additional information

• *Saving an animation file*

Choosing the animation file format

You can choose to save animations to a file in MP4 (the default), AVI, QuickTime (MOV), animated GIF, or Virtual Reality Modeling Language (VRML) format.
You can choose between compressed and uncompressed files in VRML, AVI, and QuickTime formats. Additional image size and compression options are available for MP4, AVI, and QuickTime formats.

When you save animations in MP4 format, you can compress the file using the H.264 codec if an installation of the FFmpeg program that supports H.264 encoding can be found on your system. Otherwise, the file is encoded using the MPEG-4 (Part 2) codec. For H.264 encoding, the FFmpeg program location should be included in the PATH environment where Abaqus/CAE is launched. For more information about FFmpeg, visit *https://ffmpeg.org*.

When you save animations in AVI format on Windows systems, you can compress the file using any codec on your system and you can set the quality level of the file. These options are not available on Linux systems.

Improving playback of VRML animations

Animations in VRML or Compressed VRML format might require further customizations to play back properly on client computers. Several components on the client's computer determine the quality of VRML playback, including the graphics hardware, VRML plug-in software, and the settings selected for the plug-in software. When these components differ from the components on the computer that created the VRML animation, VRML playback on the client's computer can exhibit display problems such as blurry text in the viewport annotations. You can improve client playback of a particular VRML file either by configuring plug-in settings on the client's computer or by creating a new version of the VRML animation with different viewport annotation settings.

- To customize VRML display on the client's computer, open the settings for the VRML plug-in on the client's computer, and toggle on any antialiasing settings. Enabling antialiasing can help clarify text in a VRML image or animation but can degrade performance.
- To create a new VRML animation, return to Abaqus/CAE and increase the text size or choose a different font for each of the viewport annotations in the viewport and save a new copy of the VRML animation. For more information about changing the font and type size, see *Customizing viewport annotations*.

Additional information

• *Saving animations*

Choose VRML or Compressed VRML format

1. Locate the animation file format options.
   From the main menu bar, select **Animate->Save As**; the **Save Image Animation** dialog box appears.
2. From the **Format** options, select **VRML**, or **Compressed VRML** format.
   For both VRML formats, Abaqus/CAE deactivates the **Selection** and **Frame Rate** fields; you cannot customize these settings for **VRML** or **Compressed VRML** animations.

Choose MP4 format

1. Locate the animation file format options.
   From the main menu bar, select **Animate->Save As**; the **Save Image Animation** dialog box appears.
2. From the **Format** options, select **MP4**.
3. To select additional MP4 format options, click



.

The **MP4 Options** dialog box appears.

4. In **MP4 Options** dialog box, choose one of the following methods to specify the size (in pixels) of the animation image:

   - Click **Use size on screen** to use the size of the image on the screen. (Abaqus/CAE indicates the current image size in the **MP4 Options** dialog box.) This method is the default.
   - Click **Use settings below** to set the width or height; then enter the value of your choice in the **Width** or **Height** field. You specify only one dimension; Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports.

5. Specify in the **MP4 Options** dialog box your preferred encoding.
   a. From the **Codec** list, select either **H.264 / AVC** or **MPEG-4 Part 2**. **H.264 / AVC** is widely supported by video editing and playback software, and offers generally a superior compression versus quality tradeoff, but it is available only if an FFmpeg installation with H.264 encoding enabled can be found on your system, as described above.
   b. Drag the **Quality** slider to specify the compression quality of the file. The highest value, 100, produces the highest quality animation file with the smallest degree of compression.
6. Click **OK** to close the **MP4 Options** dialog box.

   Your options settings are saved for the duration of the session.

Choose AVI format

1. Locate the animation file format options.

   From the main menu bar, select **Animate->Save As**; the **Save Image Animation** dialog box appears.
2. From the **Format** options, select **AVI**.
3. To select additional AVI format options, click



.

The **AVI Options** dialog box appears.

4. In the **AVI Options** dialog box, choose one of the following methods to specify the size (in pixels) of the animation image:

   - Click **Use size on screen** to use the size of the image on the screen. (Abaqus/CAE indicates the current image size in the options dialog box.) This method is the default.
   - Click **Use settings below** to set the width or height; then enter the value of your choice in the **Width** or **Height** field. You specify only one dimension; Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports.

5. For Linux systems, specify in the **AVI Options** dialog box whether your file should be compressed (run length encoded) or uncompressed (raw).

   From the **Compression** list in the **AVI Options** dialog box, select **Rle (8 bits/pixel)**, **Raw (8 bits/pixel)**, or **Raw (32 bits/pixel)**. Choosing **Rle (8 bits/pixel)** compression or **Raw (8 bits/pixel)** can result in a loss of colors.

6. For Windows systems, specify in the **AVI Options** dialog box whether your file should be compressed or uncompressed (raw). If you elect to compress the file, you can select one of the Abaqus/CAE predefined compressors or one of the video codecs available on your system. You can specify the compression quality of the saved animation file.
   a. From the **Codec** list, select one of the following:
      - Select **None - 8 bits/pixel** to create a raw animation using a reduced set of colors. Choosing this option can result in a loss of colors.
      - Select **None - 24 bits/pixel** to create a raw animation file without any compression.
      - Select one of the video codecs. The **Codec** list displays every codec that is available on your system.

      Some video codecs include configurable options that enable you to change certain aspects of the video file. When you select a codec that allows this configuration, the **Configure** button becomes active. You can also learn more about a particular codec by clicking the **About** button when the codec is selected.
   b. To configure a codec, click **Configure** and then choose configuration options from the dialog box that appears. Each configurable video codec provides different configuration options; for example, some codecs allow you to control the transparency of the animation or to create a scalable file.

      Note: You might encounter problems while saving an animation in AVI format with MPEG-4 codec compression using a Python script. To avoid these problems, you can use the GUI described in this procedure or you can configure the Xvid codec to switch off the encoding status window (**Xvid configuration->Other options->Encoder**: toggle off **Display encoding status**).

   c. Drag the **Quality** slider to specify the compression quality of the file. The highest value, 100, produces the highest quality animation file with the smallest degree of compression.
7. Click **OK** to close the **AVI Options** dialog box.
   Your options settings are saved for the duration of the session.

Choose QuickTime format

1. Locate the animation file format options.
   From the main menu bar, select **Animate->Save As**; the **Save Image Animation** dialog box appears.
2. From the **Format** options, select **QuickTime**.
3. To select additional QuickTime format options, click

   .
   The **QuickTime Options** dialog box appears.
4. In **QuickTime Options** dialog box, choose one of the following methods to specify the size (in pixels) of the animation image:
   - Click **Use size on screen** to use the size of the image on the screen. (Abaqus/CAE indicates the current image size in the **QuickTime Options** dialog box.) This method is the default.

- Click **Use settings below** to set the width or height; then enter the value of your choice in the **Width** or **Height** field. You specify only one dimension; Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports.

5. Specify in the **QuickTime Options** dialog box whether your file should be compressed (run length encoded) or uncompressed (raw).

   From the **Compression** list, select either **Raw (24 bits/pixel)** or **Rle (24 bits/pixel)**. **Rle (24 bits/pixel)** compression is a form of run length encoding, which provides lossless compression with no image degradation.

6. Click **OK** to close the **QuickTime Options** dialog box.

   Your options settings are saved for the duration of the session.

Choose GIF format

1. Locate the animation file format options.

   From the main menu bar, select **Animate->Save As**; the **Save Image Animation** dialog box appears.

2. From the **Format** options, select **GIF**.
3. To select additional GIF format options, click

   .

   The **GIF Options** dialog box appears.

4. In **GIF Options** dialog box, choose one of the following methods to specify the size (in pixels) of the animation image:

   - Click **Use size on screen** to use the size of the image on the screen. (Abaqus/CAE indicates the current image size in the **GIF Options** dialog box.) This method is the default.
   - Click **Use settings below** to set the width or height; then enter the value of your choice in the **Width** or **Height** field. You specify only one dimension; Abaqus/CAE computes the other dimension to maintain the aspect ratio of the viewports.

5. Click **OK** to close the **GIF Options** dialog box.

   Your options settings are saved for the duration of the session.

Controlling animation playback

During an object-based animation you can control the speed and repetition of playback, and you can stop, restart, and step through your animation.

Most industry-standard animation software supports the same functions for image-based animation, but the available options are based on the application that you choose. For object-based animations one additional option is available: you can display or suppress status information while the animation is playing.

If you are using multiple viewports to display animations, you can include or exclude individual viewports from the common animation time line. By default, each new viewport you create will be included in the synchronization.

This section explains how to control animation playback.

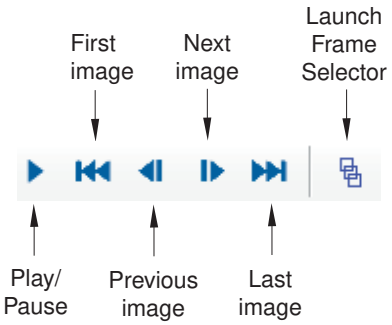
In this section:

- *Stopping, restarting, and stepping through an animation*
- *Navigating to a specific frame in the animation*
- *Controlling animation speed and repetition*
- *Showing the animation frame counter*
- *Customizing animation of X–Y plots*
- *Controlling animations in multiple viewports*
- *Customizing the time history of synchronized viewports*

Stopping, restarting, and stepping through an animation

You can use the movie player controls to play and pause your animation, to step forward or backward through the animation images, and to step to the first or last image in the animation sequence.

During animation, the movie player controls appear on the right side of the context bar.



If you display animations in multiple viewports, the movie player controls stop, restart, and step through frames in all synchronized viewports.

You can also use the right-most button in the movie player controls to launch the **Frame Selector** dialog box, which allows you to navigate directly to a particular frame in the animation. This tool is discussed in *Navigating to a specific frame in the animation*.

1. Locate the movie player controls on the right side of the context bar.
2. Click the movie player controls to execute the functions you want. The controls are listed below in the order in which they appear in the context bar, starting with the left-most arrow.

Play/Pause	Plays or pauses. When your animation is playing, this button displays the Pause symbol; when paused, this button displays the Play symbol.
First image	Stops play and displays the first image in the animation sequence.
Previous image	Stops play and steps to the previous image in the animation sequence.
Next image	Stops play and steps to the next image in the animation sequence.
Last image	Stops play and displays the last image in the animation sequence.
Launch Frame Selector	Opens the **Frame Selector** dialog box, from which you can navigate to a specific frame of the animation.

Additional information

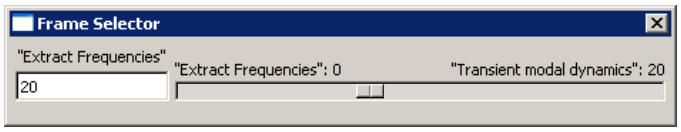
- *Controlling animation playback*
- *Navigating to a specific frame in the animation*
- *Controlling animations in multiple viewports*

Navigating to a specific frame in the animation

The **Frame Selector** dialog box allows you to navigate quickly to a particular frame in the output database in the current viewport. This tool provides two methods for controlling the current frame: you can enter a frame index or time value directly into a field on the left side of the dialog box, or you can drag the slider on the right side of the dialog box to the desired frame or time.

The labels in the dialog box are context-sensitive and depend on whether an animation is being displayed, the type of animation, and for time history animations, whether the time history is frame-based or time-based.

- When no animation is displayed in the current viewport, the **Frame Selector** labels display step names. The name of the current step appears above the field on the left side of the dialog box. The names of the first and last active steps in the output database appear above the slider. In the example below, the current frame is frame 20 of the **Extract frequencies** step, the first step is **Extract frequencies**, and the last step is **Transient modal dynamics**.



- For scale factor or harmonic animations the **Frame Selector** labels displays the frame index of the current frame, and the slider shows the frame index of the first and last active frames. The dialog box displays **Frame** above the field and the numbers of the first and last active frames above the slider.
- For time history animations the **Frame Selector** labels display frame information when you select a frame-based time history and display time information when you select a time-based time history. In the latter case the dialog box displays **Time** above the field and the time values of the first and last active frames above the slider. For more information about selecting a time history option, see *Customizing the time history of synchronized viewports*.

1. Open the **Frame Selector** dialog box.

   From the movie player controls on the right side of the context bar, click the button.

   The **Frame Selector** dialog box appears.

2. Choose one of the following methods to specify the frame you want to display:

   - Enter a number directly into the **Frame Selector** field.

     You can enter the overall frame number in the entire output database for scale factor animations, harmonic animations, or frame-based time history animations; you can enter the time for time-based time history animations; or you can enter the frame number within the current step when no animation is displayed.

   - Drag the slider to the frame index or time value that you want to display. The current viewport changes as you drag the slider.

Additional information

- *Controlling animation playback*
- *Stopping, restarting, and stepping through an animation*

Controlling animation speed and repetition

You can control the speed of your object-based animation playback, and you can choose to play your animation once or to repeat it.

If you choose to repeat the animation, you can select one of three repetition modes: loop, loop backward, or swing. **Loop** repeats by jumping from the last animation frame to the beginning of your movie. **Loop backward** plays the animation backward and repeats by jumping from the first animation frame to the end of your movie. **Swing** repeats smoothly by playing your movie from beginning to end and then playing from the end back to the beginning.

The speed control settings you specify before you create image-based animations determine the default frame rate that is used when the animation files (for example, MP4 or QuickTime) are replayed from within industry-standard animation software.

1. In the prompt area, click **Animation Options**; and click the **Player** tab in the dialog box that appears.
2. To control the playback speed of your animation, drag the **Frame rate** slider to the rate that you want.
3. To control the repetition of the images in your animation sequence:
   - Click **Play once** to suppress repetition.
   - Click **Loop** to play the images in order from the first to the last, repeatedly.
   - Click **Loop backward** to play the images in order from the last to the first, repeatedly.
   - Click **Swing** to play the images from the first to the last and then in reverse order from the last back to the first, repeatedly.
4. Click **Apply** to implement your changes.

   The animation in all animated viewports changes according to your specifications.

   Your changes are saved for the duration of the session and will affect all subsequent plots in this state.

Additional information

• *Stopping, restarting, and stepping through an animation*

Showing the animation frame counter

For object-based animations you can display or suppress animation frame count information, which is displayed in the upper-right corner of the viewport and consists of

- the scale factor of each image in your scale factor animation,
- the phase angle of each image in your harmonic animation, or
- the step and frame (plot number within the step) of each image contained in your time history animation.

You cannot customize the content or appearance of the counter, other than to turn it off entirely.

The step and increment associated with each animation step and frame are given by the state block. All of the images in a scale factor animation are based on a single step and frame. For more information on customizing the state block, see *Customizing the state block*.

1. Find the **Show frame counter** option.

   In the prompt area, click **Animation Options**; and click the **Player** tab in the dialog box that appears. The **Show frame counter** option is near the bottom of the page.

2. Toggle **Show frame counter** to display or suppress animation frame count information.
3. Click **Apply** to implement your changes.

   The animation in all animated viewports changes according to your specifications. When **Show frame counter** is on, Abaqus/CAE displays status information in the upper right corner of all animated viewports. This information is updated for every image in your animation sequence. You may need to *stop* playback to read it.

   Your changes are saved for the duration of the session and will affect all subsequent plots in this state.

Additional information

• *Stopping, restarting, and stepping through an animation*

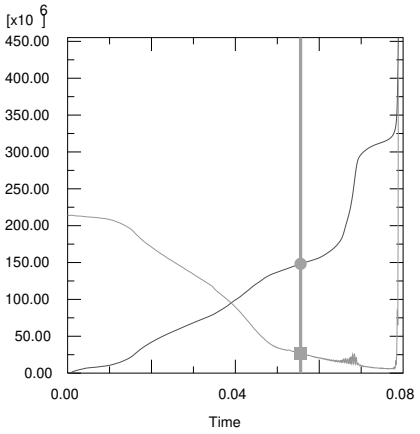
Customizing animation of X–Y plots

You can customize the appearance of the time tracker line and symbols.

When you animate a time-dependent *X–Y* plot, Abaqus/CAE adds a vertical line to the plot that moves along the *X*-axis to highlight the current time. The plot also displays a colored symbol at the data point on each *X–Y* curve that occurred closest to the current time; time tracker symbols do not appear at the exact intersection of the time tracker line and an *X–Y* curve unless that curve contains a data point at the currently displayed time.

For *X–Y* plots whose axes are not time but are obtained from output database results (for example, by combining two time-dependent *X–Y* data objects), Abaqus/CAE also adds a vertical line. In such cases, the line is added if there is only one curve in the plot.

The figure below shows an *X–Y* plot with the time tracker line and symbols displayed.



By default, the time tracker line and symbols are displayed using the highlight method that you have selected for your session; see *Choosing a highlight method* for more information. However, you can customize their appearance by changing the color, thickness, and style of the time tracker line and the color, shape, and size of the time tracker symbols. Alternatively, you can completely suppress either time tracker component. The default highlight method offers the best performance.

Note:

These line and symbol options control the appearance of the time tracker only. You can also customize the lines and symbols used to display the *X–Y* curves themselves; see *Customizing the appearance of an X–Y curve*.

1. Locate the **XY** options.

   In the prompt area, click **Animation Options**; and click the **XY** tab in the dialog box that appears.
2. Toggle off **Draw using highlight method** if you want to customize the appearance of the *X–Y* plot time tracker.
3. Toggle **Show line** to display or suppress the time tracker line.

   When **Show line** is on, the time tracker line is displayed and the line attribute options are enabled.
4. Choose the time tracker line style:
   a. Click the **Style** button to reveal the line style options (solid, dashed, etc.).
   b. From the style list, click the desired line style.

The specified line style appears on the **Style** button.

5. Choose the time tracker line thickness:
   a. Click the **Thickness** button to reveal the line thickness options.
   b. From the thickness list, click the desired line thickness.

      The specified thickness appears on the **Thickness** button.

6. Choose the time tracker line color:
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

7. Toggle **Show symbol** to display or suppress the symbols representing the current value of each selected *X–Y* curve's data points.

   When **Show symbol** is on, symbols are displayed and the symbol attributes are enabled.

8. Choose the symbol:
   a. Click the **Symbol** button to reveal the choice of symbols.
   b. From the list of symbols, click the desired symbol.

      The specified symbol appears on the **Symbol** button.

9. Choose the time tracker symbol color:
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

10. Choose the time tracker symbol size:
    a. Click the **Size** button to reveal the symbol size options.
    b. From the symbol size options, click the desired symbol size.

       The specified symbol size appears on the **Size** button.

11. Click **Apply**.

    The current viewport displays your color, thickness, style, and symbol selections.

Additional information

• *Customizing the time history of synchronized viewports*

Controlling animations in multiple viewports

By default, Abaqus/CAE animates data in every animation-eligible viewport in your session. All animations are synchronized, so the viewports play, stop, and increment together when you examine data frame by frame, animate a model plot, or animate a time-dependent *X*–*Y* plot.

You can activate and deactivate the animation of individual viewports, which enables you to display animations only for a subset of the animation-eligible viewports in your session. From the **Linked Viewports Manager**, toggle on **Link viewports** and **Animation** and use the **Linked Viewports** portion of the **Linked Viewports Manager** to deselect the viewports that you want to remain independent.

1. Link viewports using one of the following methods:
   - From the main menu, select **Viewport->Link Viewports**.

     Tip: You can activate or deactivate linked viewports by clicking  in the *Viewport toolbar*.

   - From the main menu, select **Viewport->Linked Viewports Manager** and toggle on **Link viewports**.

     Tip: You can also display the **Linked Viewports Manager** by clicking  in the *Viewport toolbar*.

2. In the **Linked Viewports Manager**, ensure that **Animation** is toggled on and the required viewports are selected in the **Linked Viewports** table.
3. If you made changes in the **Linked Viewports Manager**, click **Apply**. Alternatively, click **OK** to apply your changes and to close the manager.
4. Start a time history, scale factor, and harmonic animation using any method.

Additional information

• *Customizing the time history of synchronized viewports*

Customizing the time history of synchronized viewports

The time history settings control whether Abaqus/CAE uses a frame-based history or a time-based history when you display time history animations. Frame-based synchronization allows all synchronized viewports to increment together one frame at a time, while time-based synchronization allows viewports to display data according to a common time line.

When you choose time-based animation, you can also customize three aspects of the common time line:

- The time step, which controls how much time elapses each time you increment your display of the data.
- The minimum time, which is the earliest point from which data will be included. If you auto-compute the minimum time, Abaqus/CAE includes data starting from the beginning of the common time line.
- The maximum time, which is the latest point from which data will be included. If you auto-compute the maximum time, Abaqus/CAE includes data up to the end of the common time line.

1. Find the **Time History** options.

   In the prompt area, click **Animation Options**; and click the **Time History** tab in the dialog box that appears.
2. From the options at the top of the page, choose either **Frame-based** or **Time-based**.
3. If desired, change the **Time increment** value. The time increment applies to time-based animation only.
4. If desired, customize the **Min time** and **Max time** by entering values in the appropriate **Specify** fields. These values apply to time-based animation only.
5. Click **Apply**.

Additional information

- *Changing the period of a step*
- *Controlling animations in multiple viewports*

Querying the model in the Visualization module

This section explains how you can use the Query toolset to obtain general information about the mesh and specific Visualization module information.

In this section:

- *Understanding the Query toolset in the Visualization module*
- *Using the Query toolset*

Understanding the Query toolset in the Visualization module

The Query toolset allows you to obtain general information about your model.
The Visualization module displays the requested information in the message area, and the same information is written to the replay file.

Select **Tools->Query** from the main menu bar, or select the  tool from the Query toolset to use the Query toolset.

Items under **General Queries** provide the following information:

Node

You can obtain information on a selected node's label, deformed and undeformed coordinates, and displacement. You can select a local CSYS in which to display the coordinates.

Distance

You can obtain information on the deformed and undeformed distance between two selected nodes and the relative displacement between the nodes. You can select a local CSYS in which to display the coordinates of the picked points and the components of the result.

Angle

- You can obtain information on the deformed and undeformed angle formed between three selected nodes. The second node that you select is the angle's vertex.
- You can obtain the angle between two edges or faces or between an edge and a face.

Element

You can obtain information on a selected element's mesh type, material, section, connectivity, and current field output variables at the integration point locations. This query is available only when an output database is selected.

Mesh

You can obtain information on the name of the current output database, the number of nodes and elements in your model, and the element types.

Mass properties

You can obtain basic mass properties for the whole model in the output database or for a portion of the model. In the Visualization module, the mass properties are always calculated based on model data (undeformed shape). This query is available only when an output database is selected. Abaqus/CAE returns the following information:

- Volume
- Volume centroid
- Mass
- Center of mass
- Moments of inertia about the center of mass or about a specified point

Element face normal

You can obtain the surface normal components for an element face. You can select a local CSYS in which to display the normal.

Advanced distance

Whereas the regular distance query allows you to find the distance between two selected single entities, the advanced distance query allows you to find the minimum distance between groups of entities. These entities can be nodes, elements, vertices, edges, faces, or cells, and you can select them from the viewport or by set. You can select a local CSYS in which to display the coordinates of the closest points and the components of the result.

For more information on general model queries, see *Querying the model in the Visualization module*.

Items under **Visualization Module Queries** provide the following information:

Probe values

Abaqus/CAE displays information in the **Probe Values** dialog box as you move the cursor around the current viewport. Probing a model plot displays model data and analysis results; probing an *X–Y* plot displays *X–Y* curve data. For more information on probing, see *Probing the model*.

Stress linearization

Stress linearization is the separation of stresses through a section into constant membrane and linear bending stresses. Abaqus performs stress linearization calculations and displays the results in the form of an *X–Y* plot. Stress linearization is available for output databases only. For more information on stress linearization, see *Calculating linearized stresses*.

Active elements and nodes

Abaqus/CAE displays the label numbers of all the active nodes or active elements in the current viewport. For more information, see *Querying active node or element labels*.

Ply stack plot

Abaqus/CAE creates a new viewport and displays a representative image of a composite layup. The image shows the plies in the layup along with details of each ply, such as its fiber orientation, thickness, reference plane, and integration points. Ply stack plots are available for output databases only. For more information, see *Viewing a ply stack plot*.

- *Probing the model*
- *Calculating linearized stresses*
- *Viewing a ply stack plot*

Using the Query toolset

You can use the Query toolset to obtain general model information.

In this section:

- *Querying the model in the Visualization module*
- *Querying active node or element labels*

Querying the model in the Visualization module

1. Locate the **Query** dialog box.

   From the main menu bar, select **Tools**->**Query** or click the  tool in the Query toolset.
   The **Query** dialog box appears.

2. To obtain information on a particular node, do the following:
   a. Select **Node** from the **General Queries** field of the **Query** dialog box.
      Abaqus/CAE displays a prompt in the prompt area.
   b. Select a node from the viewport.

      The undeformed and deformed *X*-, *Y*-, and *Z*-coordinates of that node appear in the message area, along with the node's displacement. The same information that appears in the message area is also written to the replay file.
   c. Select a local CSYS from the viewport to display the node coordinates in the local coordinate system. If a local CSYS is not selected, the node coordinates will be displayed in the global Cartesian system.

       **Note:**

      To resize the message area, drag the top edge; to see information that has scrolled out of the message area, use the scroll bar on the right side.

3. To obtain information on the distance between two nodes, do the following:
   a. Select **Distance** from the **General Queries** field of the **Query** dialog box.
      Abaqus/CAE displays a prompt in the prompt area.
   b. Select two nodes from the viewport.

      The following information appears in the message area:

      - The undeformed and deformed *X*-, *Y*-, and *Z*-coordinates of each node, along with the node's displacement.
      - The absolute undeformed and deformed distances between the nodes.
      - The *X*-, *Y*-, and *Z*-components of the undeformed and deformed vector between the two nodes.
      - The absolute relative displacement between the nodes.
      - The *X*-, *Y*-, and *Z*-components of the relative displacement between the two nodes.

   c. Select a local CSYS from the viewport to display the coordinates of the picked nodes and the components of the result in the local coordinate system. The last chosen CSYS will be used for the calculations. If any local CSYS is not selected, the results will be displayed in the global Cartesian system.

4. To obtain angle information, select **Angle** from the **General Queries** field of the **Query** dialog box, and select **3 Nodes** or **Face/Edge** from the prompt area.

   - If you selected **3 Nodes**, select three nodes from the viewport. The second node that you select is the vertex of the angle.

     The following information appears in the message area:

- The undeformed and deformed *X*-, *Y*-, and *Z*-coordinates of each node, along with the node's displacement.
- The absolute undeformed and deformed angle between the nodes.

- If you selected **Face/Edge**, select two faces or two edges or select a face and an edge. The angle between the selected faces/edges is displayed in the message area.

5. To obtain information on a particular element in an output database, do the following:

a. Select **Element** from the **General Queries** field of the **Query** dialog box.

Abaqus/CAE displays a prompt in the prompt area.

b. Select an element from the viewport.

The following information appears in the message area:

- The element's label, element type, material, and section.
- The labels of connecting elements.
- The current field output variables at the integration point locations.

6. To obtain general information on the mesh, select **Mesh** from the **General Queries** field of the **Query** dialog box.

The following information appears in the message area:

- The name of the current output database.
- The number of nodes.
- The number of elements.
- The element types.

7. To obtain general information on the mass properties in an output database, do the following:

a. Select **Mass properties** from the **General Queries** field of the **Query** dialog box.

Abaqus/CAE displays a prompt in the prompt area for the last selection method used or **All elements** if this is the first time using the **Mass properties** query in the current session.

b. Choose one of the following selection methods from the list in the prompt area:

- **All elements**
- **Select elements from viewport**
- **Part instances**
- **Element sets**
- **Sections**
- **Materials**
- **Element types**
- **Display groups**

Abaqus/CAE displays additional selection fields in the prompt area, depending on the selection method that you chose.

c. Complete your selection as follows:

- If your selection from the previous step is **All elements**, click **Done** to perform the query.

- If you chose **Select elements from viewport** in the previous step, make selections from the viewport (for more information, see *Selecting objects within the viewport*"). You can select a part instance in the prompt area and enter a single element number to get mass information for that element.
- For the remaining selection methods, select the part instance, element set, section, material, element type, or display group name from the list in the prompt area, or make selections from the viewport.

When you select an object from the prompt area or when you click **Done** to complete a viewport selection, Abaqus/CAE displays the mass properties for your selection in the message area.

d. If any items in your selection have poorly defined density or thickness, click **Options** in the prompt area to display the **Mass Properties Query Options** dialog box and specify a value to be substituted by Abaqus/CAE for the undefined quantity.

Click **OK** in the dialog box to save your entries and close the dialog, or click **Reset** to return to the previous settings.

Abaqus/CAE displays the mass properties in the message area for the items you selected. The following properties are available:

- Surface area (for shell elements only)
- Area centroid
- Volume
- Volume centroid
- Mass
- Center of mass
- Moments of inertia about the center of mass or about a specified point

8. To obtain the surface normal components, do the following:

a. Select **Element face normal** from the **General Queries** field of the **Query** dialog box.

Abaqus/CAE displays a prompt in the prompt area.

b. Select an element face in the viewport, and click **Done**.

The surface normal components in the global Cartesian coordinate system of the element face appear in the message area.

c. Select a local CSYS from the viewport to display the surface normal components in the local coordinate system. If any local CSYS is not selected, the surface normal components will be displayed in the global Cartesian system.

The surface normal components will differ if you query the same element face on the undeformed shape and the deformed shape.

9. Close the **Query** dialog box when you have finished requesting information.

Additional information

- *Understanding the Query toolset in the Visualization module*
- *Querying mass properties*

Querying active node or element labels

You can use the Query toolset to request the label numbers of all currently active nodes or active elements in the current viewport. The active nodes and elements are those that are in the current display group, less any entities removed from the display by the active view cut. Abaqus/CAE displays label numbers in the prompt area for each applicable part instance and prints label numbers to the replay file.

When you request the active nodes or elements, Abaqus/CAE displays the label numbers only; it does not provide any other nodal or element data and results. To display more information about a single node or element, see *Probing nodes*, or *Probing elements*, respectively.

1. Locate the **Visualization Module Queries** options.

   From the main menu bar, select **Tools->Query** or click the  tool in the Query toolset. The **Query** dialog box appears, and the **Visualization Module Queries** are displayed in the lower portion of the dialog box.

2. Click either **Active nodes** or **Active elements**.

   In the prompt area, Abaqus/CAE displays a list of labels for all active nodes or active elements for each part instance in the model.

Additional information

- *Probing nodes*
- *Probing elements*

Probing the model

This chapter explains how you can use the Query toolset in the Visualization module to probe model and *X*–*Y* plots for output data and to probe an Abaqus/CAE model for attribute values.

For information on writing selected probe values to a file, see *Generating tabular data reports*.

In this section:

- *Understanding probing*
- *Using the Query toolset to probe the model*

Understanding probing

This section explains probing model plots and *X*–*Y* plots.

When you click the  tool in the Visualization module toolbox, Abaqus/CAE enters probe mode and displays information in the **Probe Values** dialog box as you move the cursor over the model in the current viewport. You can also enter probe mode using the  tool in the Query toolbar. Probing a model plot displays model data and analysis results; probing an *X*–*Y* plot displays *X*–*Y* curve data; and probing a model from the current model database displays the model data and the values of attributes such as loads and predefined fields. You can write this information to a file.

The data table in the bottom portion of the **Probe Values** dialog box contains two dynamic display rows (the row immediately above the row headings and the bold data row that appears first in the table) and a series of additional storage rows. Abaqus/CAE updates the dynamic rows continuously to show the probe values as you move the cursor over the model in the current viewport, and you can change the value displayed in the top display row by clicking a column heading. Abaqus/CAE updates the table with additional rows only when you select values of interest by clicking in the viewport, specifying a node or element label, or selecting the nodes or elements in a particular display group. Values accumulate in the bottom portion of the dialog box until you delete them, clear the table, or cancel probe mode. You can also add selected values to a new display group, and you can annotate individual nodes or elements with field output values at that location.

Probes of models or model plots

If the current viewport contains a model plot (an undeformed, deformed, contour, symbol, or material orientation plot) or a model from the current model database,, Abaqus/CAE allows you to probe results from the selected model or output database. The **Probe Values** dialog box identifies the step, frame, and field output variable for which Abaqus/CAE will obtain values. To learn how to change the step, frame, or field output variable, see *Selecting the results step and frame*, and *Selecting the primary field output variable*.

When you probe a model plot of a three-dimensional shell or a composite solid, the results that Abaqus/CAE returns depend on your choice of active location in the **Section Points** dialog box. Abaqus/CAE returns results from the **Top Location** section point when **Top** is the active location and returns results from the **Bottom Location** section point when the active location is set to **Bottom** or **Top and bottom**. For more information about section points, see *Selecting section point data*.

For model plots you can choose to obtain node-based or element-based data and results, as follows:

Probing nodes

When you choose to probe nodes and position the cursor over a nodal location in the current viewport, Abaqus/CAE highlights the node and displays the following information about the node in the data table:

- the node's label,
- the node's original coordinates,
- the node's deformed coordinates (using the current deformed field output variable),
- the elements sharing the node, and
- the current field output variable (including component values, if the current field output variable is a tensor or vector quantity). If a model is selected in the current viewport, the current field output variable reflects the value of a model attribute such as a load at the selected node.

You can display field output at the node whether the current primary variable was originally saved to the output database at nodal, centroidal, or integration point locations. Abaqus/CAE calculates nodal values for centroidal and integration point data by extrapolating to the nodes and averaging according to options you select. If the

node refers to unaveraged element data, all element data associated with the probed node will be displayed. For more information on averaging, see *Understanding result value averaging*.

You can also display or hide annotations that appear in the viewport for each node you select. Probe annotations show the node's label and the results at that node for the current field output variable. You can customize the font and color of the probe annotation text.

Probing elements

When you choose to probe elements and position the cursor over an element in the current viewport, Abaqus/CAE highlights the element and displays the following information about the element in the data table:

- the element's label,
- the element's connectivity, and
- the current field output variable (including component values, if the current field output variable is a tensor or vector quantity). If a model is selected in the current viewport, the current field output variable reflects the value of a model attribute such as a load at the selected element.

You can display field output at the element only if the current primary variable is an element-based variable, such as stress or strain. You can choose the output position at which field output results are calculated: integration point, centroid, element node, or element face. For each of these output positions, Abaqus/CAE calculates probe results on an element-by-element basis with no averaging.

You can also display or hide annotations that appear in the viewport for each element you select. Probe annotations show the element's label and the results at that element for the current field output variable. You can customize the font and color of the probe annotation text.

Regardless of the render style in which the model is displayed, only visible elements can be probed. If necessary, you can create a display group or use a view cut to reveal otherwise unavailable interior elements.

When an element is cut through by a crack, the cracked element splits into two parts, each part formed by a real domain and a phantom domain. When you probe cracked elements, only the contribution from one part of the elements is reported. For more information, see *Visualization*.

X–Y plot probes

If the current viewport contains an *X*–*Y* plot, Abaqus/CAE displays as probe values the *X*–*Y* curve legend text, the sequence identification of each point within the curve, and the *X*- and *Y*-coordinates of curve points. You can choose to display only the curve points that comprise an *X*–*Y* data object or to interpolate to display arbitrary points along the curve. If the *X*–*Y* data name differs from the *X*–*Y* curve legend text, the **Legend** value in the **Probe Values** dialog box displays the *X*–*Y* data name prepended to the legend text. For more information, see *Probing X–Y plots*.

Using the Query toolset to probe the model

You can use the Query toolset to probe model and *X*–*Y* plots, and to write the resulting probe values to a file.

For information on writing selected probe values to a file, see *Generating tabular data reports*.

In this section:

- *Using probe options*
- *Probing nodes*
- *Probing elements*
- *Probing X–Y plots*

Using probe options

You can use the probe options in the Query toolset to specify the values you want to obtain while probing a model or an *X–Y* plot.

Click  in the Visualization module toolbox to access the **Probe Values** dialog box. Use the options in this dialog box to control probing in the current viewport.

For model data from the current model database

Use the **Field Output** dialog box to choose the step and field output variable. For model data, the field output variable you select is actually the value of an attribute in the model, such as a load or predefined field.

For model plots

Use the **Field Output** dialog box to choose the step, frame, and field output variable. Use the **Result Options** dialog box to choose the computation methods governing the probe data and results Abaqus/CAE displays.

For model and X–Y plots

Use the **Probe Values** options to select the type of information you want to obtain while probing.

Additional information

- *Probing nodes*
- *Probing elements*
- *Probing X–Y plots*

Probing nodes

You can use the Query toolset to obtain nodal data and either the values of attributes in the selected model or analysis results from the selected model plot. As you position the cursor over a node in the model or in the model plot in the current viewport, Abaqus/CAE highlights the node. You can choose to display the following information about the node in the data table:

- the node's label,
- the node's original coordinates,
- the node's deformed coordinates (using the current deformed field output variable),
- the elements sharing the node, and
- the current field output variable (including component values, if the current field output variable is a tensor or vector quantity). If a model is selected in the current viewport, the current field output variable reflects the value of a model attribute such as a load at the selected node.



Note:

You can display or hide any of the columns in the table by clicking mouse button 3 in the table, selecting **Edit Visible Columns**, and toggling any of the columns in the dialog box that appears.

You can also obtain a node's data by entering its node label directly, and you can obtain data for all the nodes included in a selected display group. These methods enable you to probe nodal data without positioning the cursor over its location in the model plot.

When you add a node to the data table, you can toggle on the check box for that row to display a probe annotation in the viewport for that node, or you can toggle on the check box in the header row to display probe annotations for all selected nodes. Probe annotations display the node's label and the results at the node for the current field output variable. You can customize the font and color of the annotation text.

To help locate nodes in the viewport, you may want to produce a model plot with node symbols visible before using the Query toolset. For more information, see *Customizing node symbols*.

1. Locate the **Probe Values** dialog box.

   In the Visualization module toolbox, click . The **Probe Values** dialog box appears.

2. Select the step, frame, field output variable, and result options governing the probe data and results Abaqus/CAE will display. If you are probing data from the current model database, frame selections are not available, and your selection of field output variable is an attribute from the model displayed in the current viewport.

   In the **Probe Values** dialog box, the **Field Output** field identifies the current step, frame, and field output variable.

   - To change the step or frame, click .
   - To change the field output variable, click .
   - To change the result options, select **Result->Options** from the main menu.

3. In the **Probe Values** portion of the dialog box, do the following:
   a. Click the **Probe** arrow, and select **Nodes**.

b. If the current primary output variable is a tensor or vector variable, click the **Components** arrow and do either of the following:

- For tensor variables choose **Selected** to display only the selected component of the variable, **All Direct** to include all direct components of the tensor variable in the query, or **All Principals** to include all principal invariants of the tensor variable in the query.
- For vector variables choose **Selected** to display only the selected component of the variable or **All** to include all components of the vector tensor variable in the query.



Note:

If you choose the **All Direct** or **All Principals** option for tensor variables, the results averaging option **Compute scalars before averaging** is not valid; the scalars or invariants are always computed after averaging.

The types of node information available are displayed in the columns of the data table.

4. Click mouse button 3 anywhere in the data table, select **Edit Visible Columns** from the list that appears, and toggle on the node information types that you want to display as columns in the data table.

When a desired information type is toggled on, Abaqus/CAE displays that type of data as a column in the data table. However, all columns, visible or not, are included in the probe values output file.



Note:

Field output is available only if the current primary variable is an element-based variable, such as stress or strain.

5. To specify the nodes that you want to probe, choose **Select from viewport**, **Key-in label**, or **Select a display group**, and use one of the following techniques:
Select the nodes directly from the viewport:

a. Move the cursor to probe the model plot in the current viewport.

As you position the cursor over a node, Abaqus/CAE displays the values of the items you have requested in the first row of the data table.

b. To select and accumulate values of interest, and to allow subsequent writing of these values to a file, click mouse button 1 while positioning the cursor over a node.

Abaqus/CAE stores the data from the first row of the data table into the storage rows of the data table.

Enter the node labels directly into the table:

a. Click the **Part instance** arrow, and select the part instance from which you want to select node labels.

b. Enter the node label or labels that you want to include. You can specify multiple node labels at the same time by separating them with commas.

Select the nodes included in a display group:

a. Click the **Display group** arrow, and select the display group containing the nodes that you want to include in your query.

If you change your selection of display group, Abaqus/CAE replaces the contents of the data table with nodes from the newly selected display group.

b. If desired, append additional nodes to your list by selecting them from the viewport or keying them in by node label.

6. Repeat the node selection steps until the data table in the **Probe Values** dialog box includes all the nodes you want to query.
7. To delete data from the data table or to clear or sort the table, click mouse button 3. For more information, see *Entering tabular data*.
8. To create a display group using the nodes in the data table, do the following:
   a. Click mouse button 3 in the data table, and select **Create Display Group** from the menu that appears.

      The **Create Display Group** dialog box appears.

   b. Enter a name for the new display group, and click **OK**.
9. To display probe annotations for any of the nodes in the data table, toggle on the check box for the rows corresponding to the nodes you want to annotate or toggle on the check box in the header row to display probe annotations for all selected nodes.

   Note:

   You can customize the color and font for probe annotations of nodes in the **Common Plot Options** dialog box. For more information, see *Using common plot options*.

10. To save the data you have selected, click **Write to File**. For more information, see *Generating tabular data reports*."
11. To exit probe mode, do one of the following:

    - Click the cancel button in the prompt area.
    - Click **Cancel** in the **Probe Values** dialog box.
    - Click mouse button 2.
    - Produce a plot in some other (e.g. undeformed, deformed) mode.

Additional information

- *Controlling result averaging*
- *Probes of models or model plots*
- *Probing elements*
- *Querying active node or element labels*

Probing elements

You can use the Query toolset to obtain element data and either the values of attributes in the selected model or analysis results from the selected model plot. As you position the cursor over an element in the model or in the model plot in the current viewport, Abaqus/CAE highlights the element. You can choose to display the following information about the element in the data table:

- the element's label,
- the element's connectivity, and
- the current field output variable (including component values, if the current field output variable is a tensor or vector quantity). If a model is selected in the current viewport, the current field output variable reflects the value of a model attribute such as a load at the selected element.

You can also obtain an element's data by entering its element label directly, and you can obtain data for all the elements included in a selected display group. These methods enable you to probe element data without positioning the cursor over its location in the model plot.

Only exterior elements can be queried. If necessary, you can create a display group or use a view cut to reveal otherwise unavailable interior elements. For more information, see *Using display groups to display subsets of your model*."

When you add an element to the data table, you can toggle on the check box for that row to display a probe annotation in the viewport for that element, or you can toggle on the check box in the header row to display probe annotations for all selected elements. Probe annotations display the element's label and the results at the element for the current field output variable. You can customize the font and color of the annotation text.

To help locate elements in the viewport, you may want to produce a model plot with all element edges visible before using the Query toolset. For more information, see *Controlling element and surface edge visibility*.

1. Locate the **Probe Values** dialog box.

   In the Visualization module toolbox, click . The **Probe Values** dialog box appears.

2. Select the step, frame, and field output variable for which Abaqus/CAE will obtain values. If you are probing data from the current model database, frame selections are not available, and your selection of field output variable is an attribute from the model displayed in the current viewport.

   In the **Probe Values** dialog box, the **Field Output** field identifies the current step, frame, and field output variable.

   - To change the step or frame, click .
   - To change the field output variable, click .
   - To change the result options, select **Result->Options** from the main menu bar.

3. In the **Probe Values** portion of the dialog box, do the following:
   a. Click the **Probe** arrow, and select **Elements**.
   b. If the current primary output variable is a tensor or vector variable, click the **Components** arrow and do either of the following:
      - For tensor variables choose **Selected** to display only the selected component of the variable, **All Direct** to include all direct components of the tensor variable in the query, or **All Principals** to include all principal invariants of the tensor variable in the query.
      - For vector variables choose **Selected** to display only the selected component of the variable or **All** to include all components of the vector tensor variable in the query.

c. To choose the output position at which a field output variable is displayed, click the **Position** arrow and select **Integration Pt**, **Centroid**, **Element Nodal**, or **Element Face**.



Note:

If you choose the **All Direct** or **All Principals** option for tensor variables and the **Element Nodal** output position, the results averaging option **Compute scalars before averaging** is not valid; the scalars or invariants are always computed after averaging.

The types of element information available are displayed in the columns of the data table.

4. Click mouse button 3 anywhere in the data table, select **Edit Visible Columns** from the list that appears, and toggle on the element information types that you want to display as columns in the data table.

When a desired information type is toggled on, Abaqus/CAE displays that type of data as a column in the data table. However, all columns, visible or not, are included in the probe values output file.



Note:

Field output is available only if the current primary variable is an element-based variable, such as stress or strain.

5. To select the elements that you want to probe, choose **Select from viewport**, **Key-in label**, or **Select a display group**, and use one of the following techniques:
Select the elements directly from the viewport:

a. Move the cursor to probe the model plot in the current viewport.

As you position the cursor over an element, Abaqus/CAE displays the values of the items you have requested in the first row of the data table in the **Probe Values** dialog box.

b. To select and accumulate values of interest, and to allow subsequent writing of these values to a file, click mouse button 1 while positioning the cursor over an element.

Abaqus/CAE stores the data from the first row of the data table into the storage rows of the data table.

Enter the element labels directly into the table:

a. Click the **Part instance** arrow, and select the part instance from which you want to select element labels.

b. Enter the element label or labels. You can specify multiple element labels at the same time by separating them with commas.

Select the elements included in a display group:

a. Click the **Display group** arrow, and select the display group containing the elements that you want to include in your query.



Note:

If you change your selection of display group, Abaqus/CAE replaces the contents of the data table with elements from the newly selected display group. If you want to append additional elements to your list, you must select them from the viewport or key them in by element label.

6. Repeat the element selection steps until the data table in the **Probe Values** dialog box includes all the elements you want to query.

7. To delete data from the data table or to clear or sort the table, click mouse button 3. For more information, see *Entering tabular data*.

8. To create a display group using the elements in the data table, do the following:
   a. Click mouse button 3 in the data table, and select **Create Display Group** from the menu that appears.

      The **Create Display Group** dialog box appears.
   b. Enter a name for the new display group, and click **OK**.
9. To display probe annotations for any of the elements in the data table, toggle on the check box for the rows corresponding to the elements you want to annotate or toggle on the check box in the header row to display probe annotations for all selected elements.

   Note:

   You can customize the color and font for probe annotations of elements in the **Common Plot Options** dialog box. For more information, see *Using common plot options*.
10. To save the data you have selected, click **Write to File**. For more information, see *Generating tabular data reports*."
11. To exit probe mode, do one of the following:

    - Click the cancel  button in the prompt area.
    - Click **Cancel** in the **Probe Values** dialog box.
    - Click mouse button 2.
    - Produce a plot in some other (e.g. undeformed, deformed) mode.

Additional information

- *Controlling result averaging*
- *Probes of models or model plots*
- *Probing nodes*
- *Querying active node or element labels*

Probing X–Y plots

You can use the Query toolset to obtain information about curves in an *X*–*Y* plot.
As you position the cursor over an *X*–*Y* curve in the current viewport, Abaqus/CAE highlights the nearest curve point and displays the curve legend text, the point's sequence identification, and the *X*- and *Y*-coordinates of the point in the **Probe Values** dialog box. Abaqus/CAE also highlights the selected curve point with a symbol and displays the *X*- and *Y*-coordinates of that curve point next to the symbol. These symbols and coordinates persist on the *X*–*Y* plot until you leave Probe mode. You can choose to display only the curve points that comprise an *X*–*Y* data object or to interpolate to display arbitrary points along the curve.

When you select a point in Probe mode, Abaqus/CAE adds the legend text, sequence identification, and *X*- and *Y*-coordinates of the current point to the data table, which appears at the bottom of the **Probe Values** dialog box. You can clear or sort this table, delete individual rows, and write these data to a file. If the *X*–*Y* data name differs from the *X*–*Y* curve legend text, the **Legend** value in the **Probe Values** dialog box displays the *X*–*Y* data name prepended to the legend text.

While Probe mode is active, Abaqus/CAE also tracks the position of the cursor in the *X*–*Y* plot with arrows that run along the *X*- and *Y*-axes. These arrows respectively track the current *X*- and *Y*-position of the cursor, so they enable you to select your curve points with greater precision.

To help locate *X*- and *Y*-data points, you may want to produce an *X*–*Y* plot with curve symbols visible before using the Query toolset. For more information, see *Customizing the symbols used on an X–Y curve*.

1. Locate the **Probe Values** dialog box.

   In the Visualization module toolbox, click . The **Probe Values** dialog box appears.

2. At the top of the dialog box, toggle **Interpolate between points** to enable or disable the display of probe values for arbitrary points along the curve.

   When you enable interpolation, Abaqus/CAE displays the sequence identification of arbitrary points by listing the bracketing *X*–*Y* data object points in parentheses.

3. Select the information types that you want to include in your query.

   Click mouse button 3 anywhere in the data table, select **Edit Visible Columns** from the list that appears, and toggle on the information types that you want to include in the query and display as columns in the data table. When an information type is toggled on, Abaqus/CAE can display that type of data in two locations for the selected element: in the data table when you mouse over or select the element in the viewport, and in the probe values output file.

4. Move the cursor to probe the *X*–*Y* plot in the current viewport.

   As you position the cursor over a curve point, Abaqus/CAE displays the *X*- and *Y*-coordinates of the current point on the curve itself and displays the current position of your cursor along the *X*- and *Y*-axes of the *X*–*Y* plot.

5. To select and accumulate values of interest and to allow subsequent writing of these values to a file, click mouse button 1 while positioning the cursor over a curve point.

   Abaqus/CAE stores the legend text, sequence identification, and *X*- and *Y*-coordinates of the current point into the data table. Abaqus/CAE also tags the points you select in the viewport by placing a symbol at the curve point you selected and displaying the *X*- and *Y*-coordinates next to that point.

6. To delete data from the data table or to clear or sort the table, click mouse button 3. For more information, see *Entering tabular data*.
7. To save the data you have selected, click **Write to File**. For more information, see *Generating tabular data reports*.
8. To exit probe mode, do one of the following:

- Click the cancel  button in the prompt area.
- Click **Cancel** in the **Probe Values** dialog box.
- Click mouse button 2.
- Produce a plot in some other (e.g., undeformed, deformed) mode.

Additional information

• *X–Y plot probes*

Calculating linearized stresses

The calculation of linearized stresses is a capability for advanced Abaqus users.
It is most commonly used for two-dimensional axisymmetric models.

For detailed information on linearized stress result quantities and the methods of calculation used, see *Computing the stress components*.

In this section:

- *Understanding stress linearization*
- *Stress linearization example*
- *Obtaining linearized stress results*

Understanding stress linearization

Stress linearization is the separation of stresses through a section into constant membrane and linear bending stresses. Stress linearization is available for output databases only.
To linearize stresses in Abaqus/CAE, the following procedure is used:

- You define a section through your model. You specify the endpoints of the section by selecting nodes in the model, coordinates in space, or a saved path.
- Abaqus/CAE defines the stress line by interpolating between the two endpoints to obtain a user-specified number of equal intervals along a straight line.
- You can save the stress line as a path; the saved path includes the endpoints and each interval point along the stress line. If you save the stress line as a path, Abaqus/CAE always creates a point list path, regardless of the method that you used to define the endpoints.
- The specified stress results are obtained at equally spaced intervals along the line in a local coordinate system defined by the line.
- Abaqus/CAE performs stress linearization calculations and displays the results in the form of an *X*–*Y* plot. You can choose to save the data and/or to write it to a file.

If you linearize stresses across multiple part instances, the results will be averaged for any points that exist in more than one part instance.

To obtain linearized stresses along a line, select **Tools->Query** from the main menu bar, then select **Stress linearization** from the **Visualization Module Queries** options.

In this section:

- *Defining the stress line*
- *Extraction of the stress results*

Defining the stress line

You can define the endpoints of the section for which you want to obtain linearized stresses by selecting nodes in the model, points in space, or a saved path. You select nodes in the model by selecting the nodes directly from the viewport or by typing the node labels, and you define points in space by typing the point coordinates. If you select a saved path, Abaqus/CAE uses the path endpoints as the endpoints of the stress line—other points along the path are ignored. (For more information on paths, see *Viewing results along a path*.)

A node label is a persistent means of referring to a given node in a part instance; that is, node labels do not change as your model deforms. Therefore, if you specify the endpoints of the section using node labels or a saved node list path, these points will be equally applicable to the undeformed or the deformed model shape. However, points in space remain fixed and are independent of your model. For example, coordinates that coincide with a nodal location on the undeformed shape may not coincide with any location on the deformed shape. If you specify the endpoints of the section using coordinates in space, a point list path, or a radial path, be sure to specify the coordinates with respect to the correct shape. Edge list paths cannot be used to define a stress line.

For the best results for an axisymmetric model, the endpoints chosen should be normal to the interior and exterior surfaces of your model and to the midplane (see *Figure 1*).

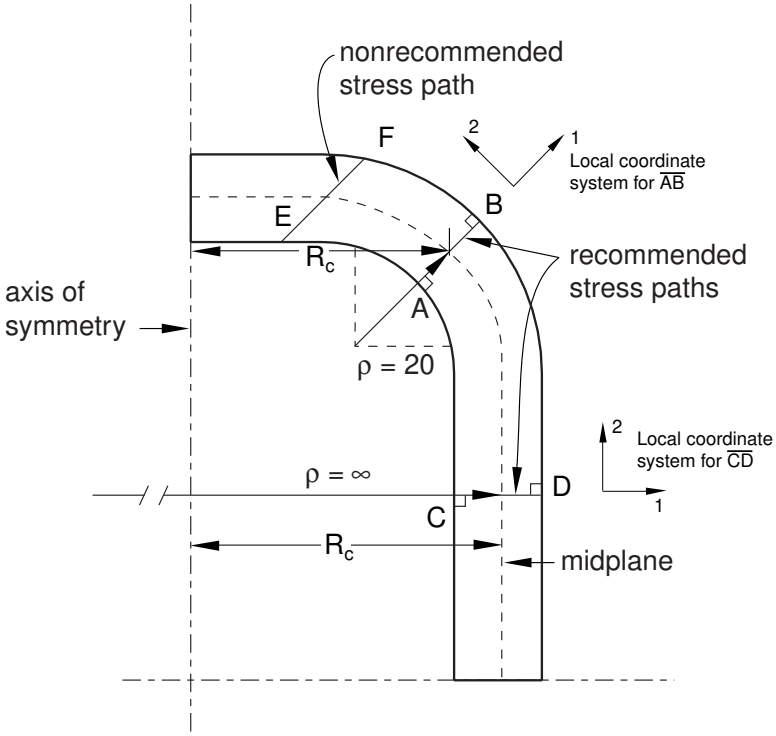


Figure 1: Recommended stress paths.

The section can extend from surface to surface, or it can be entirely in the interior of the model. The section can also pass through more than one part instance. The following guidelines should also be considered when selecting the endpoints for the section:

- Avoid paths that traverse spatial discontinuities.
- Avoid paths along a line separating discontinuities.
- Use display groups to isolate individual regions prior to specifying the section.

If the section extends across spatial discontinuities, such as a hole within a part instance or a space between part instances, an error message will be given.

After you choose the endpoints of the section, you can specify the number of segments into which the line should be divided or you can accept the default number. Abaqus/CAE defines the stress line by interpolating between the two endpoints to obtain the specified number of equal intervals along a straight line.

Extraction of the stress results

The stresses at each point along the stress line are obtained by interpolating the unique nodal stress values using the averaging criteria defined in the **Result Options** dialog box (for more information on specifying the averaging criteria, see *Controlling result averaging*). Abaqus/CAE uses a local coordinate system that is defined by the stress line to obtain the stress results. The local *x*-axis lies along the stress line; it originates at the first point on the stress line and is defined by the vector connecting the first and second points on the stress line. The local *y*- and *z*-axes are determined as follows:

For two-dimensional models:

The local *y*-axis is obtained by vector multiplication of the global *z*-axis with the local *x*-axis. The local *z*-axis is the same as the global *z*-axis.

For three-dimensional models:

If the local *x*-axis is parallel to the global *y*-axis:

If the positive local *x*-axis lies along the positive global *y*-axis, the local *y*-axis will be the negative of the global *x*-axis. If the positive local *x*-axis lies along the negative global *y*-axis, the local *y*-axis will be the global *x*-axis. The local *z*-axis is obtained by vector multiplication of the local *x*-axis and the local *y*-axis.

If the local *x*-axis is not parallel to the global *y*-axis:

The local *z*-axis is obtained by vector multiplication of the local *x*-axis and the global *y*-axis. The local *y*-axis is obtained by vector multiplication of the local *z*-axis and the local *x*-axis.

The stresses are then linearized. The stress linearization calculations differ for nonaxisymmetric and axisymmetric models. For more information on these calculations, see *Computing the stress components*.

Abaqus displays the resulting linearized stresses in the form of an *X–Y* plot. Three separate curves are presented for each stress component representing the following:

1. The actual stress distribution across the stress line.
2. The linearized membrane stress.
3. The sum of the linearized membrane stress and the linearized bending stresses.

In addition, the linearized stress output for each stress component and the stress invariant calculations for the linearized stresses are written to a report file.

Stress linearization example

Figure 1 shows an example of a stress line defined for an axisymmetric model of a pressure vessel.

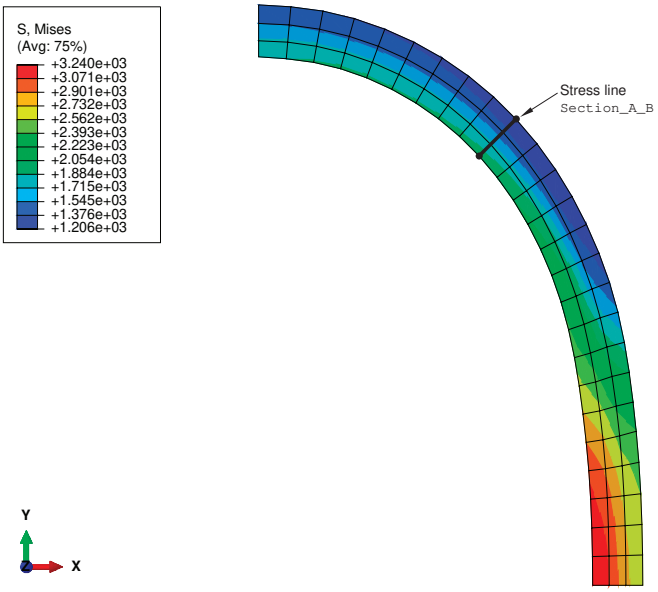


Figure 1: Stress line through an axisymmetric model of a pressure vessel.

The stress line `Section_A_B` is defined through the vessel wall. *Figure 2* and *Figure 3* show the basic settings and computations, respectively, that you use to linearize the S22 stress component for the undeformed model shape.

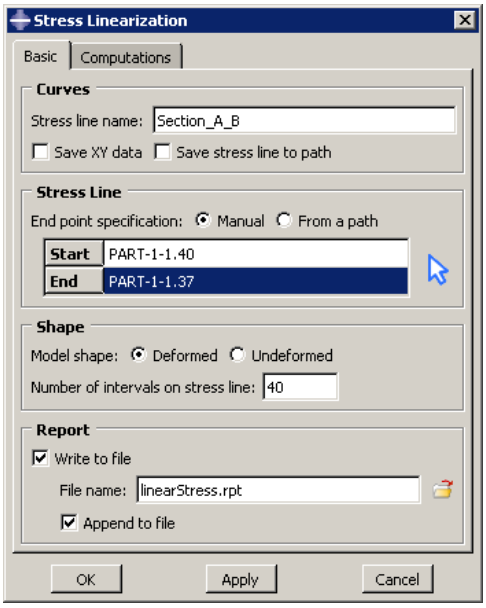


Figure 2: Stress linearization basic specifications.

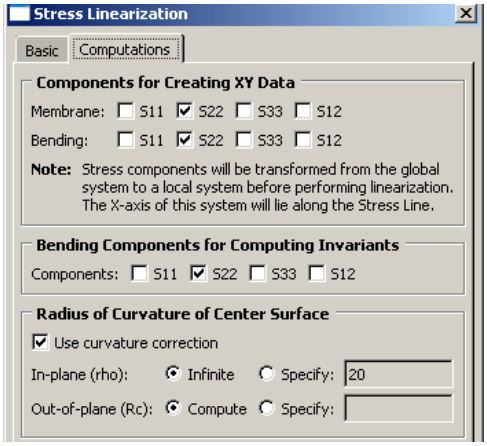


Figure 3: Stress linearization computations.

When you click **OK** or **Apply** in the **Stress Linearization** dialog box, Abaqus/CAE creates an *X*–*Y* plot of the S22 stress component (oriented normal to the stress line) and of the resulting linearized stresses, as shown in *Figure 4*.

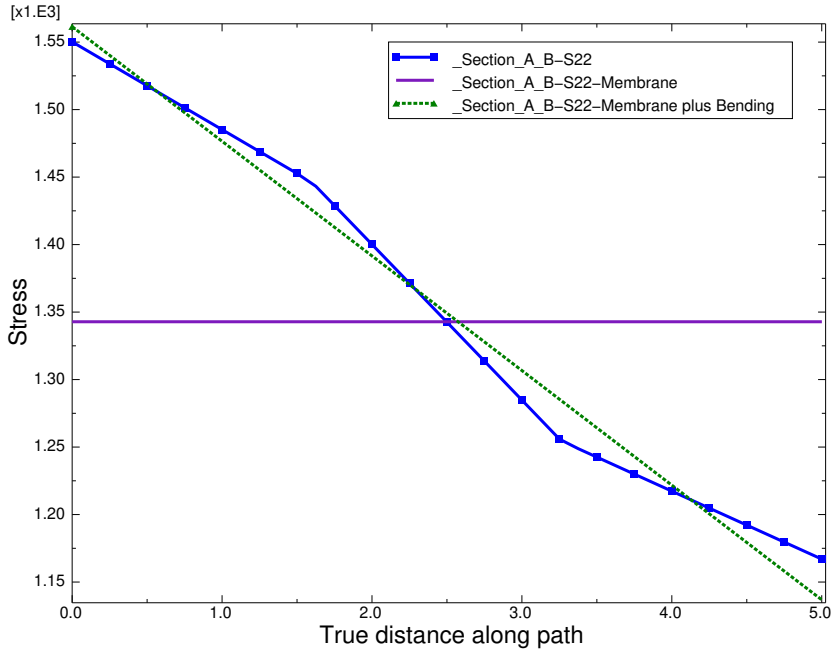


Figure 4: Linearized stress plot.

The following output is also written to a file called `linearStress.rpt`:

```
*******************************************************************************
Statically Equivalent Linear Stress Distribution across a Section,
written on Thu Sep 09 11:20:19 2010

Source
-------

   ODB: Job-1.odb
   Step: Step-1
   Frame: Increment      1: Step Time =    1.000
```

```
Linearized Stresses for stress line 'Section_A_B'
  Start point, Point 1 - (18.429651260376, 26.8930339813232, 0)
  End point, Point 2   - (22.0184745788574, 30.3756923675537, 0)
  Number of intervals  - 40

------------------------------ COMPONENT RESULTS ------------------------------


                       S11         S22         S33         S12

         0          -462.376     1550.19     1450.75     74.7673
  0.125021          -453.722     1542.06     1445.35     74.6265
  0.250043          -445.068     1533.93     1439.95     74.4865
  0.375064          -436.413      1525.8     1434.55     74.3473
  0.500086          -427.759     1517.67     1429.15     74.2089
  0.625107          -419.114     1509.55     1423.76     74.0714
  0.750128           -410.46     1501.42     1418.36     73.9345
   0.87515          -401.806      1493.3     1412.96     73.7983
   1.00017          -393.152     1485.17     1407.56      73.663
   1.12519          -384.497     1477.04     1402.16     73.5284
   1.25021          -375.842     1468.92     1396.76     73.3946
   1.37524          -367.187     1460.79     1391.37     73.2615
   1.50026          -358.531     1452.67     1385.97     73.1293
   1.62528          -348.574     1443.22      1379.7     72.8307
    1.7503           -333.79     1428.85     1370.22       71.77
   1.87532          -319.007     1414.48     1360.74     70.7052
   2.00034          -304.227      1400.1     1351.26     69.6367
   2.12536          -289.448     1385.72     1341.78     68.5648
   2.25039          -274.656     1371.33     1332.29     67.4908
   2.37541          -259.847     1356.91     1322.81     66.4061
   2.50043          -245.037     1342.49     1313.32     65.3195
   2.62545          -230.228     1328.07     1303.83     64.2284
   2.75047          -215.421     1313.64     1294.34     63.1328
   2.87549          -200.613      1299.2     1284.84     62.0327
   3.00051          -185.807     1284.76     1275.34     60.9282
   3.12554          -171.002     1270.32     1265.84     59.8191
   3.25056          -156.197     1255.88     1256.34     58.7056
   3.37558          -149.216     1248.82     1251.71      57.583
    3.5006          -143.031     1242.52     1247.58     56.4609
   3.62562          -136.844     1236.21     1243.45       55.34
   3.75064          -130.658     1229.91     1239.32     54.2204
   3.87566          -124.471     1223.61     1235.19     53.1021
   4.00069          -118.283     1217.31     1231.06      51.985
   4.12571          -112.095     1211.02     1226.93     50.8691
   4.25073          -105.907     1204.72      1222.8     49.7545
   4.37575          -99.7185     1198.42     1218.67     48.6412
   4.50077          -93.5296     1192.13     1214.55      47.529
   4.62579          -87.3403     1185.83     1210.42     46.4182
   4.75081          -81.1506     1179.54      1206.3     45.3086
   4.87584          -74.9605     1173.25     1202.17     44.2002
   5.00086            -68.77     1166.96     1198.05     43.0931

     Membrane
(Average) Stress    -253.255     1342.83     1317.88     62.6971

     Bending
 Stress, Point 1     -209.122     218.613     140.324           0

  Membrane plus
```

```
Bending, Point 1    -462.376     1561.45      1458.2     62.6971

     Bending
 Stress, Point 2      184.485    -206.054    -140.324           0

  Membrane plus
Bending, Point 2      -68.77     1136.78     1177.55     62.6971

   Peak Stress,
     Point 1               0    -11.2522    -7.44933     12.0701

   Peak Stress,
     Point 2               0     30.1809     20.4932     -19.604

------------------------------ INVARIANT RESULTS ------------------------------

Bending components in equation for computing
membrane plus bending stress invariants are:   S22

                     Max.        Mid.        Min.       Tresca       Mises
                     Prin.       Prin.       Prin.      Stress      Stress
     Membrane
(Average) Stress     1345.29     1317.88    -255.714     1601.01     1587.48

  Membrane plus
Bending, Point 1     1563.61     1317.88    -255.418     1819.03     1709.46

  Membrane plus
Bending, Point 2     1317.88      1139.6    -256.077     1573.95     1492.82

   Peak Stress,
     Point 1          132.875    -10.5186    -209.855      342.73     298.128

   Peak Stress,
     Point 2          186.936     27.7292    -119.831     306.767     265.732
```

The `S22` corresponds to the S22 stress shown in *Figure 4*. The actual stress values plotted in the curve `Section_A_B_S22` do not appear in the report. The linearized membrane and membrane-plus-bending stress curves are generated from the values shown for `S22`. The reported invariants are calculated from the selected linearized components.

Obtaining linearized stress results

You can obtain linearized stresses along a line through your model. Abaqus/CAE separates the stress results along the line into constant membrane and linear bending stresses and displays the results in the form of an *X*–*Y* plot.

1. Locate the stress linearization options.

   From the main menu bar, select **Tools->Query** or click the  tool in the Query toolset.

   The **Query** dialog box appears.

2. Select **Stress linearization**.

   The **Stress Linearization** dialog box appears.

3. In the **Stress line name** field, provide a name for the stress line. This name will be used as a prefix for the linearized results.
4. To save the *X*–*Y* data that will be generated, toggle on **Save XY data**. The data will be available for the duration of your Abaqus/CAE session.
5. To save the endpoints and interval points as a path, toggle on **Save stress line to path**. The points of the stress line will be saved for the duration of your Abaqus/CAE session as a point list path with the same name as the stress line.
6. Select the endpoints for the stress line by selecting nodes or points in space or by selecting a saved path.

Manual

This is the default method. You can select nodes directly from the viewport or type in node labels or points in space.

To select nodes directly from the viewport:

Click  to the right of the **Start** and **End** fields, and click on the desired nodes in the viewport.

The node labels, including the part instance name, will appear in the text field for the **Start** and **End** points of the stress line.

To type in node labels or points in space:

1. In the **Start** text field, enter a part instance name and node label or the coordinates of a point in space. The part instance name and node label must be of the form `Instance.Node`; specify coordinate values as *X*-, *Y*-, and *Z*-coordinates separated by spaces or commas.

   If you do not know the part instance names in the model, use the previous method to select a node directly from the viewport. Alternatively, you can select **Tools->Query** or click the  tool in the Query toolset and choose the **Probe values** method to determine the instance name and node label (for more information, see *Understanding probing*).

2. Repeat the preceding step to complete the **End** text field.

From a path

Toggle on **From a path**, and select a path name from the list that appears; you cannot use an edge list path to define the endpoints of a stress line. (For more information on paths, see *Viewing results along a path*.)

Abaqus/CAE uses the endpoints of the saved path as the endpoints of the stress line. The points are defined in the same manner as they were originally defined in the path—a node list path provides node points on the model, and a point list path or circular list path provides the coordinates of points in space.

Regardless of the method you use to select the endpoints, Abaqus highlights the stress line in the viewport and labels the start and end. If you selected node points or a node list path to define the endpoints of the stress line, the labels in the viewport indicate the node numbers.



Note:

If you chose to save the stress line as a path in Step 5, Abaqus/CAE*always* saves a point list path—even if you selected nodes, node labels, or a node list path to define the stress line.

7. Choose the model shape for which to obtain the stress results. The default is to obtain the results for the deformed model shape. Toggle on **Undeformed** to obtain results for the undeformed model shape.
8. Specify the number of intervals into which the stress line should be divided. Type a positive integer greater than 0 into the **Number of intervals on stress line** text field. If you do not enter a number, Abaqus will use a default number of intervals.
9. Select **Use Maximum Stress Value** to use the maximum available value at a point.

   By default, Abaqus/CAE will use the average of multiple values available at a point.
10. By default, Abaqus/CAE writes the linearized stress values (including all available components of stress and the computed linearized stress invariants) to a file called `linearStress.rpt`. If you do not wish to write this report, you can toggle off **Write to file** in the **Report** area of the dialog box.

    You can specify a new name for the report file by entering the name in the **File name** field or clicking  and choosing from the list of existing files that appears.

    If you write the report to an existing file, the new data will be appended to the file by default; if you wish to overwrite the file, toggle off **Append to file**.
11. Click the **Computations** tab.
12. Select the stress components to be linearized by toggling on each membrane and bending component.
13. Select the bending components to use for calculating invariants.
14. For axisymmetric models enter an approximate value for the in-plane radius of curvature of the midplane of your model at the location of the stress line. The default value is **Infinite**, which implies a lack of curvature. To specify a radius of curvature, click **Specify** and enter a number in the text field.
15. For axisymmetric models enter an approximate value for the out-of-plane radius of curvature of the midplane of your model at the location of the stress line. The default value is **Compute**, which allows Abaqus to calculate the radius based on the axisymmetric shape and the position of the selected stress line. To specify a radius of curvature, click **Specify** and enter a number in the text field.
16. For nonaxisymmetric models select whether Abaqus should use curvature correction. If curvature correction is selected, specify a local coordinate system or use the default (global) coordinate system; if you use a local coordinate system for curvature correction, you can also use that local coordinate system to evaluate stress line orientation.
17. Click **OK**.

An *X–Y* plot similar to the one shown in *Figure 1* appears in the viewport.

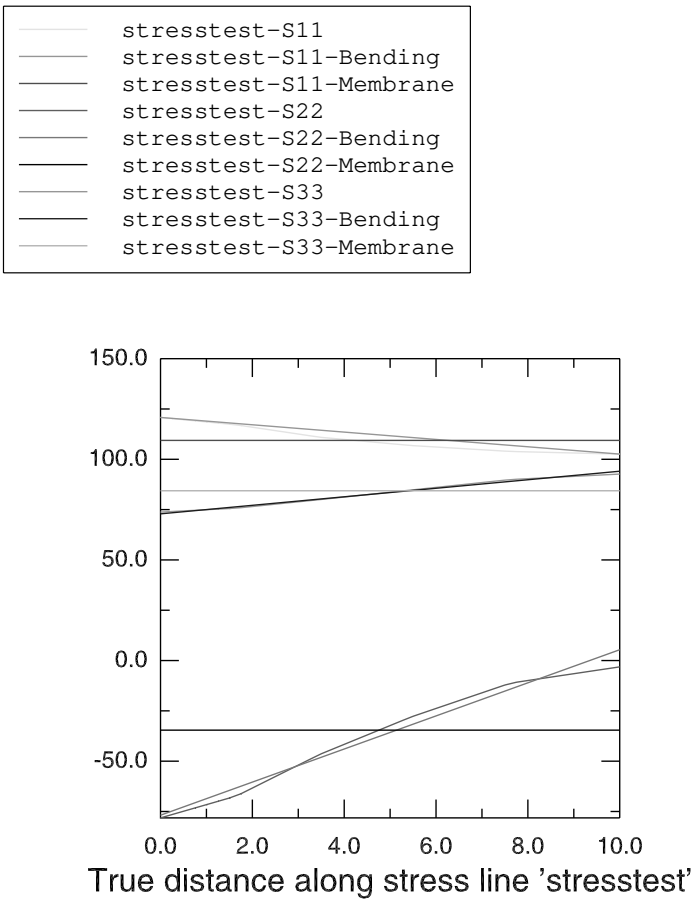


Figure 1: Stress linearization results.

Additional information

• *X–Y plotting*

Viewing a ply stack plot

A ply stack plot shows the plies from the selected region of a composite model. You can create a ply stack plot in either the Property module or the Visualization module. This chapter describes ply stack plotting.

In this section:

- *Understanding ply stack plots*
- *Customizing ply stack plots*

Understanding ply stack plots

A ply stack plot is a graphical representation of the plies from a selected region of a composite model. In effect, a ply stack plot is a core sample of the plies in the selected region. The region can be from a composite layup or from a composite section.

You can customize the appearance of a ply stack plot. For example, the image can show the sequence of plies, the orientation of the fibers in each ply, the material in each ply, the relative thickness of the ply, and the location of the reference surface. *Figure 1* illustrates a ply stack plot.

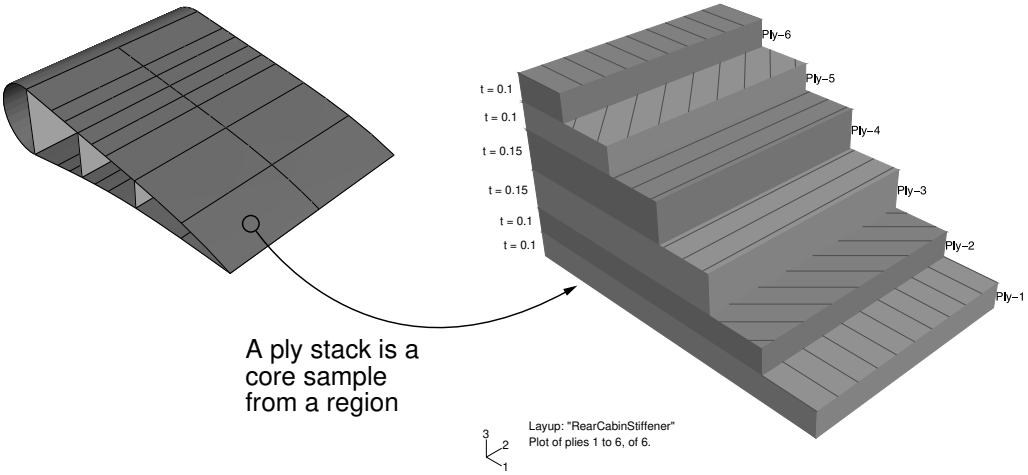


Figure 1: A ply stack plot.

The staircase appearance has no corresponding physical meaning; it is just a mechanism that allows you to view all of the plies in a region. For example, the plies are not ending at each stair.

The triad in a ply stack plot represents the element orientation system, and it shows the shell normal or stacking direction (the 3-direction) and the 1- and 2-directions for the plies. The fibers are always drawn in the 1–2 plane at an angle with respect to the 1-direction. In solid composite layups the fibers in a ply do not always run parallel to the 1–2 plane (e.g., if the 3-direction of the ply orientation and the element stacking direction are not aligned). In this situation the fibers in the ply stack plot are not a true depiction of the fibers in the composite, but rather a graphical representation of the rotation angles in the composite section or layup definition: the angle drawn in the ply stack plot is the rotation angle specified in the ply table measured about the element stacking direction axis. For more information, see *Understanding composite layups and orientations*.

Ply stack plots do not draw fibers for ply orientations that are based on a user-defined coordinate system or a rotation angle distribution. Abaqus/CAE displays an asterisk (for coordinate systems) or a caret (for rotation angle distributions) on such plies to signify that it cannot draw lines in the 1–2 plane that accurately represent the fiber direction for the ply. Similarly, if you use a discrete field distribution to define a ply's thickness, Abaqus/CAE draws the ply in the ply stack plot based on the average thickness of the other uniform plies in the layup and displays the discrete field name next to the plot (if thickness labels are turned on).

It is important to know the position of the reference surface when you are modeling a composite with conventional shell elements. The mesh will be positioned on the reference surface, and contact is defined relative to the reference surface. The ply stack plot allows you to view the reference surface relative to other plies in the composite, as shown in *Figure 2*.

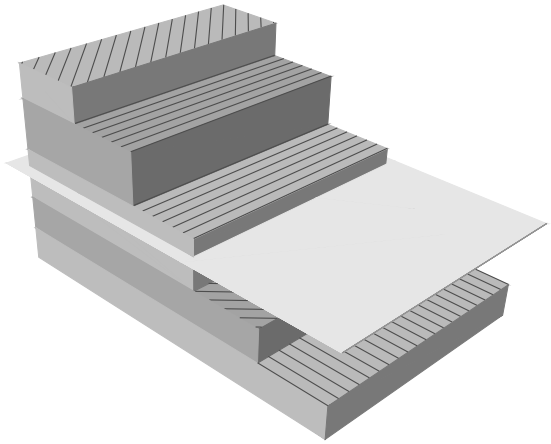


Figure 2: The reference surface in a ply stack plot.

You can display a ply stack plot in the Property module after you have created a composite layup or assigned a composite section to a region. You can also display a ply stack plot in the Visualization module after you have analyzed a model with a composite layup or a composite section. For more information, see *Defining composite layups* and *Creating composite shell sections*.

To produce a ply stack plot, select **Tools->Query** from the main menu bar and select **Ply stack plot** from the dialog box that appears. After you have displayed a ply stack plot, you can customize its appearance by clicking the **Ply Stack Plot Options** button from the prompt area or by clicking the tool in the toolbox (in the Visualization module). For detailed instructions on customizing a ply stack plot, see *Customizing ply stack plots*.

Customizing ply stack plots

This section explains how you use the **Ply Stack Plot Options** dialog box to customize the appearance of a ply stack plot.

In this section:

- *Using ply stack plot options*
- *Customizing basic ply stack plot options*
- *Customizing the appearance of fibers in a ply stack plot*
- *Customizing the reference surface in a ply stack plot*
- *Customizing the labels in a ply stack plot*

Using ply stack plot options

Understanding ply stack plots describes how you can create a ply stack plot. After you have displayed a ply stack plot, you can customize its appearance by clicking the **Ply Stack Plot Options** button from the prompt area. Alternatively, you can do the following:

- In the Property module, choose **View->Ply Stack Plot Options**.
- In the Visualization module, choose **Options->Ply Stack Plot** or click the  tool in the toolbox.

Abaqus/CAE displays the **Ply Stack Plot Options** dialog box, which allows you to do the following:

- Choose which plies are displayed in the ply stack plot and how the plies are displayed. For more information, see *Customizing basic ply stack plot options*.
- Choose whether fibers are displayed in the ply stack plot and the appearance of the fibers. For more information, see *Customizing the appearance of fibers in a ply stack plot*.
- Choose whether the reference surface is displayed in the ply stack plot and the appearance of the reference surface. For more information, see *Customizing the reference surface in a ply stack plot*.
- Customize the text and symbols that are displayed in a ply stack plot. For more information, see *Customizing the labels in a ply stack plot*.

Customizing basic ply stack plot options

Click the **Basic** tab to choose which plies are displayed in the ply stack plot and how the plies are arranged. If your composite layup contains many plies, the ply stack plot can become confusing and hard to interpret. To make the ply stack plot more readable, the basic ply stack plot options allow you to view only a manageable number of plies.

1. Locate the basic options.
   1. Click the **Ply Stack Plot Options** button from the prompt area, or do one of the following:
      - In the Property module, choose **View->Ply Stack Plot Options**.
      - In the Visualization module, choose **Options->Ply Stack Plot** or click the  tool in the toolbox.
   2. Click the **Basic** tab in the **Ply Stack Plot Options** dialog box that appears.
2. From the **Display** options, do the following:
   - Select the render style—shaded, filled, or wireframe.
   - Choose whether to display the edges of each ply and the style and thickness of the edges.
   - Choose whether to display the plies as a “staircase” image descending from the last ply to the first ply or as an image that is symmetric about the center ply or plies.
3. From the **Visible Plies** options, choose which plies to display. Abaqus/CAE updates the ply stack plot when you modify the display parameters, which allows you to quickly and conveniently display the plies in a composite. You can choose the following:
   - Display the specified plies, starting with the **Start** ply and ending with the **End** ply.
   - Display **Number** of plies, starting with the **Start** ply. Click **Increment by Number** to increment the **Start** value by **Number** of plies instead of by 1. This allows your ply stack plot to be a window that moves incrementally across the layup, with each increment displaying the next adjacent group of plies.
   - Display **Number** of plies, centered around the **Center** ply. Click **Increment by Number** to increment the **Center** value by **Number** of plies instead of by 1. This allows your ply stack plot to be a window that moves incrementally across the layup, with each increment displaying the next adjacent group of plies.
   - Display **Number** of plies, centered around the **Center** ply. Drag the **Center** slider to move across the layup.
4. Choose the color of the **Even** and **Odd** plies:
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

   Abaqus/CAE applies your choice of color relative to the plies that are displayed, not to the actual ply number. For example, if you are displaying plies 2, 3, and 4; Abaqus/CAE will apply the **Odd** color to plies 2 and 4 and the **Even** color to ply 3.

5. Click **Apply** to implement your changes.

   The ply stack plot changes to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent ply stack plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing the appearance of fibers in a ply stack plot

Click the **Fiber** tab to choose whether fibers are displayed in the ply stack plot and the appearance of the fibers. The fibers are always drawn in the 1–2 plane at an angle with respect to the 1-direction. For more information, see *Understanding composite layups and orientations*.

If you use a coordinate system to define the ply's orientation, Abaqus/CAE cannot determine the orientation within an element without knowing the spatial orientation of the element. Therefore, Abaqus/CAE displays an asterisk (*) in place of the ply fibers to signify that it cannot draw lines in the 1–2 plane that represent the fiber direction. For the same reason, Abaqus/CAE cannot draw lines that accurately represent the fiber direction if the ply's orientation is defined using a discrete field distribution; a caret (^) in place of the ply fibers indicates that the additional rotation in a ply's orientation is defined using a discrete field.

In solid composite layups the fibers in a ply do not always run parallel to the 1–2 plane (e.g., if the 3-direction of the ply orientation and the element stacking direction are not aligned). Since fibers in a ply stack plot are always drawn in the 1–2 plane, they are not a true depiction of the fibers in the solid composite. Instead the fibers are a graphical representation of the rotation angles in the composite section or layup definition: the angle drawn in the ply stack plot is the rotation angle specified in the ply table measured about the element stacking direction axis.

1. Locate the fiber options.
   1. Click the **Ply Stack Plot Options** button from the prompt area, or do one of the following:
      - In the Property module, choose **View->Ply Stack Plot Options**.
      - In the Visualization module, choose **Options->Ply Stack Plot** or click the  tool in the toolbox.
   2. Click the **Fiber** tab in the **Ply Stack Plot Options** dialog box that appears.
2. Do the following:
   - Toggle on **Show fibers** to display fibers in the ply stack plot.
   - Choose the color, style, and thickness of the lines that represent fibers in the ply stack plot.
   - Drag the **Spacing** slider to change the number of fibers that are displayed.
3. Click **Apply** to implement your changes.

   The ply stack plot changes to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent ply stack plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing the reference surface in a ply stack plot

Click the **Reference Plane** tab to choose whether the reference surface is displayed in the ply stack plot and to control the appearance of the reference surface. Conventional shell elements use the reference surface to define the geometry of the part. Therefore, you can display the reference surface only on shell composites; you cannot display the reference surface on continuum shell composite layups or solid composite layups. In addition, if the ply's offset is defined by a distribution, Abaqus/CAE cannot display the reference surface.

1. Locate the reference surface options.
   1. Click the **Ply Stack Plot Options** button from the prompt area, or do one of the following:
      - In the Property module, choose **View->Ply Stack Plot Options**.
      - In the Visualization module, choose **Options->Ply Stack Plot** or click the  tool in the toolbox.
   2. Click the **Reference Plane** tab in the **Ply Stack Plot Options** dialog box that appears.
2. Do the following:
   - Toggle on **Show reference surface** to display the reference plane in the ply stack plot.
   - Choose the color and translucency of the reference surface.
   - Toggle on **Show reference outline** to display lines around the edge of the reference surface. Choose the color, style, and thickness of the lines.
3. Click **Apply** to implement your changes.

   The ply stack plot changes to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent ply stack plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing the labels in a ply stack plot

Click the **Labels** tab to customize the text and symbols that are displayed in a ply stack plot; for example, material and ply names, the state block, and integration points.

1. Locate the labels options.
   1. Click the **Ply Stack Plot Options** button from the prompt area, or do one of the following:
      - In the Property module, choose **View->Ply Stack Plot Options**.
      - In the Visualization module, choose **Options->Ply Stack Plot** or click the  tool in the toolbox.
   2. Click the **Labels** tab in the **Ply Stack Plot Options** dialog box that appears.
2. Select the font of all the text in the ply stack plot.
   a. Click **Set Font for All Labels** to display the **Select Font** dialog box.
   b. Use the font selector dialog box to set the font properties that you want.
   c. Click **OK** to close the **Select Font** dialog box and to return to the **Ply Stack Plot Options** dialog box.
3. Toggle on an option to display it.
4. Choose the color of the text:
   a. Click the color sample .
      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.
      The color sample changes to the selected color.
5. Click **Apply** to implement your changes.
   The ply stack plot changes to reflect your specifications.
   By default, your changes are saved for the duration of the session and will affect all subsequent ply stack plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Generating tabular data reports

You can produce a tabular data report of *X*–*Y* data objects, field output results, probe values, or free body cuts. Abaqus/CAE writes the report to the file name of your choice. Generate a tabular report to save data values beyond the duration of the session or to print these values. In addition, you can use a tabular report as ASCII input in subsequent Abaqus/CAE sessions.

This chapter explains how to produce and customize a tabular report.

In this section:

- *Producing a tabular report*
- *Using tabular report options*
- *Selecting report data*
- *Specifying your report file name*
- *Controlling report layout, width, format, and coordinate system*
- *Sorting field output data*
- *Formatting report values*
- *Reporting data values, minimums, maximums, and totals*

Producing a tabular report

You can produce tabular reports of *X–Y* data objects, field output results, free body cuts, and probe values.

To produce a tabular report of *X–Y* data objects, field output results, or free body cuts, you first specify the report content, format, and destination. Abaqus/CAE generates the report when you apply these customization selections.

To produce a report of probe values, you first produce and then probe a model or *X–Y* plot. While probing, you can select probe values of interest and you can write the selected values to a file.

Depending on your selections, your report may include negative element labels. Negative labels indicate internal elements created by Abaqus/CAE. These elements are given negative labels so you can easily distinguish them from elements that you created while defining the model.

Produce a report of X–Y data, field output results, or free body cuts

1. From the main menu bar, select **Report->XY**, **Report->Field Output**, or **Report->Free Body Cut**. The **Report XY Data**, **Report Field Output**, or **Report Free Body Cut** dialog box becomes available.
2. Use the options to customize the report content, format, and destination.
3. When you have finished, click **Apply** to generate the report.

Produce a report of probe values

1. From the main menu bar, select **Tools->Query**, then click **Probe values** in the dialog box that appears. The **Probe Values** dialog box appears.
2. Move the cursor to probe the model or *X–Y* plot in the current viewport.
3. To select and accumulate values of interest in the data table in the bottom portion of the **Probe Values** dialog box, click mouse button 1 while probing.
4. When you have finished, click **Write to File** to access the report file options and to generate the report.

Using tabular report options

You can customize tabular report characteristics such as the objects to include, report layout, and sort order. Select **Report->XY**, **Report->Field Output**, or **Report->Free Body Cut** to customize and generate a tabular listing of *X–Y* data objects, field output variable values, or free body cuts, respectively. Select **Tools->Query** to generate a tabular report of probe values. The following list identifies the tabular report characteristics that you can customize and the sections where you can find information.

You can customize:

- The *X–Y* data objects or field output variables to include in the report. See *Selecting report data*.
- The free body cuts to include in the report. Abaqus/CAE includes all active free body cuts in the current output database in a free body cut report. You can activate and suppress free body cuts from the **Free Body Cut Manager**. See *Displaying, hiding, and highlighting free body cuts*.
- The probe values to include in the report. See *Probing the model*.
- The file name to which the report is written. See *Specifying your report file name*.
- The layout and width of the report. See *Controlling report layout, width, format, and coordinate system*.
- How field output or probe values are sorted. For field output, see *Sorting field output data*; for probe values, see *Entering tabular data*.
- The format and significant digits of report values. See *Formatting report values*.
- Whether or not to include minimum and maximum value summaries or column totals. See *Reporting data values, minimums, maximums, and totals*.
- The form of complex results in the report. See *Controlling the form of complex results*.

Selecting report data

This section explains how to select *X–Y* data objects to include in an *X–Y* data report, how to select field output variables to include in a field output report, and how to select the options and thresholds to use for a free body cut report.

To learn how to control the contents of a tabular report of probe values, see *Probing the model*.

In this section:

- *Selecting X–Y data objects*
- *Selecting field output variables*
- *Selecting field output section points*
- *Selecting options and thresholds for free body cut reports*

Selecting X–Y data objects

You can select one or more *X–Y* data objects to include in your *X–Y* report.

1. Locate the *X–Y* report **XY Data** options.

   From the main menu bar, select **Report->XY**; then click the **XY Data** tab in the dialog box that appears.

2. Select the *X–Y* data objects to include in your report from the list of available data objects. By default, all *previously saved* *X–Y* data objects are listed. Choose one of the following options to adjust the contents of this list:

   - Click **All XY data** to browse all previously saved *X–Y* data objects.
   - Click **XY plot in current viewport** to browse only those *X–Y* data objects included in the *X–Y* data plot in the current viewport.

   Abaqus/CAE lists the *X–Y* data objects you have requested.

3. To limit the list for easier selection, type a filter pattern into the **Name filter** field, and press the **[Enter]** key to apply the filter. Click  to the right of the text field to display information about how to construct a filter.

   Abaqus/CAE lists only those *X–Y* data objects that meet your specification.

4. From the list of *X–Y* data object names and descriptions, select one or more data objects to include in your report. For more information on selecting multiple items in dialog boxes, see *Selecting multiple items from lists and tables*.

To generate your report, configure any desired customization options on the **Setup** page, which is also in the **Report XY Data** dialog box; and click **Apply**. When you have finished, click **Cancel** to close the dialog box.

Additional information

• *Using tabular report options*

Selecting field output variables

You can select one or more field output variables to include in your tabular report.
The variables that are available consist of those you have saved to the output database. You can generate reports for the current frame or for multiple frames.

Abaqus/CAE can calculate and report values for a given variable at a variety of positions. The possible report positions are: integration point, centroid, element nodal, and unique nodal.

Element nodal and unique nodal positions both involve reporting results at the nodes of the model; however, reporting of unique nodal values produces only a single value at each node, whereas reporting of element nodal values produces one value for each element that has a contribution at that node. Only unique nodal values are provided in the case of node-based variables such as displacement. For element-based variables such as stress, extrapolation and averaging must be carried out to obtain unique nodal values. The use of averaging threshold values and the control of averaging across regional and material boundaries will affect the element nodal values reported for element based results (for more information, see *Selecting result options*).

The following table summarizes which output positions can be reported, depending on what position was used when the variable was saved to the output database:

	Report Position			
Output Database Position	**Integration point**	**Centroid**	**Element Nodal**	**Unique Nodal**
Integration Point	Yes	Yes	Yes	Yes
Centroid	No	Yes	Yes	Yes
Element Nodal	No	No	Yes	Yes
Unique Nodal	No	No	No	Yes

Contact surface variables such as CNORMF, CSHEARF, CPRESS, COPEN, CSLIP, CSHEAR1, and CSHEAR2 can be reported only at the element nodal position.

When generating the field output report for element-based field output at any output position, Abaqus/CAE writes the field values from the specified position if the same position is available in the output database.

For information on selecting the section point locations at which to report output variables, see *Selecting field output section points*.

1. Locate the options for field output reports.

   From the main menu bar, select **Report->Field Output**.

2. Choose the frames from which Abaqus/CAE reads field output data.

   - Choose **Specify** to use the current step/frame. To change the current step or frame, click . For more information, see *Selecting the results step and frame*.
   - Choose **All active steps/frames** to use all active steps/frames. Click **Active Steps/Frames** to change the steps or frames from which Abaqus/CAE reads field output data. For more information, see *Activating and deactivating steps and frames*.

3. Click the **Variable** tab to display the **Output Variables** options.
4. Click the **Position** arrow to reveal possible positions at which to list variables for your selection; then choose the desired position. Multiple variables can be *combined in a single table* only if they are reported at the same position.

The list of variables is refreshed to show only those that can be reported at the selected position.

5. Select the field output variables to include in your report using either the check box next to each variable in the list (this method is the default) or the **Edit** text field at the bottom of the page.



Note:

You must use the check box method if you want to specify an individual section point at which to report values.

To use the check box method:

1. To select a variable and all of its components, click that variable's check box.
2. To choose among the individual components of a variable, click the arrow next to that variable's check box to list its components; then click individual component check boxes to select them.

To use the edit method:

In the **Edit** text field, enter the names of the variables and components to include in your report. To be valid, a variable must be available on the output database and reportable at the current position.

To generate your report, configure the remaining options on the **Setup** page, which is also in the **Report Field Output** dialog box; then click **Apply**. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Selecting the results step and frame*
- *Using tabular report options*

Selecting field output section points

If your field output report includes section point variables, you can choose the section point at which Abaqus/CAE reports values for those variables. The default is to report values at all available section points.

1. Locate the section point options.

   From the main menu bar, select **Report->Field Output**; then click the **Variables** tab in the dialog box that appears.

2. *Use the check box method* to select section variables.
3. At the **Section point** label at the bottom of the dialog box, do one of the following:

   - click **All** to report values at all available section points, or
   - click **Single** to indicate that you will select one section point at which to report values.

     If you have selected at least one element section point variable, the section point **Settings** button becomes available.

4. To select one or more section points, click **Settings**.

   The **Field Report Section Point Settings** dialog box appears. Use this dialog box to select the desired section points. For more information, see *Selecting section point data by category*.

5. Click **OK** to apply your section point selections and to close the **Field Report Section Point Settings** dialog box.

To generate your report, configure the remaining options on the **Setup** page, which is also in the **Report Field Output** dialog box; then click **Apply**. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Selecting field output variables*
- *Using tabular report options*

Selecting options and thresholds for free body cut reports

You can create a tabular report of the resultant forces and moments in all active free body cuts in your session. You can activate or suppress individual free body cuts by toggling their **Show** check boxes in the **Free Body Cut Manager**; for more information, see *Displaying, hiding, and highlighting free body cuts*. Abaqus/CAE reports resultant force and moment data with respect to the global coordinate system, regardless of the settings selected in the **Component Resolution** options for free body cuts or for view cuts.

You can generate reports for the current frame or for multiple frames. You can also specify custom threshold values for force data and moment data in the report; forces and moments smaller than these thresholds are reported as zero in the report. Selecting appropriate thresholds enables you to produce cleaner report data; for example, replacing small report values such as `1.234E-17` with `0` makes report data easier to read. The default threshold for both force and moment values is `1E-006`.

1. Locate the options for free body cut reports.

   From the main menu bar, select **Report->Free Body Cut**.

2. Choose the frames from which Abaqus/CAE reads free body cut data.

   - Choose **Specify** to use the current step/frame. To change the current step or frame, click . For more information, see *Selecting the results step and frame*.
   - Choose **All active steps/frames** to use all active steps/frames. Click **Active Steps/Frames** to change the steps or frames from which Abaqus/CAE reads free body cut data. For more information, see *Activating and deactivating steps and frames*.

3. From the **File** options, specify the report file name. For specific instructions, see *Specifying your report file name*.
4. From the **Output Format** options, specify the presentation format for the report data. For specific instructions, see *Controlling report layout, width, format, and coordinate system*.
5. If desired, specify a threshold value for forces and a threshold value for moments from the **Thresholds** options. Abaqus/CAE will report force and moment values smaller than these thresholds as zero.

To generate your report, click **Apply**. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Selecting field output variables*
- *Using tabular report options*

Specifying your report file name

Abaqus/CAE writes your tabular data report to the file name of your choice. If you choose to write your report to an existing file, you can further specify whether the new information should be appended to the current file contents or whether the file should be overwritten. The default is to append the new information.

1. Locate the **File** options.

   ### For a report of X–Y data or field output:

   From the main menu bar, select**Report->XY** or **Report->Field Output**; then click the **Setup** tab in the dialog box that appears. The **File** options are at the top of the page.

   ### For a report of free body cuts:

   From the main menu bar, select**Report->Free Body Cut**. The **File** options are at the top of the page.

   ### For a report of probe values:

   From the main menu bar, select**Tools->Query**, then click **Probe values** in the dialog box that appears. The **Probe Values** dialog box appears.

   Move the cursor to probe the model or *X–Y* plot in the current viewport, then click mouse button 1 to store values of interest in the dialog box. When you are finished, click **Write to File**. The **Report Probe Values** dialog box appears; the **File** options are at the top of the dialog.

2. To select a file name using the standard file browser, click **Select**.

   The **File Selection** dialog box appears. Use the dialog to filter and browse existing files. For more information on using this dialog box, see *Using file selection dialog boxes*. When you are finished, click **OK** to close the dialog box.

3. Type into the **Name** field the name of the file to which your report is to be written.
4. Toggle **Append to file** to append this report to the current contents (if any) of the file you have specified. When **Append to file** is off, existing file contents will be overwritten.

To generate your report, configure the remaining options in the dialog box; then click **Apply** (for *X–Y*, field output, or free body cuts) or **OK** (for probe values reports). When you have finished, click **Cancel** to close the dialog box.

Additional information

• *Using tabular report options*

Controlling report layout, width, format, and coordinate system

Options are available to format the report layout for ease of reading and exporting.

For tabular reports of *X–Y* data, field output, or probe values you can limit the width of the table; for example, to the number of characters your printer can support.

An *X–Y* data report can include one or multiple *X–Y* data objects. Similarly, a field output report can include one or multiple variables. For both types of reports, if you include multiple items, you can choose to present each item in its own table or to combine items into a single table. If you combine multiple items into a single table, additional columns of data are formed and the width of the table increases.

To combine *X–Y* data objects into a single table, Abaqus/CAE aligns the *X*-values of the data objects. If any data objects do not have matching *X*-values, you can choose to have Abaqus/CAE compute the missing points by interpolation and extrapolation. To learn more about interpolation and extrapolation, see *Understanding X–Y data interpolation and extrapolation*.

For field output reports Abaqus/CAE creates a separate table (of a single variable or of combined variables, according to your specification) for every region in the current display group. A region is a portion of your model that has compatible materials, section properties, and element types. Abaqus/CAE identifies the region associated with each table in the report. (For more information on display groups, see *Using display groups to display subsets of your model*; for more information on regions, see *Understanding result value averaging*.)

For field output and free body cuts Abaqus/CAE enables you to generate report output in either annotated format or in comma-separated value (CSV) format. In annotated format, results are presented so that they can be read easily; in CSV format, results are intended to be exported and read in spreadsheet applications.

You can also report data from free body cuts in the global coordinate system or in the local coordinate systems for which they were defined.

1. Locate the **Output Format** options.

For a report of X–Y data or field output:

From the main menu bar, select**Report->XY** or **Report->Field Output**; then click the **Setup** tab in the dialog box that appears. The **Output Format** options are in the center of the page.

For a report of free body cuts:

From the main menu bar, select**Report->Free Body Cut**. The **Output Format** options are in the center of the page.

For a report of probe values:

From the main menu bar, select**Tools->Query**, then click **Probe values** in the dialog box that appears. The **Probe Values** dialog box appears.

Move the cursor to probe the model or *X–Y* plot in the current viewport, then click mouse button 1 to store values of interest in the dialog box. When you are finished, click **Write to File**. The **Report Probe Values** dialog box appears; the **Output Format** options are in the center of the dialog.

2. For *X–Y* data reports, select the **Layout** option of your choice:

- Select **Single table for all *X–Y* data** to combine all the selected data objects into a single table.

  Toggle **Interpolate between X values (if necessary)** to have Abaqus compute missing points. When this option is toggled off, missing points (if any) appear as “No Value” in the table.

- Select **Separate table for each *X–Y* data** to have each of the selected data objects appear in its own table.

3. For field output reports, select the format of your choice:

   - Select **Annotated format** to include the data such as step/frame and output position at the beginning of the report. Field output is written only under steps/frames where it is available.

     Toggle **Separate table for each field output variable** to have each of the selected output variables appear in its own table.
   - Select **Comma-separated values (CSV)** to generate a report that includes a single header containing all the selected field output variables and combines the field values from all regions and instances into a single table. If any of the selected field output is not available in a particular step/frame, an entry of **NoValue** appears in the table column corresponding to that field.

     When you select this format, the options to use separate tables for each field output variable, to specify the page width, to include column totals, and to include a summary of the minimum and maximum values are unavailable.
   - To include information about the local coordinate system at each value, if applicable, toggle **Element orientations** in the **Data** options area. Coordinate system columns, if specified, might not adhere to the page width.

4. For *X–Y* data reports or field output reports, select the **Page width (characters)** option of your choice:

   - Select **No limit** to allow an unlimited table width.
   - Select **Specify** to limit the table's width. Then, in the **Specify** text field, enter the maximum number of characters that can appear along the width of the table.

5. For free body cut reports, perform the following:

   - Select either **Normal annotated format** or **Comma-separated values (CSV)** as the report format.
   - Select either **Global CSYS** or **Local CSYS** as the coordinate system for the free body cuts.

To generate your report, configure the remaining options in the dialog box; then click **Apply** (for *X–Y*, field output, or free body cut reports) or **OK** (for probe values reports). When you have finished, click **Cancel** to close the dialog box.

Additional information

• *Using tabular report options*
• *Formatting report values*
• *Understanding X–Y data interpolation and extrapolation*

Sorting field output data

For tabular field output reports you can choose to sort all of the data in the table according to the values in any single column of the table and you can choose to sort in either ascending or descending order.

You control which columns Abaqus/CAE includes in the table using the **Variable** options in the **Report Field Output** dialog box. (For more information, see *Selecting field output variables*.) The table has one column for each field output variable you have selected, and additional columns to identify the origin of the field output; for example, if your table includes unique nodal quantities Abaqus/CAE provides a **Node Label** (node number) column.

Negative element labels are used in the reports to distinguish internal elements created by Abaqus/CAE from elements that you created while defining the model. If your report includes internal elements and you choose to sort by element labels, these elements will be grouped at the top or bottom of your report.

For information on sorting the data in a tabular report of probe values, see *Entering tabular data*.

1. Locate the field output report sorting options.

   From the main menu bar, select **Report->Field Output**; then click the **Setup** tab in the dialog box that appears. The **Sort by** options are in the middle of the page.
2. Click the **Sort by** arrow to reveal the list of column choices.
3. Select the column to sort by from the list.
4. Click either **Ascending** or **Descending** to choose the order of the sort.

To generate your report, configure the remaining options in the dialog box; then click **Apply**. When you have finished, click **Cancel** to close the dialog box.

Additional information

- *Using tabular report options*

Formatting report values

You can specify the number of significant digits (for *X–Y* data reports, field output reports, and probe value reports) or the total number of decimal places (for free body cut reports) to present in your tabular data report. For *X–Y* data reports, field output reports, and probe value reports, the default number of significant digits is 6; for free body cut reports, the default number of decimal places is 3. More digits give greater precision but take more space and increase the width of the tables; the number of significant digits or decimal places used determines the width of each column in the table. In addition, you can select one of the following number formats for your data values:

Automatic

Very large values and very small values are expressed in scientific format. All remaining values are expressed with no exponent and using the specified number of significant digits. This format is available for *X–Y* data reports, field output reports, and probe value reports.

Engineering

Values greater than or equal to 1000 (or values less than or equal to 0.001) are expressed as a number ranging from 1 to 999 that is multiplied by 10 to the *n*th power, where *n* is a product of 3 (for example, `20.5E+03` or `17.76E+06`). All remaining values are expressed using the specified number of significant digits.

Fixed

All values are expressed without using exponent values. When you select this format, the number 501,000 is displayed in the legend as `501000.00`. This number format is available only for free body cut reports, and your legend values may be displayed differently if you change the **Decimal places** option.

Scientific

All values are expressed as a number between 1 and 10 that is multiplied by an appropriate power of 10 (for example, `2.05E+04` or `1.776E+07`).

The default format is **Engineering** for *X–Y* data reports, field output reports, and probe value reports and **Scientific** for free body cut reports.

1. Locate the **Output Format** options.

For a report of X–Y data or field output:

From the main menu bar, select**Report->XY** or **Report->Field Output**; then click the **Setup** tab in the dialog box that appears. The **Output Format** options are in the center of the page.

For a report of free body cuts:

From the main menu bar, select**Report->Free Body Cut**. The **Output Format** options are in the center of the page.

For a report of probe values:

From the main menu bar, select**Tools->Query**, then click **Probe values** in the dialog box that appears. The **Probe Values** dialog box appears.

Move the cursor to probe the model or *X–Y* plot in the current viewport, then click mouse button 1 to store values of interest in the dialog box. When you are finished, click **Write to File**. The

Report Probe Values dialog box appears; the **Output Format** options are in the center of the dialog.

2. Specify the precision level for numerical values in the report:

   - For *X*–*Y*, field output, or probe value reports, specify this value in the **Number of significant digits** text field.
   - For free body cut reports, enter this value in the **Decimal places** text field.

   You can also click the arrows next to the applicable field until the desired number appears.

3. Click the **Number Format** button, and select the number format of your choice from the list that appears.

To generate your report, configure the remaining options in the dialog box; then click **Apply** (for *X*–*Y*, field output, or free body cut reports) or **OK** (for probe values reports). When you have finished, click **Cancel** to close the dialog box.

Additional information

• *Using tabular report options*

Reporting data values, minimums, maximums, and totals

You have the option to include minimum and maximum value summaries or column totals in reports.

For tabular reports of *X–Y* data or field output, you can include or suppress *X–Y* data or field output values, column minimum and maximum value summaries, and column totals. By default, Abaqus/CAE includes only the actual data values in reports of *X–Y* data. In reports of field output values, Abaqus/CAE includes by default the data values as well as column summaries and totals.

For tabular reports of probe values, Abaqus/CAE always includes the actual data values in the report. You can include or suppress column minimum and maximum value summaries and column totals for any columns in which such values would be meaningful.

1. Locate the **Data** or **Data Values** options.

For a report of X–Y data or field output:

From the main menu bar, select**Report->XY** or **Report->Field Output**; then click the **Setup** tab in the dialog box that appears. The **Data** options are at the bottom of the page.

For a report of probe values:

From the main menu bar, select**Tools->Query**, then click **Probe values** in the dialog box that appears. The **Probe Values** dialog box appears.

Move the cursor to probe the model or *X–Y* plot in the current viewport, then click mouse button 1 to store values of interest in the dialog box. When you are finished, click **Write to File**. The **Report Probe Values** dialog box appears; the **Data Values** options are at the bottom of the dialog box.

2. For *X–Y* data reports or field output reports, you must choose one or more of the following items to include in your report. If you generate field output reports in comma-separated value (CSV) format, the options to include column totals and to include a summary of the minimum and maximum values are unavailable.
   - Toggle **XY data** or **Field output** to include the actual *X–Y* data or field output values, respectively. Toggling this option off results in a report that consists of only the column total or min/max data you request.
   - Toggle **Column totals** to include totals for each column.
   - Toggle **Column min/max** to include a summary of the minimum and maximum value occurring in each column.

To generate your report, configure the remaining options in the dialog box; then click **Apply** (for *X–Y* reports or field output reports) or **OK** (for probe values reports). When you have finished, click **Cancel** to close the dialog box.

Additional information

• *Using tabular report options*

Customizing plot display

This chapter explains how to customize the appearance of your plots by selecting various plot state–dependent and plot state–independent options.

Using display groups to display subsets of your model and *Overlaying multiple plots* describe additional ways to customize the display of your model in the viewport.

In this section:

- *Selecting plot display options*
- *Customizing render style, translucency, and fill color*
- *Customizing element and surface edges*
- *Customizing model shape*
- *Customizing model labels*
- *Displaying multiple plot states*
- *Displaying element and surface normals*
- *Customizing the appearance of display bodies*
- *Customizing camera movement*
- *Controlling the display of model entities*
- *Controlling the display of constraints in the Visualization module*
- *Customizing general model display*

Selecting plot display options

You can customize how your model appears in undeformed, deformed, contour, symbol, and material orientation plots.
For the undeformed and deformed plot states, all of the customization options applied to the model shape are plot state–independent. That is, Abaqus/CAE applies the customizations to all undeformed and deformed plots, including those that also display contours, symbols, or material orientations. The plot state–independent options govern the render style and fill color; element and surface edges; element colors; the appearance of model labels, node symbols, and display bodies; the movement of the camera; the display of various model entities; the smoothness of curved edges; the color of elements with no results; and model sweeping and extrusion. Use the **Options** menu in the main menu bar to choose from these plot state–independent customization options.

Select **Tools->Color Code** from the main menu bar to color individual elements. Select **Options->Display Body** from the main menu bar to customize the appearance of all display bodies in your model. Select **View->View Options** from the main menu bar to customize the movement of the camera. To choose from the remaining plot state–independent options, select **View->ODB Display Options** from the main menu bar. If you choose to superimpose the deformed model shape on the undeformed shape for the same plot state, select **Options->Superimpose** from the main menu bar to customize the undeformed shape independently from the deformed shape. Like the other plot state–independent customizations, these plot options are applied to superimposed plots in all plot states.



Note:

Displaying model labels and/or symbols in the viewport can increase the amount of time needed to refresh the display for plots and animations.

In addition to the plot state–independent customization options, you can select plot state–dependent options for the contour, symbol, and material orientation plot states. The plot state–dependent options are unique to each plot type. For example, you can customize the contour color spectrum, the size and style of symbols, and the display of material orientation triad axes. If a similar customization such as the color of model edges in the common and contour plot options can be made using both independent and dependent plot options, the plot state–dependent option overrides the independent option.

In this section:

- *Saving customizations for use in subsequent sessions*
- *Available customization options*

Saving customizations for use in subsequent sessions

By default, plot display customizations persist for the current session only, but you can save customizations to a file if you want to apply them to subsequent sessions. Abaqus/CAE offers the following methods to retain plot display customizations:

Saving customizations to the display options file

Select **File->Save Display Options** from the main menu bar to save all your plot state–dependent and plot state–independent customization options as display options. Display options are stored in a file called `abaqus_2025.gpr`, and Abaqus/CAE loads the settings in this file automatically upon startup as the new default settings throughout your session. For plot state customization, saving your display options to a file enables you to store settings such as the render style, the visibility of prescribed conditions and the various types of datum geometry, and the display of mesh, element, and node labels.

For more information, see *Saving your display options settings*.

Saving customizations to an XML file, model database, or output database

Select **File->Save Session Objects** from the main menu bar to save selected plot state–dependent and plot state–independent customization options to an XML file, the model database, or an output database. When you use this method, you can save all the customization options or you can save just the customization options related to one or more plot states. For example, you can save customizations you make in the common plot options and in the contour plot options to a file.

If you save customizations to a model database or an output database, those settings are used when you open the file. When you load customizations from an XML file, these customizations become the new settings for your session. You can load settings from a file by selecting **File->Load Session Objects**.

For more information, see *Managing session objects and session options*.

Available customization options

Many options are available to customize how your model appears in undeformed, deformed, contour, symbol, and material orientation plots.

Abaqus/CAE provides the following customization options:

Render style

Render style is the style in which Abaqus/CAE displays your model. Render style is a plot state–independent option; that is, you set it once for all undeformed, deformed, contour, symbol, and material orientation plots. However, you can set a different value of render style for use with the undeformed shape in a superimposed plot. Render style choices include wireframe, filled, hidden, and lightsource-shaded. For more information, see *Customizing render style, translucency, and fill color*.

Element and surface edge style and visibility

Edge style is the line style and thickness of element and surface edges; edge visibility is the extent to which Abaqus/CAE displays these edges. Edge style and visibility options are plot state–independent, but you can set them differently for use with the undeformed shape in a superimposed plot. For more information, see *Customizing element and surface edges*.

Element and surface edge color

You can control the overall color of element and surface edges using the plot state–independent **Color & Style** options. In addition, you can customize the edge color of individual elements and surfaces using the plot state–independent **Color Code** dialog box. By default, the color code selections override the overall colors. For more information on overall edge coloring, see *Selecting overall element and surface edge color*. To learn more about individual item edge coloring, see *Customizing the display color of individual objects*. Individual item edge coloring applies only to wireframe and hidden render style plots.

Element face and surface fill color

You can control the overall fill color of element faces and surfaces using the plot state–independent **Color & Style** options. In addition, you can customize the face color of individual elements using the plot state–independent **Color Code** dialog box. By default, the color code selections override the overall colors. For more information on overall fill color, see *Selecting overall fill color*. To learn more about individual item color filling, see *Customizing the display color of individual objects*. Individual item fill coloring applies only to filled and shaded render style plots.

Model labels and node symbols

You can change the visibility and customize the color and font of element, node, and face labels and the color and style of node symbols. All these options are plot state–independent. For more information, see *Customizing model labels*.

Element and surface normals

You can choose to display arrows representing the element or surface normals in the model. The arrows are visible in all plot states, and you can control the arrow size, color, and arrowhead style. For more information, see *Displaying element and surface normals*.

Display body appearance

You can customize the render style; edge visibility, color, and style; fill color; scaling; and translucency for all display bodies in your model. These options are plot state–independent when applied to display bodies. For more information, see *Customizing the appearance of display bodies*.

Camera movement

You can select a coordinate system and match the motion of that coordinate system with the camera. You can also make the camera follow the rotation of the selected coordinate system. If you are using the camera in movie mode, you can also position the camera in the model at the origin of the coordinate system. For more information, see *Customizing camera movement*.

Entity display

You can control the display of symbols representing various model entities, including boundary conditions, connectors, coordinate systems, and point elements. Entity display options are plot state–independent. For more information, see *Controlling the display of model entities*.

Constraints display

You can control the display of analysis constraints that were applied in the Interaction module; for example, tie constraints or kinematic coupling constraints. For more information, see *Controlling the display of constraints in the Visualization module*.

General model display options

Abaqus/CAE offers several other plot state–independent general model display options. These options include:

- **Sweep/Extrude** to control the three-dimensional display of two-dimensional models. For more information, see *Sweeping and extruding your model*.
- **Mirror/Pattern** to simulate the display of a complete model when only a representative portion of the model was analyzed. For more information, see *Using mirrors and patterns to display your results*.
- **Curved Lines & Faces** to control how quadratic and cubic element edges and faces are displayed. For more information, see *Refining curved edges and faces*.
- **Elements with No Results** to control the color of elements having no results (for example, rigid surfaces) in contour plots. For more information, see *Coloring elements with no results*.
- **Idealizations** to control the display of beam profiles, shell thickness, and thermal fluid pipe elements. For more information, see *Controlling beam profile display for postprocessing* and *Controlling shell thickness display for postprocessing*, and *Controlling the display of thermal fluid pipe elements*.
- **Model Change** to control the visibility of components deactivated by model change interactions. For more information, see *Viewing removed elements*.

Customizing render style, translucency, and fill color

This section explains how to customize the render style, translucency, and fill color of your model.

In this section:

- *Choosing a render style*
- *Customizing translucency*
- *Selecting overall fill color*

Choosing a render style

Render style is the style in which Abaqus/CAE displays your model. Render style is plot state–independent; that is, you set it once for all undeformed, deformed, contour, symbol, and material orientation plots.

You can also set a second render style that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it.  Possible render styles are **Wireframe**, **Hidden**, **Filled**, and **Shaded**; these styles are shown in *Figure 1*. An explanation of these choices follows.

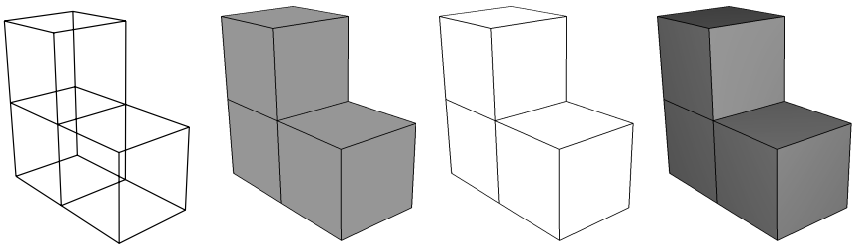


Figure 1: Model showing render style options. From left to right: the wireframe, filled, hidden, and lightsource-shaded render styles.

Wireframe

Displays model edges; both interior and exterior edges are potentially visible. Wireframe plots produce a frame-like visual effect in which model faces are not displayed.

Filled

Displays model faces “painted” in a uniform color. Filled plots produce a solid rather than frame-like appearance in which only exterior faces are visible.

Hidden

Displays a wireframe plot in which edges obscured by the model are not visible. Hidden plots produce a solid rather than frame-like appearance.

Shaded

Displays a filled plot in which a light source appears to be directed at the model. Shaded plots produce a highly three-dimensional visual effect.

1. Locate the common or superimposed **Render Style** options. The common render style always applies to the deformed shape; it applies to the undeformed shape when it is plotted individually in any plot state (undeformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose render style applies.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Basic** tab in the dialog box that appears. The **Render Style** options become available.

2. From the **Render Style** list, select the render style that you want.

   Tip: You can also select the render style for the current plot state (excluding display bodies and the undeformed shape when it is superimposed) by clicking the wireframe , hidden , filled , or shaded  icons located in the Render Style toolbar.

   Note:

   Symbol and material orientation plots in the shaded, filled, and hidden render styles are easier to view if you activate *translucency*.

3. Click **Apply** to implement your changes.

   The render style changes to reflect your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Customizing translucency*
- *Choosing a render style*

Customizing translucency

Translucency options control the degree of transparency of your model, and they apply to plots in all render styles except wireframe. Translucency is plot state–independent; that is, you set it once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the translucency that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it.

For shaded symbol and material orientation plots, symbol plot arrows and material orientation triads located in the interior of the model are easier to view if you turn on translucency.

1. Locate the common or superimposed **Translucency** options. The common translucency setting always applies to the deformed shape; it applies to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose translucency style applies.

   From the main menu bar, select **Options->Common** or **Superimpose**. Click the **Other** tab in the dialog box that appears; then click the **Translucency** tab. The **Translucency** options appear.

2. Toggle on **Apply translucency** to activate the translucency options.
3. Choose the percentage of translucency.

   Drag the slider to the percentage of translucency that you want. A value of .00 indicates transparent plotting; 1.00 indicates opaque plotting. The default translucency is .30.

4. Click **Apply** to implement your changes.

   The translucency changes to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Customizing render style, translucency, and fill color*
- *Coloring elements with no results*

Selecting overall fill color

Fill color is the color in which Abaqus/CAE displays your model faces. This color is plot state–independent; that is, you set it once for all undeformed, deformed, symbol, and material orientation plots. You can also set the fill color that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it. Fill color applies to filled and lightsource-shaded *render style* plots only; it is not applicable to contour plots.

You can choose one display color for your entire model, or you can optionally override this display color for selected items, such as particular elements. For example, you can display your model in green, with a group of elements shown in red. For more information on individual item coloring, see *Customizing the display color of individual objects*.

1. Locate the common or superimposed fill color options. The common fill color always applies to the deformed shape; it applies to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose fill color applies.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Color & Style** tab in the dialog box that appears.

   The **Fill color in filled/shaded plots** options are in the center of the **Color & Style** page.

2. Choose the fill color:

   a. Click the **Fill color in filled/shaded plots** color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Plot Options** dialog box.

3. Toggle **Allow color code selections to override options in this dialog** off to display all elements in the fill color specified above. Toggle it on to override this color using individual item color selections. The default is on. For more information, see *Customizing the display color of individual objects*.

4. Click **Apply** to implement your changes.

   The fill color changes to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing element and surface edges

This section explains how to customize the appearance of element and surface edges in your model.

In this section:

- *Controlling element and surface edge visibility*
- *Defining model feature edges*
- *Selecting overall element and surface edge color*
- *Customizing element and surface edge style*

Controlling element and surface edge visibility

You can control the extent to which Abaqus/CAE displays element and surface edges. Edge visibility is plot state–independent; that is, you set it once for all undeformed, deformed, symbol, and material orientation plots. You can also set the edge visibility that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it. The following choices are available to control edge display: **All edges**, **Exterior edges**, **Feature edges**, **Free edges**, and **No edges**. The first four options are shown in *Figure 1*.

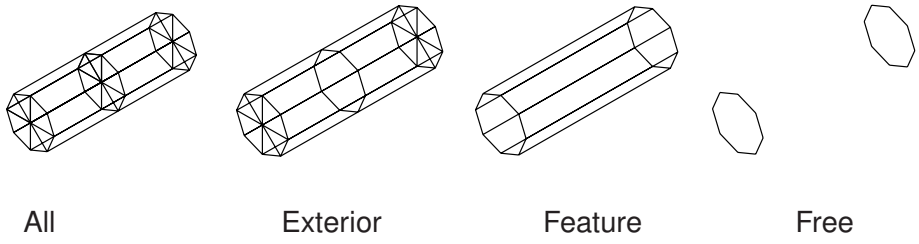


Figure 1: Model showing edge display options.

Note:

If your output database includes both homogeneous solid elements and either composite solid elements or continuum shell elements, Abaqus/CAE displays a free edge or a feature edge at the boundary between the homogeneous solid geometry and the composite geometry or continuum shell geometry.

1. Locate the common or superimposed **Visible Edges** options. The common visible edges setting always applies to the deformed shape; it applies to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose visible edges setting applies.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Basic** tab in the dialog box that appears. The **Visible Edges** options are on the right side of the **Basic** page.

2. To choose the element and surface edges you want to display, click one of the following:

All edges

Displays all element and surface edges. To see element edges on the interior of the model, you must also set the *render style* to wireframe.

Exterior edges

Displays only edges on the exterior of the model.

Feature edges

Displays only edges on the exterior of the model that are calculated to be feature edges. Feature edges lie between elements that have normals that differ by more than the “feature angle.” The feature angle is set in the **View->ODB Display Options->General** dialog box; for more information, see *Defining model feature edges*.

Free edges

Displays only edges that belong to a single element. Free edge display is particularly useful for locating potential holes or cracks in your mesh.

No edges

Suppresses the display of all edges. This choice is available only for plots in the *filled or shaded* render styles.



Note:

When elements are duplicated (i.e., share all the same corner nodes), Abaqus/CAE will not display the element edges in a free or feature edge plot.

3. Click **Apply** to implement your changes.

   The edge visibility changes according to your specification.

   Your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Defining model feature edges*
- *Selecting overall element and surface edge color*
- *Customizing element and surface edge style*
- *Controlling edge visibility*

Defining model feature edges

The common and superimposed plot options determine which model edges appear in all plot states. When you specify that only feature edges are to be visible, Abaqus/CAE determines which edges meet this criteria. Feature edges lie between elements having normals that differ by more than the “feature angle.”You can customize the feature angle. Larger angles will reduce the number of feature edges; conversely, smaller angles will cause more edges to be visible. The default is 20°. The setting of the feature angle applies to all plot states. However, all segment edges are drawn for analytical rigid surfaces regardless of the feature angle setting.

In *Figure 1* the plot on the left shows feature edges when the feature angle is set to 0°, the plot in the middle shows feature edges on the same model but with the feature angle set to 15°, and the plot on the right shows the model with the feature angle set to 30°.

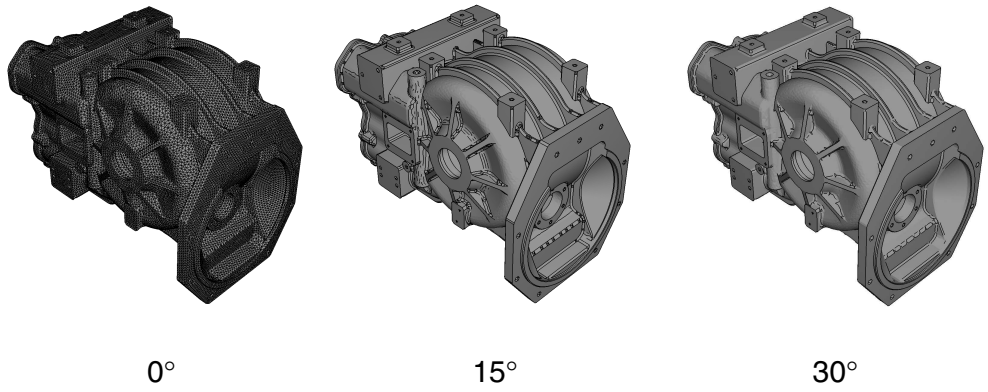


Figure 1: Plots showing feature angles of 0°, 15°, and 30°.

1. Locate the **Feature Angle** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **General** tab in the dialog box that appears. The **Feature Angle** options are at the bottom of the **General** page.
2. Drag the **Feature Angle** slider to the desired feature angle.
3. Click **Apply** to implement your changes.

   Feature edges for all plot states change to reflect your feature angle specification. By default, your feature angle specification is saved for the duration of the session. If you want to retain this change for subsequent sessions, save it to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Controlling element and surface edge visibility*
- *Defining mesh feature edges*

Selecting overall element and surface edge color

You can customize the color in which Abaqus/CAE displays element and surface edges. With the exception of contour plots, edge color is plot state–independent; that is, you set it once for all undeformed, deformed, symbol, and material orientation plots. You can choose a different edge color for certain *render styles*, and you can set the edge color that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it. For contour plots you can choose a different edge color for certain *contour types*.

You can choose one display color for your entire model, or you can optionally override this display color for selected items, such as particular elements. For example, you can display your model in green, with a group of elements shown in red. For more information on individual item coloring, see *Customizing the display color of individual objects*.

1. Locate the edge color options (common, superimposed, or contour) that you want to customize.
   - For all plots except superimposed and contour plots:

     From the main menu bar, select**Options->Common**; then click the **Color & Style** tab in the dialog box that appears.

     The **Color** options are at the top of the **Color & Style** page.
   - For contour plots:

     From the main menu bar, select**Options->Contour**; then click the **Color & Style** tab in the dialog box that appears. Click the **Model Edges** tab.
   - For the undeformed shape in a superimposed plot:

     From the main menu bar, select**Options->Superimpose**; then click the **Color & Style** tab in the dialog box that appears.

     The **Color** options are at the top of the **Color & Style** page.

2. Choose the color of the element and surface edges for wireframe and hidden render style plots and line-type contours. (For all of these plots Abaqus/CAE does not display model faces.)
   a. Click the color sample , which is labeled either **Edges in wireframe/hidden plots** or **In line plots**.

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Plot Options** dialog box.

3. Choose the color of the element and surface edges for filled and shaded render style plots and banded- and quilt-type contour plots. (For all of these plots Abaqus/CAE displays model faces.)
   a. Click the color sample , which is labeled either **Edges in filled/shaded plots** or **In banded/quilt plots**.

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Plot Options** dialog box.

4. Toggle **Allow color code selections to override options in this dialog** off to display all edges in the color specified above. Toggle it on to override this color using color code selections. The default is on. This option is not available for contour plots.
5. Click **Apply** to implement your changes.

   The edge color changes according to your specification. If you specify an edge color for wireframe, hidden, or line plots that is the same color as the viewport background, Abaqus/CAE draws the edges in a contrasting color so that your plot will not disappear. Edges drawn in the viewport background color in filled or shaded plots may obscure the corresponding model faces if you zoom out or if the faces are so small that the edges begin to overlap.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Controlling element and surface edge visibility*
- *Customizing element and surface edge style*
- *Choosing background colors*

Customizing element and surface edge style

You can customize the style and thickness in which Abaqus/CAE displays element and surface edges.

For example, *Figure 1* shows a plot with default element edges on the left and customized edges on the right. Edge style options are plot state–independent; that is, you set them once for all undeformed, deformed, symbol, and material orientation plots. You can also set the edge style options that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it.

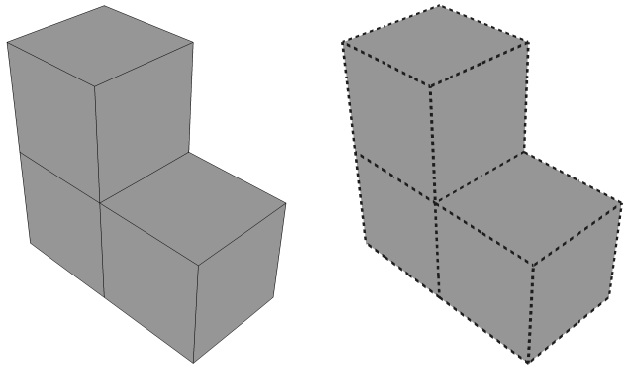


Figure 1: Models showing default and customized element edges.

1. Locate the common or superimposed **Edge Attributes** options. The common edge attributes always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose edge attributes apply.

   From the main menu bar, select **Options->Common** or **Superimpose**. Click the **Other** tab in the dialog box that appears; then click the **Color & Style** tab in the dialog box that appears.

   The **Edge Attributes** options are at the bottom of the **Color & Style** page.

2. Choose the style of the element and surface edges:
   a. Click the **Style** button to reveal the edge style options.
   b. Click the edge style you want.

      The specified edge style appears on the **Style** button.

3. Choose the thickness of the element and surface edges:
   a. Click the **Thickness** button to reveal the edge thickness options.
   b. Click the edge thickness you want.

      The specified edge thickness appears on the **Thickness** button.

4. Click **Apply** to implement your changes.

   The element and surface edge style changes according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Controlling element and surface edge visibility*
- *Selecting overall element and surface edge color*

Customizing model shape

This section explains how to customize the shape of your model.

In this section:

- *Scaling deformations*
- *Scaling coordinates and shrinking the model*

Scaling deformations

Deformations are the values of a deformed field output variable; for example, displacement or velocity. Abaqus/CAE computes the deformed shape by applying the deformations to the undeformed nodal coordinates. You can scale the deformations to magnify, reduce, or otherwise distort the deformed model shape. For example, *Figure 1* displays a deformed shape contour plot on the left and the same plot with the deformation magnified 15 times on the right.

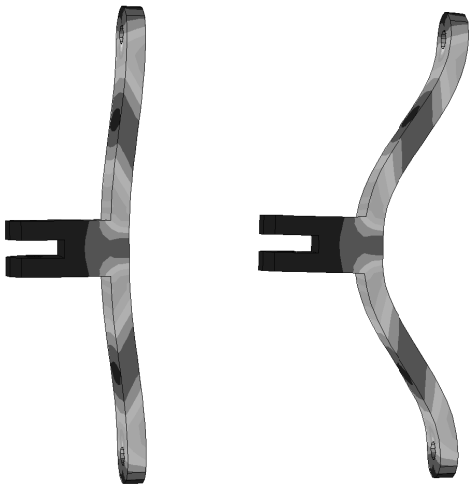


Figure 1: Contour plots showing default and magnified deformation values.

Deformation scaling is plot state–independent; that is, you set it once for the deformed shape in all deformed, contour, symbol, and material orientation plots. The default scaling is a uniform factor of 1.00 for large-displacement analyses. For small-deformation analyses—for example, a perturbation analysis—Abaqus/CAE scales the deformation such that the maximum deformation is 10% of the largest model dimension.

1. From the main menu bar, select **Options->Common**. Click the **Basic** tab in the dialog box that appears. The **Deformation Scale Factor** options are in the lower left corner of the **Basic** page.
2. Choose one of the following scale factor options:
   - Click **Auto-compute** to request that Abaqus automatically compute and uniformly apply a single scale factor to all *X*-, *Y*-, and *Z*-components of deformation values.
   - Click **Uniform** to specify and uniformly apply a single scale factor to all *X*-, *Y*-, and *Z*-components of deformation values.

     When **Uniform** is on, a **Value** specification box becomes available. Click the **Value** box, and enter your scale factor.
   - Click **Nonuniform** to specify individual scale factors to be applied to the *X*-, *Y*-, and *Z*-components of deformation values.

     When **Nonuniform** is on, **X**-, **Y**-, and **Z**-component scale factor specification boxes become available. For each component you want to scale, click the component (**X**, **Y**, or **Z**) scale factor box and enter your scale factor.
3. Click **Apply** to implement your changes.

   The deformed shape changes according to your specification. The *state block*, if active, changes to show the current deformation scale factors.

Your changes are saved for the duration of the session and will affect all subsequent deformed shape plots.

By default, your changes are saved for the duration of the session and will affect all subsequent deformed shape plots. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Scaling coordinates and shrinking the model*

Scaling coordinates and shrinking the model

You can magnify, reduce, or otherwise distort the shape of your model by scaling, and you can shrink each element about its centroid. Scaling modifies all nodal coordinates in each of the *X*-, *Y*-, and *Z*-directions. Shrinking reduces each element in size uniformly about its centroid. For example, the elements shown in *Figure 1* have been shrunk by a factor of 0.40.

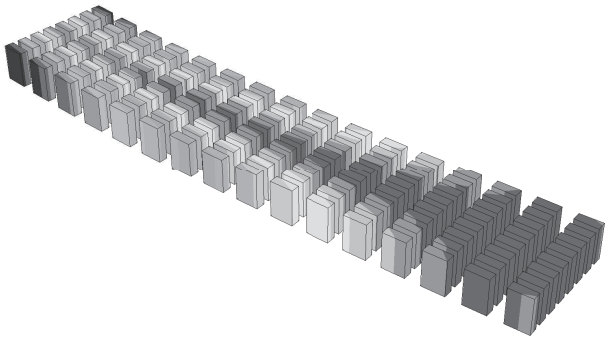


Figure 1: Elements showing the effect of shrinking.

Scaling and shrinking options are plot state–independent; that is, you set them once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the scaling and shrinking options that Abaqus/CAE will apply to the undeformed shape when the deformed shape is superimposed on it.

1. Locate the common or superimposed **Scaling** options. The common scaling options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose scaling options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Other** tab in the dialog box that appears. Click the **Scaling** tab; the **Scaling** options appear.

2. Toggle **Shrink elements** to request or suppress shrinking each element about its centroid.
   a. When **Shrink elements** is on, the shrink **Factor** slider becomes available.
   b. Drag the **Factor** slider to the desired shrink factor.

      A value of 0.00 indicates no shrinking. A value of 0.90 shrinks all elements to dots.

3. Toggle **Scale coordinates** to request or suppress model scaling by nodal coordinate *X*-, *Y*-, and *Z*-directions.
   a. When **Scale coordinates** is on, scale factors become available. Click the **X**, **Y**, and **Z** boxes to enter scale factors for the *X*, *Y*, and *Z* nodal coordinates, respectively.

4. Click **Apply** to implement your changes.

   The model display changes according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Scaling deformations*

Customizing model labels

This section explains how to customize the appearance of element labels, face labels, node labels, and node symbols for undeformed, deformed, contour, symbol, and material orientation plots.

In this section:

- *Setting the label font*
- *Customizing element labels*
- *Customizing face labels*
- *Customizing node labels*
- *Customizing node symbols*
- *Displaying nodal-averaged orientations*

Setting the label font

Abaqus/CAE uses a single font to display all element, face, and node labels. You can set the properties (family, size, boldness, italics) of this font for all plot states, and you can set them separately for use with the undeformed shape in a superimposed plot.

1. Locate the common or superimposed label font options. The common font options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose label font options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Labels** tab in the dialog box that appears. The font options are at the top of the **Labels** page.

2. Click **Set Font for All Model Labels**.

   The **Select Font** dialog box appears.

3. Use the **Select Font** dialog box to set the font properties that you want.
4. When you have set the desired font properties, click **OK**.

   The **Select Font** dialog box closes.

5. Click **Apply** to implement your changes.

   The element, face, and node labels change according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Customizing element labels*
- *Customizing face labels*
- *Customizing node labels*

Customizing element labels

Element labels are numeric labels (element numbers) that identify each element. For example, the elements are labeled in *Figure 1*.

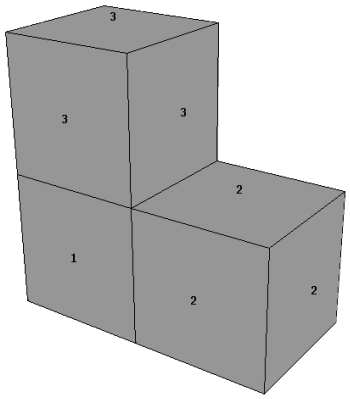


Figure 1: Model showing element labels.

Element labels are plot state–independent; that is, you set them once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the element labels that Abaqus/CAE will display on the undeformed shape when the deformed shape is superimposed on it. Toggle **Show element labels** to display or suppress element labels and to choose their color.

1. Locate the common or superimposed element label options. The common element label options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose element label options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Labels** tab in the dialog box that appears.

2. Toggle **Show element labels** to display or suppress numeric element labels.

   When **Show element labels** is on, the element label color options become available.

3. Choose the color of the element labels:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color

4. Click **Apply** to implement your changes.

   The labels change according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to

retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Setting the label font*

Customizing face labels

Face labels identify the order of faces within each element. For example, the faces are labeled in *Figure 1*.

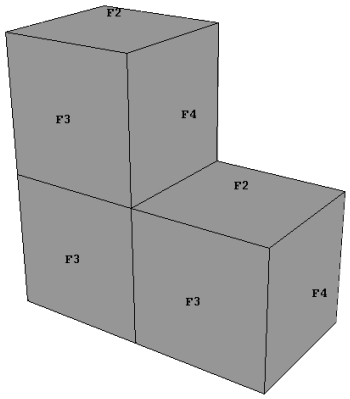


Figure 1: Model showing face labels.

Face labels are plot state–independent; that is, you set them once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the face labels that Abaqus/CAE will display on the undeformed shape when the deformed shape is superimposed on it. Toggle **Show face labels** to display or suppress element labels and to choose their color.

1. Locate the common or superimposed face label options. The common face label options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose face label options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Labels** tab in the dialog box that appears.

2. Toggle **Show face labels** to display or suppress numeric face labels.

   When **Show face labels** is on, the face label color options become available.

3. Choose the color of the face labels:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

4. Click **Apply** to implement your changes.

   The labels change according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to

retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Setting the label font*

Customizing node labels

Node labels are numeric labels (node numbers) identifying each node. For example, the nodes are labeled in *Figure 1*.

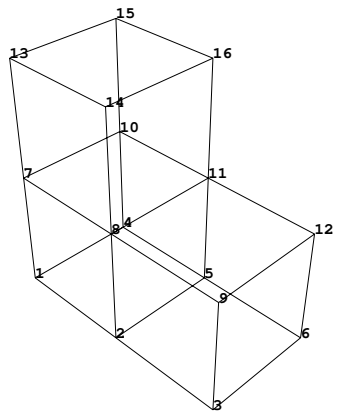


Figure 1: Model showing node labels.

Node labels are plot state–independent; that is, you set them once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the node labels that Abaqus/CAE will display on the undeformed shape when the deformed shape is superimposed on it. Toggle **Show node labels** to display or suppress node labels and to choose their color.

1. Locate the common or superimposed node label options. The common node label options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose node label options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Labels** tab in the dialog box that appears.

2. Toggle **Show node labels** to display or suppress numeric node labels.

   When **Show node labels** is on, the node label color options become available.

3. Choose the color of the node labels:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

4. Click **Apply** to implement your changes.

   The labels change according to your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to

retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Setting the label font*

Customizing node symbols

Node symbols (circles, squares, triangles, etc.) identify the location of each node. For example, filled circles are used to symbolize the nodes shown in *Figure 1*.

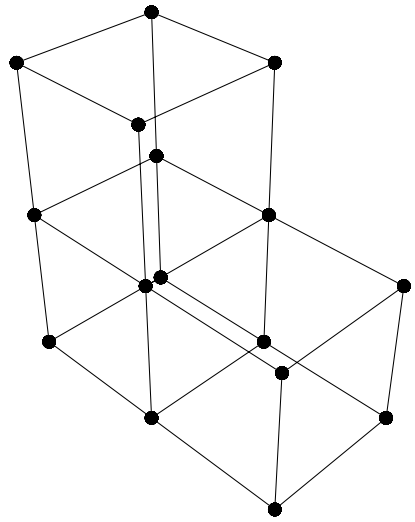


Figure 1: Model showing node symbols.

Node symbols are plot state–independent; that is, you set them once for all undeformed, deformed, contour, symbol, and material orientation plots. You can also set the node symbols that Abaqus/CAE will display on the undeformed shape when the deformed shape is superimposed on it. Toggle **Show node symbols** to display or suppress node labels and to choose their color.

1. Locate the common or superimposed node symbol options. The common node symbol options always apply to the deformed shape; they apply to the undeformed shape when it is plotted individually in any plot state (undeformed, deformed, contour, symbol, or material orientation). When the undeformed shape is plotted with the deformed shape, the superimpose node symbol options apply.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Labels** tab in the dialog box that appears.

2. Toggle **Show node symbols** to display or suppress node symbols.

   When **Show node symbols** is on, the node symbol color, type, and size options become available.

3. Choose the color of the node symbols:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color. The new color does not appear in the viewport until you click **OK** or **Apply** in the **Plot Options** dialog box.

4. Choose the type (circle, square, triangle, etc.) of the node symbols:

   a. Click the **Symbol** button to reveal the node symbol type options.

b. Click the symbol type you want.

The specified symbol type appears on the **Symbol** button.

5. Choose the size of the node symbols:

a. Click the **Size** button to reveal the node symbol size options.

b. Click **Small**, **Medium**, or **Large**.

The specified symbol size appears on the **Size** button.

6. Click **Apply** to implement your changes.

The symbols change according to your specification.

By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Customizing node labels*

Displaying nodal-averaged orientations

For line- and banded-type contour plots, element results are extrapolated and conditionally averaged from the element centroids or integration points to the nodes to create the plots. You can use the **Result Options** dialog box to control how Abaqus/CAE applies averaging to a model. One option allows you to compute invariants (scalars) and extract components after averaging the results at each node; when you choose to compute the results in this order, Abaqus/CAE applies the vector or tensor orientation from one of the adjacent elements as the orientation for each node. Under the **Other** tab of the **Contour Plot Options** dialog box, toggle **Show nodal-averaged vector/tensor orientations when computing scalars after averaging** to display or suppress the averaged orientations on the contour plot. When this option is enabled, Abaqus/CAE draws a triad at each node indicating the nodal-averaged orientation for the plotted vector or tensor. By default, the 1-axis appears in teal, the 2-axis appears in yellow, and the 3-axis appears in red. You can use the same procedures described in *Customizing material orientation plot triads*, to customize the appearance of these triads.

If you choose to compute scalars prior to averaging the results at the nodes, orientations are not averaged and do not appear in your contour plot.

Additional information

- *Selecting result options*

Displaying multiple plot states

In addition to superimposing the undeformed and deformed shape for a single plot state such as contour plots, you can choose to display multiple plot states in a viewport.

By toggling on **Allow Multiple Plot States**  in the toolbox, you can choose any combination of the undeformed and deformed plot shapes with contours, symbols, and material orientations for the selected step and frame of the current output database. Abaqus/CAE displays the deformed plots using the **Common Plot Options** and the undeformed plots using the **Superimposed Plot Options** in combination with the other plot state–independent and dependent options for each plot state.

If you toggle off **Allow Multiple Plot States**, Abaqus/CAE uses the following order in combination with the last plot states that were displayed to select the single plot state to display:

- Undeformed
- Deformed
- Contours
- Symbols
- Material Orientations

For example, if you display contours and material orientations, Abaqus/CAE retains the contour plot when you toggle off **Allow Multiple Plot States**. This is the same order that the plot states are listed in the **Plot** menu and displayed in the toolbox.

To display more complex plot combinations, you must use an **Overlay Plot**. Overlay plots allow you to display animations or *X*–*Y* plots along with the stationary plot states listed above. You can also use overlay plots to plot results from multiple steps, frames, or output databases. For more information, see *Overlaying multiple plots*.

Additional information

- *Using common plot options*
- *Superimposing deformed and undeformed model plots*

Displaying element and surface normals

When you create an undeformed or deformed plot, you can choose to display arrows in the plot that indicate the directions of the element or surface normals. You can display normal directions for beams, pipes, frame elements, contact surfaces, membranes, rigid elements, rigid surfaces, shells, three-dimensional solids, and trusses. (For information about normal directions for a particular type of element or surface, see *Abaqus Elements Guide*.)

1. Locate the common or superimposed normals options. The common normals options apply to the undeformed and the deformed shape when they are plotted individually. The superimpose normals options apply to the undeformed shape when it is plotted with the deformed shape.

   From the main menu bar, select **Options->Common** or **Superimpose**; then click the **Normals** tab in the dialog box that appears.

2. Toggle on **Show normals**.

   The **Normals** options become available.

3. Toggle on **On elements** to display the normals on elements in your model or **On surfaces** to display the normals on surfaces in your model.
4. In the area of the dialog box labeled **Colors**, select the colors in which you want the different types of normals to appear.
5. In the area of the dialog box labeled **Style**:
   a. Click the arrow next to the **Length** field, and select the arrow length of your choice. The default selection is **Medium**.
   b. Click the arrow next to the **Line Thickness** field, and select the arrow thickness of your choice.
   c. Click the arrow next to the **Arrowhead** field, and select the arrowhead style of your choice.
6. Click **Apply** to implement your changes.

   Arrows appear in the viewport that indicate the normal directions of the elements or surfaces in the model. In general, the arrows appear at the centroid of exterior element faces.

Customizing the appearance of display bodies

A display body is a part instance that does not take part in the analysis but is visible during postprocessing. You can customize the appearance of such part instances in your model using the **Display Body Options** in the Visualization module. The options available for display bodies are similar to those available for all of the plot states in Abaqus/CAE: render style; edge visibility, color, and style; fill color; scaling; and translucency can all be customized. Similar to the common plot options, display body options are plot state–independent; i.e., the options applied to display bodies are reflected in all plot states. Display body options apply to all display bodies in your model; to customize the appearance of individual display bodies, you can create display groups based on part instances (see *Creating or editing a display group*). Individual item coloring can also be applied to individual display body part instances to override the edge and/or fill colors specified as general display body options; see *Color coding geometry and mesh elements* for more information.

To locate the display body options, select **Options->Display Body** from the main menu bar. Click the following tabs to customize the appearance of all display bodies in the current viewport:

- **Basic**: Choose render style (*Choosing a render style*) and edge visibility (*Controlling element and surface edge visibility*).

  Note:

  The render style icons located in the Render Style toolbar do not apply to display bodies in any plot state. In addition, by default, only free edges are shown for display bodies while all exterior edges are shown for other model components.

- **Color & Style**: Control model edge color (*Selecting overall element and surface edge color*), model edge style (*Customizing element and surface edge style*), and model fill color (*Selecting overall fill color*).
- **Other**: The **Other** page contains the following tabs:
  - **Scaling**: Control model scaling and shrinking (*Scaling coordinates and shrinking the model*).
  - **Translucency**: Control shaded render style translucency (*Customizing translucency*).

Additional information

- *Defining display body constraints*
- *Display bodies*

Customizing camera movement

You can customize two groups of camera options in your model by using the **View Options** in the Visualization module, as shown in *Figure 1*.

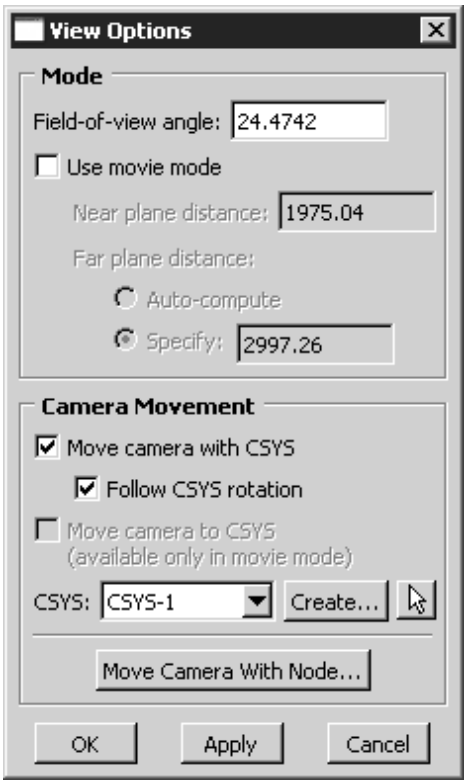


Figure 1: The View Options dialog box.

The **Mode** options are available in all Abaqus/CAE modules except the Sketch module; the **Mode** options are discussed in *Manipulating the view and controlling perspective* and are saved only for the current session. The **Camera Movement** options are available only in the Visualization module. They control the position and movement of the camera with respect to a selected local coordinate system. You can save the settings of the **Camera Movement** options by selecting **File->Save Options** from the main menu bar.

To locate the view options, select **View->View Options** from the main menu bar. To use the **Camera Movement** options, you must first toggle on **Move camera with CSYS**, then select a local coordinate system using one of the following methods:

- Select a coordinate system from the list provided in the dialog box,
- Click **Create** to create a new local coordinate system,
- Click  to pick a local coordinate system from the viewport, or
- Click **Move Camera With Node** or use the  button in the context bar to create a rectangular coordinate system following a single node.



Note:

To reset the camera to follow the global coordinate system, either toggle off **Move camera with CSYS** or click the  button in the context bar.

You can select from the following camera movement options:

Move camera with CSYS

Select this option to fix the origin of the selected coordinate system in the viewport. If you create an animation or change the ODB frame, Abaqus/CAE moves all objects in the viewport relative to the position of the selected local coordinate system. If the selected local coordinate system is not a moving coordinate system, you will not observe any difference from the default settings that fix the camera to the (stationary) global coordinate system.

If you select **Move camera with CSYS**, it is recommended that you pause any animation in the current viewport before attempting to manipulate the view.

Follow CSYS rotation

Select this option to fix both the position and the rotation of the local coordinate system in the viewport. All model rotations are transformed so that the model appears to rotate about the selected coordinate system. This option is available only when **Move camera with CSYS** is toggled on.

Move camera to CSYS

This option is available only when **Use movie mode** and **Move camera with CSYS** are toggled on. **Use movie mode** allows the camera to move into and through the model instead of viewing the model from a distance (for more information, see *Camera modes and view terminology*). When you apply **Move camera to CSYS**, the camera moves to the origin of the selected coordinate system. Abaqus/CAE toggles this option off immediately after you click **Apply** in the **View Options** dialog box to avoid conflicts with the view manipulation tools, some of which also reposition the camera.



Note: When you apply **Move camera to CSYS**, Abaqus/CAE may display a blank view in the viewport. The most common cause for this blank view is the **Near plane distance** setting; apply a smaller **Near plane distance** to view portions of the model that are close to the camera.



Warning: If you pan, rotate, or magnify the view in the Visualization module, it is recommended that you first pause any animations in the viewport. Since animations and view manipulations both affect the view, their instructions to Abaqus/CAE may conflict with each other and create undesirable results.

Additional information

- *Understanding camera modes and view options*
- *Creating coordinate systems during postprocessing*

Controlling the display of model entities

The **Entity Display** options in the **ODB Display Options** dialog box allow you to control the display of symbols. The displayed symbols represent the following entities in your model:

- boundary conditions applied during the analysis,
- connectors used in the model (the highlighting of endpoints, the display of local orientation axes associated with each connector, and the display of connector type labels can be controlled individually),
- coordinate systems written to the current output database or created in the Visualization module during the current session, and
- point elements, including reference points, mass and inertia elements, springs and dashpots, spot welds and distributing coupling (DCOUP*) elements, tracer particles, continuum particle elements, discrete particle elements, and rendering of particle element edges.

Model entity display is plot state–independent. Abaqus/CAE displays model entity symbols for the current *step and frame* of your output database. You can customize the appearance of the model entity symbols in the Visualization module only by changing their size.

For information on the symbols representing boundary conditions, see *Understanding symbols that represent prescribed conditions*; for information on the symbols representing connectors, see *Understanding symbols that represent interactions, constraints, and connectors*. For information on controlling the display of these entities in assembly-related modules, see *Controlling the display of attributes*.



Tip: You can also control the display of connectors using display groups; each connector is listed as an element set. See *Creating or editing a display group* for more information on display groups.

The render particle element edges option allows the display of symbols for particle element edges. You can use Wireframe, Hidden, and Filled render style options for particle elements.

Only session coordinate systems and reference points are displayed by default. The display of model entities can affect performance significantly. To maximize the performance of Abaqus/CAE, be selective about the entities you choose to display.

1. From the main menu bar, select **View->ODB Display Options**. Click the **Entity Display** tab in the dialog box that appears.

   The entity display options become available.

2. Toggle the entities that you want to display. Only the entities that you select (if they have been defined for the current model) will appear in the viewport.
3. If desired, drag the **Symbol size** slider left or right to modify the size of the symbols in the viewport.
4. Click **Apply** to implement your changes.

   The plot in the current viewport changes to reflect your specification.

   By default, your changes are saved for the duration of the session and will affect all subsequent plots in the current viewport and in any new viewports created from the current viewport. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Creating boundary conditions*
- *Understanding constraints*
- *Understanding connectors*
- *Creating coordinate systems during postprocessing*

- *Why are datum coordinate systems so important?*

Controlling the display of constraints in the Visualization module

In the Visualization module you can selectively control the display of analysis constraints using display groups and the **ODB Display Options** dialog box. These controls are useful when debugging models that contain large numbers of constraints. By turning the display of some constraints on and off, you can more easily understand complex models.

There are two levels of controls for showing or hiding constraints:

- You must first add the desired constraints to a display group for them to be visible in the viewport. You can create and edit display groups to include or exclude different constraints. See *Creating or editing a display group*, for more information about display groups.
- Different types of constraints can be selected or hidden in the **Constraints** tab of the **ODB Display Options** dialog box. From the main menu bar, select**View->ODB Display Options** to access these options.

The following types of constraints can be displayed or hidden:

- tie constraints
- rigid body constraints (ties and pins only)
- shell-to-solid couplings
- distributing couplings
- kinematic couplings
- multi-point constraints



Note:

These types of constraints are created in the Interaction module, not the Assembly module. See *Understanding constraints* for more information.

Table 1 shows how each type of constraint is displayed by Abaqus/CAE.

Table 1: How constraints are shown in the viewport.

Constraint type	Display in viewport
Tie	Spidering of red lines drawn between the nodes on the main and secondary surfaces of the constraint.
Rigid body	The rigid body regions (elements, nodes, or surfaces) are highlighted in the viewport.
Shell-to-solid coupling	Spidering of yellow lines drawn between the nodes on the shell edge and the solid face.
Distributing coupling	Spidering of orange lines drawn between the control point and the coupling points (nodes) in the constraint region.
Kinematic coupling	Spidering of purple lines drawn between the control point and the coupling points (nodes) in the constraint region.
Multi-point constraint (Abaqus/Explicit only)	The nodes comprising the multi-point constraint are connected with yellow lines.

You can change the default colors of the constraint spider lines in the **Color Code** dialog box. For more information, see *Coloring constraints in the Visualization module*.

Additional information

- *Understanding constraints*
- *Creating or editing a display group*

Customizing general model display

This section explains several of the general model display options.

In this section:

- *Sweeping and extruding your model*
- *Using mirrors and patterns to display your results*
- *Refining curved edges and faces*
- *Coloring elements with no results*
- *Controlling beam profile display for postprocessing*
- *Controlling shell thickness display for postprocessing*
- *Controlling the display of thermal fluid pipe elements*
- *Viewing removed elements*

Sweeping and extruding your model

You can display axisymmetric models as planar, two-dimensional shapes or "sweep" the models through a specified angle, producing a three-dimensional visual effect.
Similarly, you can view just the modeled sector of a cyclic symmetric structure or "sweep" the sector to see a specified portion of the entire model. (For more information on cyclic symmetric structures, see *Analysis of Models that Exhibit Cyclic Symmetry*.) In addition, you can view two-dimensional models (or three-dimensional cylindrical rigid surfaces) as planar or extrude them to a specified depth, producing a three-dimensional visual effect. Sweeping and extruding are particularly useful for displaying contour plots of two-dimensional elements and contact surfaces.

Axisymmetric analytical rigid surfaces can be rotated about an axis (swept), as can the following elements: ACAX*n*, CAX*n*, CAXA*n*, CGAX*n*, DCAX*n*, DCCAX*n*, DSAX*n*, FAX*n*, MAX*n*, MGAX*n*, RAX*n*, SAX*n*, and SAXA*n*. Models that contain three-dimensional axisymmetric analytical rigid surfaces or CAXA elements are swept by default (the default start angle, end angle, and number of sectors vary depending on the model type). If the model contains both analytical rigid surfaces and CAXA elements, only the CAXA elements are swept by default. When you display boundary conditions for swept axisymmetric elements, symbols appear on only the original nodes and not on the swept nodes.

Cyclic symmetric models are not swept by default. When you display contours on a swept cyclic symmetric model, only nodal quantities can be contoured on the swept sectors; scalar, vector, and tensor quantities at integration points can be contoured only on the original sector.

You can extrude analytical rigid surfaces and all of the planar, two-dimensional solid elements in the Abaqus library along the *Z*-direction. For a list of these elements, see *Two-Dimensional Solid Element Library*.

Figure 1 shows an axisymmetric planar model in a planar view (left) and the same model with an axisymmetric sweep angle of 90° and 10 sweep sectors (right).

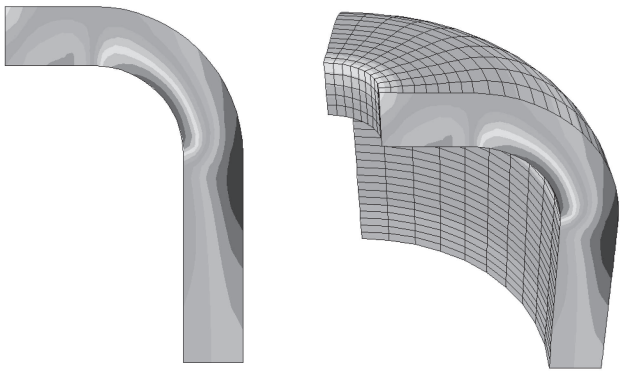


Figure 1: Axisymmetric model with planar and swept display.

Figure 2 shows the original modeled sector of a cyclic symmetric fan (left) and sectors 1–4 of the swept model (right).

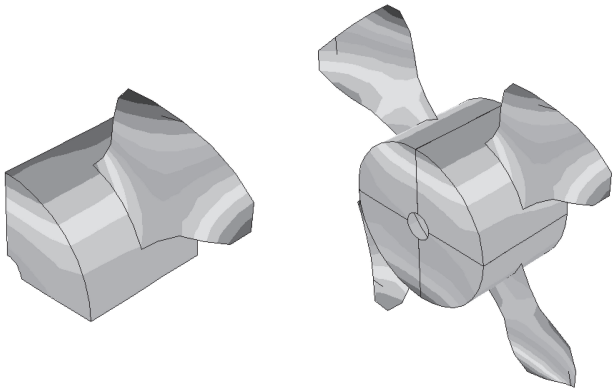


Figure 2: Cyclic symmetric model: single sector and swept view of multiple sectors.

Figure 3 shows a two-dimensional planar model in a planar view (left) and the same model displayed with an extrusion depth of 0.1 (right).

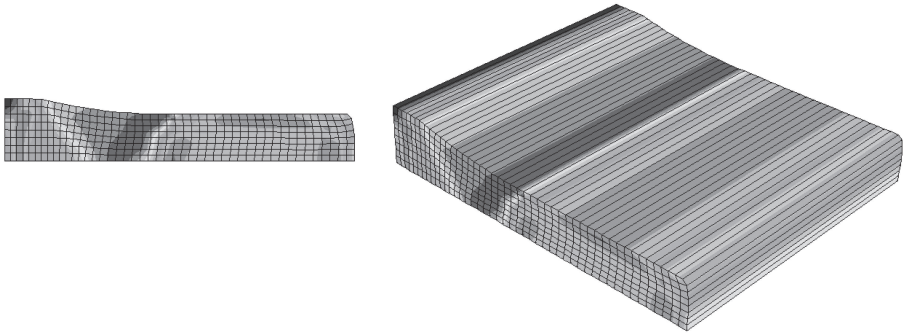


Figure 3: Two-dimensional model with planar and extruded display.

Additional information

- *Using mirrors and patterns to display your results*
- *Refining curved edges and faces*

Sweep elements in your axisymmetric model

1. Locate the **General Sweep** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Sweep/Extrude** tab in the dialog box that appears. The **General Sweep** options are in the upper portion of the page.

   Note:

   If there are no elements or analytical rigid surfaces in your model that can be swept, the **General Sweep** options will not be available.

2. Toggle **Sweep elements**.
3. Specify the sweep angle by typing a number (in degrees) in the **Sweep from** field (the default is 0°) and a number (in degrees) in the **To** field (the default is 180° for all models).

   Abaqus/CAE sweeps the two-dimensional model counterclockwise about the axis of rotation from the first specified angle to the second.

4. Specify the number of segments to use along the circumferential direction when sweeping by typing a positive integer in the **Number of segments** field or by clicking on the arrows next to the field. The default number of segments used varies depending on the model type.

   The angle between the segments is updated automatically in the dialog box. As you increase the number of segments, curves in the model appear smoother (the angle between them decreases). However, a smaller number of segments plots faster.

5. Click **Apply** to implement your changes.

   Your sweep specifications are reflected in all plot states and, by default, they are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

   If a contour plot is displayed, the contours appear on all swept faces.

   If a symbol plot is displayed, the model is swept as specified, but the symbols appear only on the original planar model faces.

Sweep analytical rigid surfaces in your axisymmetric model

1. Locate the **General Sweep** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Sweep/Extrude** tab in the dialog box that appears. The **General Sweep** options are in the upper portion of the page.

   Note:

   If there are no elements or analytical rigid surfaces in your model that can be swept, the **General Sweep** options will not be available.

2. Toggle on **Sweep analytical rigid surfaces** if it is not currently active.
3. Specify the sweep angle by typing a number (in degrees) in the **Sweep from** field and a number (in degrees) in the **To** field (the default value is 180° when you first open an axisymmetric model and 360° when you first open a three-dimensional model).

   Abaqus/CAE sweeps the analytical rigid surface counterclockwise about the axis of rotation from the first specified angle to the second.

4. Specify the number of segments to use along the circumferential direction when sweeping analytical rigid surfaces by typing a positive integer in the **Number of segments** field or by clicking on the arrows next to the field. By default, Abaqus/CAE uses 10 segments for axisymmetric models and 20 segments for three-dimensional models.

   The angle between the segments is updated automatically in the dialog box. As you increase the number of segments, curves in the model appear smoother (the angle between them decreases). However, a smaller number of segments plots faster.

5. Click **Apply** to implement your changes.

   Your sweep specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Sweep your cyclic symmetric model

1. Locate the **Sector Sweep** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Sweep/Extrude** tab in the dialog box that appears. The **Sector Sweep** options are in the center portion of the page.



Note:

If there are no cyclic symmetry elements in your model, the **Sector Sweep** options will not be available.

2. Toggle **Sweep cyclic symmetry sectors**.
3. Click the arrow next to the **Sector selection** field to choose from the following available sweep methods:

By Number

This is the default sweep method. The total number of sectors in the model is given in the dialog box. You can specify which of these sectors should be displayed by typing positive integers between 1 and the total number of sectors (delimited by commas) in the **Sectors** field. The sectors are numbered counterclockwise from the original sector.

Abaqus/CAE displays only the specified sectors (whether or not they are adjacent).

By Angle

The sector angle (the angle between each sector) is given in the dialog box. You can specify the sweep angle in increments of the sector angle by typing a number (in degrees) in the **Sweep from** field (the default is 0°) and a number (in degrees) in the **To** field (the default is the sector angle). Both numbers should be divisible by the sector angle.



Tip: You can also use the arrows next to the text fields to specify the angles.

Abaqus/CAE sweeps the two-dimensional model about the axis of rotation by the number of degrees you specify. If you enter a sweep angle that is not divisible by the sector angle, Abaqus/CAE rounds up to the next number that is.

All Sectors

If you choose this sweep method, all sectors will be displayed; in other words, the model will be swept 360° about the axis of rotation.

4. Click **Apply** to implement your changes.

   Your sweep specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

   If a contour plot is displayed, the contours of nodal quantities appear on all swept faces; contours of scalar, vector, and tensor quantities at integration points appear only on the original sector.

   If a symbol plot is displayed, the model is swept as specified, but the symbols appear only on the original planar model faces.

Extrude elements in your model

1. Locate the **Extrude** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Sweep/Extrude** tab in the dialog box that appears. The **Extrude** options are in the lower portion of the page.



Note:

If there are no elements in your model that can be extruded, the **Extrude** options will not be available.

2. Toggle **Extrude elements**.

   When **Extrude elements** is on, the **Depth** option becomes available.

3. Specify the depth of the extrusion by typing a positive number (in model units) in the **Depth** field. The default depth is 1.0.

   Abaqus/CAE extends your model the specified number of units in the negative *z*-direction.

   Use the view manipulation rotation tool to view the extruded model in three dimensions.

4. Click **Apply** to implement your changes.

   Your extrude specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

   If a contour plot is displayed, the contours appear on all extruded faces.

   If a symbol plot is displayed, the model is extruded as specified, but the symbols appear only on the original planar model faces.

Extrude analytical rigid surfaces in your model

1. Locate the **Extrude** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Sweep/Extrude** tab in the dialog box that appears. The **Extrude** options are in the lower portion of the page.



Note:

If there are no elements or analytical rigid surfaces in your model that can be extruded, the **Extrude** options will not be available.

2. Toggle on **Extrude analytical rigid surfaces** if it is not currently active.

   When **Extrude analytical rigid surfaces** is on, the **Depth** options become available.

3. Select a depth for the extrusion.

   You can either accept the default depth by selecting **Auto-compute**, or you can specify a custom depth by typing a positive number (in model units) in the **Specify** field. The default extrusion depth for analytical rigid surfaces is the maximum value that does not stretch the bounding box.

   Abaqus/CAE extends your model the specified number of units in the negative *z*-direction.

   Use the view manipulation rotation tool to view the extruded model in three dimensions.

4. Click **Apply** to implement your changes.

   Your extrude specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Using mirrors and patterns to display your results

In the Sketch module you can use mirrors and patterns to copy a common feature and complete the sketch of a feature. Similarly, in the Visualization module you can copy analysis results representing a repetitive portion of a model to visualize results for the entire model.
To mirror results, you can select the coordinate system and up to three mirrors corresponding to the principal planes of the selected system. To pattern results, you can select the coordinate system and apply a rectangular or circular pattern. You can apply mirrors, rectangular patterns, and circular patterns in conjunction; and you can select their order of application to produce the desired view.



Note:

There are no restrictions to prevent copied results from obscuring portions of the original data or other copies in the viewport.

Mirrors and patterns are a visualization aid only. Any numerical representation of the results, such as the contour legend, indicates only the portion of the model that was analyzed. Use great care in selecting your model and the mirror and pattern settings so that the displayed results will accurately simulate the complete model.

Figure 1 shows one quarter of a plastic fastener as it was modeled for analysis (left) and the same model mirrored about two symmetry planes to display the complete fastener (right). (The rigid hole is a swept profile and, thus, appears complete prior to mirroring the fastener.)

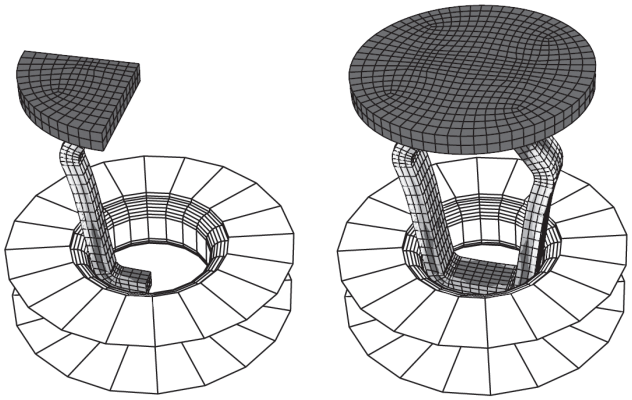


Figure 1: Quarter-symmetry model (left) and complete fastener display (right).

Figure 2 shows a rectangular pattern of completed fasteners.

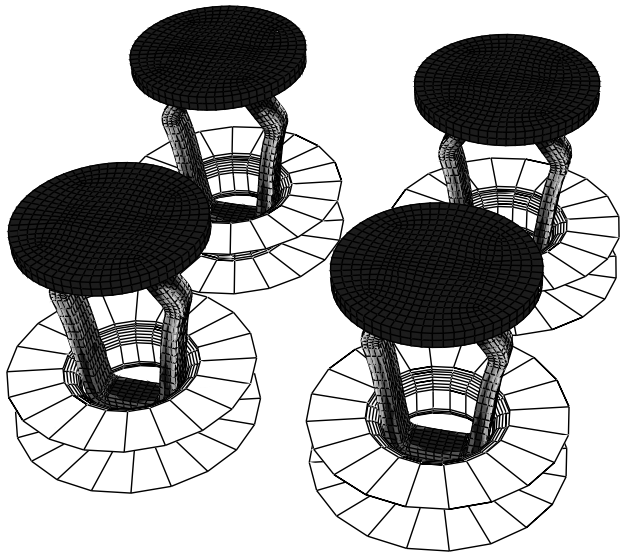


Figure 2: A rectangular pattern of fasteners.

Additional information

- *Sweeping and extruding your model*
- *Refining curved edges and faces*

Mirror your model

1. Locate the **Mirror** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Mirror/Pattern** tab in the dialog box that appears. The **Mirror** options are in the upper portion of the page.

2. Accept the default global coordinate system, or select a coordinate system from the **Mirror CSYS** list.

   You can select any coordinate system that exists in the output database. If a suitable coordinate system does not already exist, you can create a new coordinate system using one of the methods described in *Creating coordinate systems during postprocessing*.

3. Toggle on the mirror planes that you want Abaqus/CAE to use to copy the model results.
4. Toggle on **Mirror display bodies** if you also want Abaqus/CAE to copy display bodies.
5. If you apply mirrors in conjunction with rectangular or circular patterns, select the **Order of Operations** that Abaqus/CAE will use to create the copies.
6. Click **Apply** to implement your changes.

   Your specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

   If a contour plot is displayed, the contours appear on all faces.

   If a symbol or material orientation plot is displayed, the symbols or material orientations appear only on the original model faces.

Pattern your model

1. Locate the **Pattern** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **Mirror/Pattern** tab in the dialog box that appears. The **Pattern** options are in the middle of the page.

2. Accept the default global coordinate system, or select a coordinate system from the **Pattern CSYS** list.

   You can select any coordinate system that exists in the output database. If a suitable coordinate system does not already exist, you can create a new coordinate system using one of the methods described in *Creating coordinate systems during postprocessing*.

3. Edit the rectangular and circular pattern parameters as follows:

   - Rectangular pattern

     - Enter the **Number** of results that you want to display in the *X*-, *Y*-, and *Z*-directions of the selected coordinate system.
     - Enter the **Offset** distance, in model units, between copies for each direction.

       To place the copies adjacent to each other, the offset in each direction must be the same as the model dimension in that direction.

       Tip: If you do not know the model dimensions, use the query tool  to determine the distance between the first and last model nodes in the desired direction. (For more information, see *Understanding the Query toolset in the Visualization module*.)

   - Circular pattern

     - Select the axis of rotation.
     - Enter the **Number** of results that you want to display around the selected axis.
     - If desired, edit the **Total angle**Abaqus/CAE will use to position the copies.

       Abaqus/CAE places the original result at 0°. All subsequent copies are positioned according to the **Total angle** divided by the **Number** of copies.

4. If you use both rectangular and circular patterns or if you use a pattern in conjunction with mirrors, select the **Order of Operations** that Abaqus/CAE will use to create the copies.
5. Click **Apply** to implement your changes.

   Your specifications are reflected in all plot states and are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

   If a contour plot is displayed, the contours appear on all faces.

   If a symbol or material orientation plot is displayed, the symbols or material orientations appear only on the original model faces.

Refining curved edges and faces

If your model contains quadratic elements, cubic elements, 2-node spring elements, or analytical rigid surfaces having curved segments, you can control the refinement with which Abaqus/CAE displays these curved model components. Similarly, if your model contains 2-node dashpot elements, you can control the refinement of the symbol used to represent those elements. Click the **General** tab in the **ODB Display Options** dialog box to choose a refinement level between extra coarse and extra fine for the display of curved edges and faces. When you select **Extra Coarse** refinement, Abaqus/CAE displays curved elements using straight lines; midside nodes have no effect on the shape displayed for these elements. **Extra Coarse** refinement will treat springs and dashpots as straight lines. For example, in *Figure 1* the **Extra Coarse** refinement level was used to create the undeformed plot on the left, and the **Extra Fine** refinement level was used to produce an undeformed plot of the same model on the right.

Note:

As the refinement level increases, the display performance will decrease. The **Medium** setting should produce acceptable results for most models.

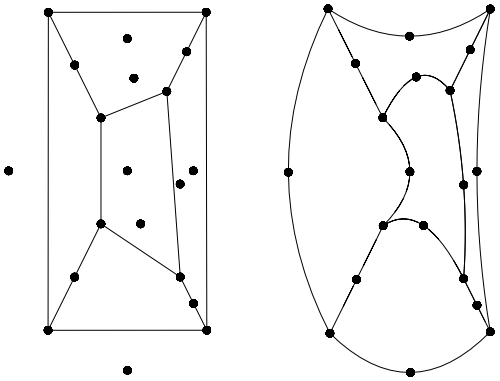


Figure 1: Model plotted with different refinement levels (extra coarse and extra fine).

1. Locate the **Curved Lines & Faces** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **General** tab in the dialog box that appears. The **Curved Lines & Faces** options are at the top of the **General** page.
2. Click the **Refinement level** button to reveal the refinement options.
3. In the refinement list, click the level you want. Extra fine produces the smoothest curves but may slow plotting; conversely, coarse refinement gives a rough appearance but plots most quickly.

   The refinement level you specify appears on the **Refinement Level** button.
4. Click **Apply** to implement your changes.

   Curved edges and faces for all plot states change to reflect your refinement specification, which is saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Sweeping and extruding your model*

Coloring elements with no results

To produce a contour plot, Abaqus/CAE reads results for the variable and frame you specify from the output database. Results for a particular variable and frame may not be available or may not be applicable for one or more elements included in the plot. You can specify the display color for elements that have no results. The default color is white.

For example, the contour plot in *Figure 1* shows the von Mises stress generated when a blank sheet is deformed using a die and punch (the punch is not shown in the figure). In this case the color gray was selected for elements with no results. The die elements appear gray because the die is a rigid surface for which no stress results exist.

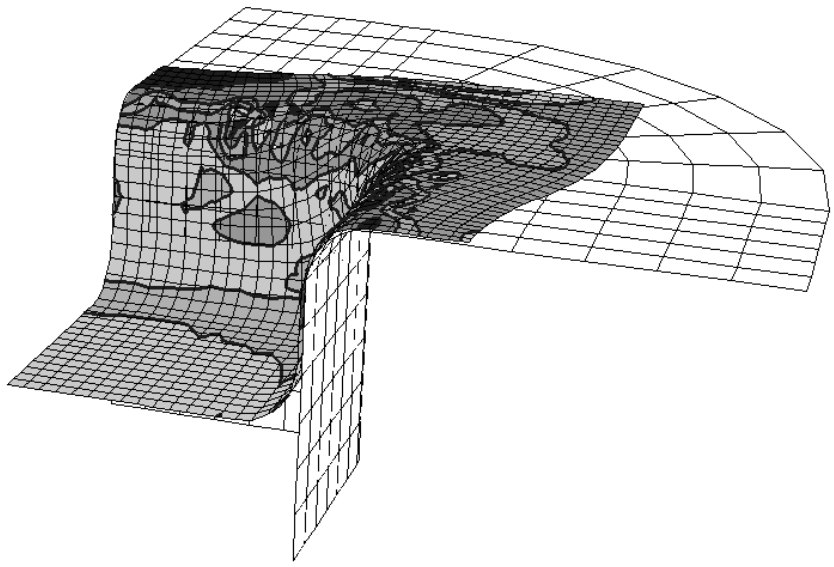


Figure 1: Contour plot of blank and die.

1. Locate the **Elements with No Results** options.

   From the main menu bar, select **View->ODB Display Options**. Click the **General** tab in the dialog box that appears. The **Elements with No Results** options are in the center of the **General** page.

2. Click the color sample .

   Abaqus/CAE displays the **Select Color** dialog box.

3. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

4. Click **OK** to close the **Select Color** dialog box.

   The color sample changes to the selected color.

5. Click **Apply** to implement your changes.

   Elements with no results in contour plots (including display bodies) change to reflect your coloring specification, which is saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Producing a contour plot*

- *Customizing render style, translucency, and fill color*

Controlling beam profile display for postprocessing

Beam rendering is available in the Visualization module for undeformed, deformed, and contour plots only. For deformed plots Abaqus/CAE scales the displacement components of the deformation based on the deformed field output variable but does not scale the twisting of the beam profile using the rotational components of the deformation when you change the deformation scale factor.

You can view a realistic display of the beam profile for results data by selecting **View->ODB Display Options** in the Visualization module.

The rendering of beam profiles requires the presence of beam orientation data in the output database. Beam normals are written to the output database automatically for all the steps that include field output. For analyses that include geometric nonlinearity, you can display beam profiles during postprocessing only if nodal output is requested in the model. For more information, see *Using a Beam Section Integrated during the Analysis to Define the Section Behavior*.

Abaqus/CAE does not render the tapering of beam profiles along their length. If your model includes tapered beam sections, Abaqus/CAE renders these beams using the beam's starting profile along its entire length. For more information about tapered beams, see *Creating beam sections*.

Beam elements that have BRADIUS values are rendered using circular beam profiles with the radius coming from the BRADIUS field in the output database.

Displaying beam profiles is useful for checking that the correct profile has been assigned to a particular region and that the assigned beam orientation results in the expected orientation of the profile. For example, *Figure 1* shows the box beam profiles displayed on the light-service crane described in *Example: cargo crane*.

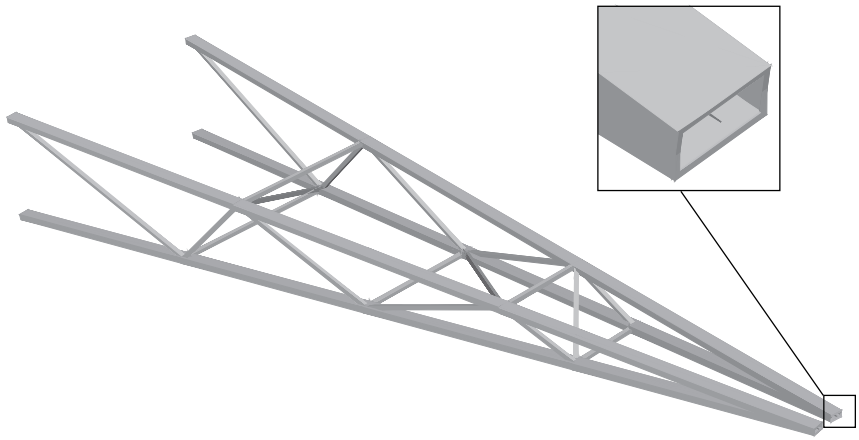


Figure 1: The cargo crane example with beam profiles displayed.

Note:

If your results include quadratic beam elements, Abaqus/CAE does not consider the midside nodes in those elements when the beam profiles are rendered.

If you assign a general beam section to a wire, Abaqus/CAE displays the beam profile as an ellipse with a cross-sectional area and moments of inertia (I_{11} and I_{22}) that match the values you specified. If you assign a truss section to a wire, Abaqus/CAE displays the truss profile as a circle with a cross-sectional area that matches the value you specified.

Abaqus/CAE renders beam profiles according to the current settings for color coding and translucency. When these settings change, the color and translucency of the beam profiles change as well. When beam profiles are displayed, Abaqus/CAE disables scaling and shrinking of the model.

1. Locate the **Render beam profiles** option.

   From the main menu bar, select **View->ODB Display Options**. In the dialog box that appears, click the **General** tab.
2. From the bottom of the dialog box, toggle on **Render beam profiles** to display beam profiles.
3. If desired, apply a **Scale factor** to the beam profiles to increase or decrease their size. The default value is 1.
4. + Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE displays the profile of the beam section with the appropriate dimensions and in the correct orientation. By default, your changes are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

- *Producing a contour plot of linear beam section stresses*
- *Controlling beam profile display*

Controlling shell thickness display for postprocessing

Rendering of shell thickness is available in the Visualization module for all plot states.

If you use shell elements to model relatively thin components in your analysis, you can use the **View->ODB Display Options** to view the actual thickness of these shell elements in your model during postprocessing.

For deformed and contour plots Abaqus/CAE scales the displacement components of the deformation but does not scale the rotational components of the deformation when you change the deformation scale factor. For contour plots Abaqus/CAE applies colors based on your selection of active section point locations. If the top section point or bottom section point is currently active, Abaqus/CAE displays the contour for that section point throughout the shell thickness; if both top and bottom section points are currently active, Abaqus/CAE creates a linear contour gradient from the top values to the bottom values through the shell thickness.

Displaying shell thickness after a sizing optimization allows you to view the resulting change in shell thickness across the design area of your structure as the optimization progresses.

Abaqus/CAE determines shell thickness for postprocessing using the shell thickness definition specified for the model in its undeformed state. If your model undergoes large deformations or large rotations during the analysis, shell thickness might not display well in the deformed plot state.

Displaying shell thickness enables you to examine the thickness of shell geometry relative to the rest of the model. You can apply a scale factor to reduce or increase the display of shell thickness for your session. *Figure 1* shows the effect of scale factor changes to a model.

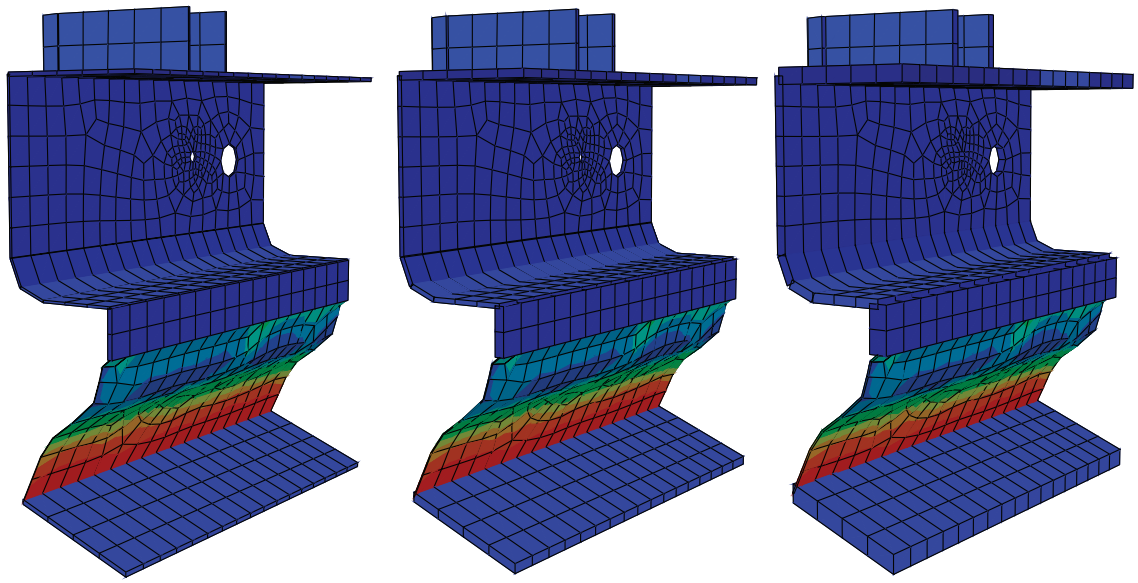


Figure 1: From left to right: shell thickness scale factor settings of 0.5, 1 (default), and 2.

Abaqus/CAE renders shell thickness for three-dimensional shell elements only; thickness is not displayed for axisymmetric shell elements, such as SAX1 elements. When shell thickness is displayed, Abaqus/CAE also renders the edges of shell geometry unless a view cut is displayed in the viewport. Abaqus/CAE renders shell thickness according to the current settings for color coding and translucency. When these settings change, the color and translucency of the shell thickness change as well.

1. Locate the **Render shell thickness** option.

   From the main menu bar, select **View->ODB Display Options**. In the dialog box that appears, click the **General** tab.

2. From the bottom of the dialog box, toggle on **Render shell thickness** to display shell thicknesses for the shell sections in your model.
3. If desired, apply a **Scale factor** to the shell thicknesses to increase or decrease their thickness. The default value is 1, which produces a realistic rendering of the shell thickness settings.
4. + Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE displays the shell sections in the selected part or assembly with the appropriate thickness. By default, your changes are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Controlling shell thickness display*

Controlling the display of thermal fluid pipe elements

Rendering of thermal fluid pipe elements is available in the Visualization module for undeformed, deformed, and contour plots only.
There is no deformation associated with thermal fluid pipe elements; the deformed and undeformed plots are the same.

If you use thermal fluid pipe elements to model the flow of a liquid through a pipe in your analysis, you can use the **View->ODB Display Options** to visualize the fluid pipe along with the fluid pressure, fluid temperature, and wall temperature contours. To clearly visualize the contours and plot shapes, you should create a display group that contains only the thermal fluid pipe elements.

When you select the regions individually, Abaqus/CAE renders the fluid region as a solid cylinder and the inner wall and outer wall as thin hollow cylinders. If you choose to render both the inner wall and outer wall, a thick-walled cylinder is displayed. Temperature variation across the wall is visible on the end caps. When you render multiple regions, only the outermost region is visible on the surface of the cylinder; all the selected regions are rendered at the end caps. *Figure 1* shows the nodal temperature (NT11) on the inner wall and outer wall of a fluid pipe.

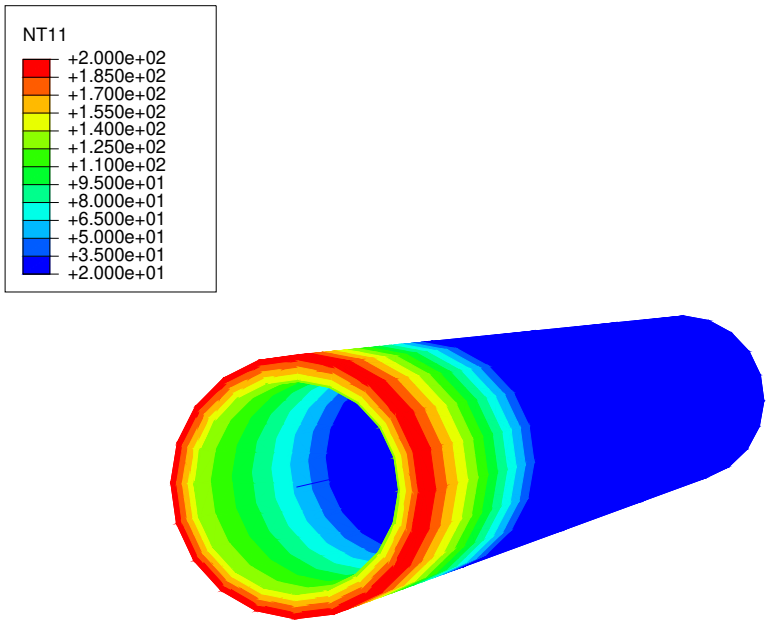


Figure 1: Rendering of fluid pipe thermal elements.

Abaqus/CAE renders thermal fluid pipe elements according to the current settings for color coding and translucency. When these settings change, the color and translucency of the thermal fluid pipe elements change as well. When thermal fluid pipe elements are displayed, Abaqus/CAE disables scaling and shrinking of the model.

1. Locate the **Fluid pipe thermal elements** option.

   From the main menu bar, select **View->ODB Display Options**. In the dialog box that appears, click the **General** tab.

2. From the bottom of the dialog box, toggle on **Fluid pipe thermal elements** to render the thermal fluid pipe elements in your model.
3. Toggle the following regions that you want to render:

   - **Fluid region**

- **Inner wall**
- **Outer wall**

4. If desired, apply a **Scale factor** to the radius of each of the selected regions to increase or decrease its thickness. The default value is 1.
5. + Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE displays the thermal fluid pipe elements in the selected part or assembly. By default, your changes are saved for the duration of the session. If you want to retain the changes you applied for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Selecting plot display options*

Viewing removed elements

You can create model change interactions that deactivate selected components of the model (geometry, elements, skins, and stringers). Skins, geometry, and elements can be removed from the viewport for model change interactions on model geometry. Skins, stringers, and elements can be removed from the viewport for model change interactions on an orphan mesh. You can control the visibility of these removed components from the **ODB Display Options** dialog box.

The behavior of the viewport depends on the status of the visibility toggle and on the plot state. Elements deactivated in a model change interaction can be removed from the viewport for all plot states. When the option to remove deactivated elements in the viewport is toggled off, the deactivated elements are colored gray in the contour plot but remain unchanged for all other plot states. If the option to display beam profiles is toggled off, Abaqus/CAE will not remove deactivated elements with beam profiles from the viewport, regardless of the status of the removed element toggle.

1. From the main menu bar, select **View->ODB Display Options**. Click the **General** tab in the dialog box that appears. The **Model Change** display option is at the bottom of the **General** page.
2. Toggle on **Account for deactivated elements** from the viewport.
3. Click **OK** to implement your changes and to close the dialog box.

Additional information

• *Defining a model change interaction*

Customizing viewport annotations

The viewport annotations generated by Abaqus/CAE consist of the legend, the title block, the state block, the 3D compass, and the view triad; they are provided to help you identify and interpret the contents of your plot. In addition, you can add customized text and arrow annotations to a viewport (for more information, see *Managing viewports on the canvas*). You can use the **Viewport Annotation Options** to toggle all viewport annotations on and off, and you can also customize the appearance of the annotations generated by Abaqus/CAE.

For information on the 3D compass, see *The 3D compass*. For information on the view triad, see *Customizing the view triad*.

In this section:

- *Customizing the legend*
- *Customizing the title block*
- *Customizing the state block*
- *Using viewport annotation options*

Customizing the legend

The legend is a key to help you interpret your plots. For example, a contour plot legend shows the values that each contour color represents; a symbol plot legend shows the values that each vector color represents. The *X–Y* plot legend refers to each curve by label and shows the curve's style.

Figure 1 shows a contour plot legend and a symbol plot legend.

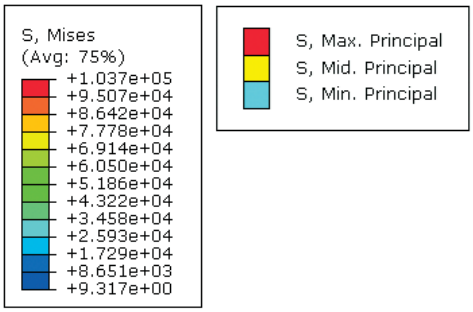


Figure 1: Contour plot legend and symbol plot legend.

You can display or suppress legends for all plot states, and you can customize the following:

- the position of the legend,
- the font and color of the legend text,
- the appearance of a box outlining the legend,
- whether minimum and maximum plot values are shown,
- the appearance of the legend background (the rectangular area behind the legend),
- the formatting of the numbers displayed for the digit values, and
- the number of digits displayed for the legend values.

You cannot directly specify the size of the legend. However, by varying the size of the text font or the number of digits in the legend values, you can increase or decrease the legend's size.

The content of the legend depends on the plot type and on plot state–specific options such as the number of contour intervals. For more information on contour plot legend content, see *Understanding contour plotting*.

This section describes how to customize the legend for all plot states except *X–Y* plots. To customize the *X–Y* plot legend, see *Customizing the X–Y plot legend*.

1. Locate the **Legend** options.

   From the main menu bar, select **Viewport->Viewport Annotation Options**.

2. To display or suppress legends for all plot states except *X–Y* plots, toggle **Show legend** in the **General** tab.

   When **Show legend** is on, legend options become available.

3. Click the **Legend** tab. The basic legend options are at the top of the **Legend** page.
4. To display or suppress a box outlining the legend, toggle **Show bounding box**. This “bounding box” visually separates the legend from the underlying plot.
5. To display or suppress the minimum and maximum values associated with contour plots, toggle **Show min/max values**.
6. Establish the position of the upper left corner of the legend.

To control the *X*- or *Y*-position, click the **% Viewport X** or **% Viewport Y** boxes, respectively. Enter the positions you want as a percentage of the total width of the current viewport.

7. Customize the legend text font.
   a. Click **Set Font**.
      The **Select Font** dialog box appears.
   b. Use the **Select Font** dialog box to choose the font characteristics you want. For more information, see *Customizing fonts*.
   c. When you are done, click **OK** to implement your changes and to close the **Select Font** dialog box.
8. Choose the color of the legend text.
   a. Click the color sample .
      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.
      The color sample changes to the selected color.
9. To customize the background, click one of the following:
   - **Match viewport** to match the background to the viewport color.
   - **Transparent** to eliminate the background and show only the text.
   - **Other color** to reveal other background color options.

   If you selected **Other color**:
   a. Click the color sample .
      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.
      The color sample changes to the selected color.
10. To change the format of the legend values, click the **Format** arrow and select one of the following formats:
   - Select **Scientific** to display legend values in scientific notation, which expresses legend values as a decimal value between 1 and 10 multiplied by a power of 10. When you select this format for the number `501,000`, the value appears in the legend as `5.01e5`.
   - Select **Fixed** to display legend values without using exponent values. When you select this format, the number `501,000` is displayed in the legend as `501000.00`.
   - Select **Engineering** to select engineering notation, which expresses values in a similar format to scientific notation but allows exponent values that are multiples of three only. When you select this format, the number `501,000` is displayed in the legend as `501E3`.



Note:

All three of these examples are displayed with two decimal places. Your legend values may be displayed differently if you change the **Decimal places** option.

11. To choose the number of digits shown in legend values, click the **Decimal places** arrow. For legend values in scientific or fixed format, the **Decimal places** option controls the number of digits to the right of the decimal point. For legend values in engineering format, this option controls the number of significant digits to the left of the exponent value. For all formats, larger **Decimal places** values require more display space.

The specified number of digits appears in the **Decimal places** box.

12. Click **Apply** to implement your changes.

In the current viewport, legends for all plots change to reflect your specifications, which are saved for the duration of the session.

Customizing the title block

The title block contains text identifying the model and results shown in the current plot. This text indicates:

- The name of the output database (from the name of the analysis job).
- The description of the model.
- The product name (usually Abaqus/Standard or Abaqus/Explicit) and release used to generate the output database.
- The date the output database was last modified.

You can display or suppress the title block for all plot states, and you can display or suppress the appearance of a box bounding the title block. In addition, you can control the position, text font, and text color of the title block and the appearance of the title block background. You cannot directly specify the size of the title block. However, by varying the size of the text font, you can increase or decrease the title block's size. The content of the title block is fixed and cannot be customized.

1. Locate the **Title Block** options.

   From the main menu bar, select **Viewport->Viewport Annotation Options**.
2. To display or suppress title blocks for all plot states, toggle **Show title block** in the **General** tab.

   When **Show title block** is on, the title block options become available.
3. Click the **Title Block** tab. The basic title block options are at the top of the **Title Block** page.
4. To display or suppress a box outlining the title block, toggle **Show bounding box**. This “bounding box” visually separates the title block from the underlying plot.
5. Establish the position of the upper left corner of the title block.

   To control the *X*- or *Y*-position, click the **% Viewport X** or **% Viewport Y** boxes, respectively. Enter the positions you want as a percentage of the total width of the current viewport.
6. Customize the title block text font.
   a. Click **Set Font**.

      The **Select Font** dialog box appears.
   b. Use the **Select Font** dialog box to choose the font characteristics you want. For more information, see *Customizing fonts*.
   c. When you are done, click **OK** to implement your changes and to close the **Select Font** dialog box.
7. Choose the color of the title block text.
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.
   b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.
8. To customize the background, click one of the following:

   - **Match viewport** to match the background to the viewport color.
   - **Transparent** to eliminate the background and show only the text.
   - **Other color** to reveal other background color options.

   If you selected **Other color**:

a. Click the color sample .

Abaqus/CAE displays the **Select Color** dialog box.

b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

c. Click **OK** to close the **Select Color** dialog box.

The color sample changes to the selected color.

9. Click **Apply** to implement your changes.

In the current viewport, title blocks for all plots change to reflect your specifications, which are saved for the duration of the session.

Customizing the state block

The state block contains text identifying the analysis results associated with the current plot. This information includes step, frame, results variables, deformation magnification factor, eigenmode, and eigenvalue, as applicable.

You can display or suppress the state block for all plot states, and you can display or suppress the appearance of a box bounding the state block. In addition, you can control the position, text font, and text color of the state block and the appearance of the state block background. You cannot directly specify the size of the state block. However, by varying the size of the text font, you can increase or decrease the state block's size. The content of the state block is fixed and cannot be customized.

1. Locate the **State Block** options.

   From the main menu bar, select **Viewport->Viewport Annotation Options**.

2. To display or suppress state blocks for all plot states, toggle **Show state block** in the **General** tab.

   When **Show state block** is on, state block options become available.

3. Click the **State Block** tab. The basic state block options are at the top of the **State Block** page.
4. To display or suppress a box outlining the state block, toggle **Show bounding box**. This “bounding box” visually separates the state block from the underlying plot.
5. Establish the position of the upper left corner of the state block.

   To control the *X*- or *Y*-position, click the **% Viewport X** or **% Viewport Y** boxes, respectively. Enter the positions you want as a percentage of the total width of the current viewport.

6. Customize the state block text font.
   a. Click **Set Font**.

      The **Select Font** dialog box appears.

   b. Use the **Select Font** dialog box to choose the font characteristics you want. For more information, see *Customizing fonts*.
   c. When you are done, click **OK** to implement your changes and to close the **Select Font** dialog box.

7. Choose the color of the state block text.
   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

      The color sample changes to the selected color.

8. To customize the background, click one of the following:

   - **Match viewport** to match the background to the viewport color.
   - **Transparent** to eliminate the background and show only the text.
   - **Other color** to reveal other background color options.

   If you selected **Other color**:

   a. Click the color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   c. Click **OK** to close the **Select Color** dialog box.

The color sample changes to the selected color.

9. Click **Apply** to implement your changes.

   In the current viewport, state blocks for all plots change to reflect your specifications, which are saved for the duration of the session.

Using viewport annotation options

Use the viewport annotation options to customize the legend, the title block, the state block, the 3D compass, and the view triad for all plot states. In addition, the viewport annotation options control the display of arrow and text annotations.

1. From the main menu bar, select **Viewport->Viewport Annotation Options**.

   The **Viewport Annotation Options** dialog box appears. It contains the following tabs:

   - **General**: Control the display of all viewport annotations—those generated by Abaqus/CAE and any arrow and text annotations that you created.
   - **Triad**: Control the appearance and position of the view triad.
   - **Legend**: Control the appearance and position of the legend and whether minimum and maximum values are shown.
   - **Title Block**: Control the appearance and position of the title block.
   - **State Block**: Control the appearance and position of the state block.

2. Use the **General** tab to control the display of viewport annotations for all plots in the current viewport. You can display all viewport annotations in the current viewport by clicking **Set all on**, or you can hide all viewport annotations in the current viewport by clicking **Set all off**.
3. Use the remaining tabs in the dialog box to customize the viewport annotations generated by Abaqus/CAE. To create and customize arrow and text annotations, see *Working with viewport arrow and text annotations*.

Additional information

- *What are arrow and text annotations?*
- *Customizing the 3D compass*
- *Customizing the view triad*
- *Customizing the legend*
- *Customizing the title block*
- *Customizing the state block*

Using toolsets

This part describes how to use each of the toolsets in Abaqus/CAE, except those in the Visualization module. Toolsets in the Visualization module are discussed in *Viewing results*.

In this section:

- *The Amplitude toolset*
- *The Analytical Field toolset*
- *The Attachment toolset*
- *The CAD Connection toolset*
- *The Customize toolset*
- *The Datum toolset*
- *The Discrete Field toolset*
- *The Edit Mesh toolset*
- *The Feature Manipulation toolset*
- *The Filter toolset*
- *The Free Body toolset*
- *The Options toolset*
- *The Geometry Edit toolset*
- *The Partition toolset*
- *The Query toolset*
- *The Reference Point toolset*
- *The Set and Surface toolsets*
- *The Stream toolset*
- *The Virtual Topology toolset*

The Amplitude toolset

Amplitudes allow you to specify arbitrary time or frequency variations of load, displacement, and some interaction attributes throughout a step using step time or throughout an analysis using total time. The Amplitude toolset allows you to create and manage amplitudes.

In this section:

- *Understanding the role of the Amplitude toolset*
- *Understanding the amplitude editors*
- *Selecting an amplitude type to define*
- *Using tabular data to define an amplitude curve*
- *Using equally spaced data to define an amplitude curve*
- *Using periodic data to define an amplitude curve*
- *Using modulated data to define an amplitude curve*
- *Using exponential decay data to define an amplitude curve*
- *Defining a solution-dependent amplitude curve*
- *Using smooth step data to define an amplitude curve*
- *Defining a spectrum*
- *Defining a user-defined amplitude curve*
- *Defining a PSD definition*

Understanding the role of the Amplitude toolset

The Amplitude toolset allows you to create any type of amplitude that is supported by Abaqus/Standard or Abaqus/Explicit.
Amplitudes created in the Amplitude toolset always involve relative data, while amplitudes defined directly in the input file can involve either relative or absolute data. (For more information, see “Specifying relative or absolute data” in *Amplitude Curves*.) You can also use the Amplitude toolset to define a spectrum to be used in a response spectrum analysis.

Select **Tools->Amplitude->Create** from the main menu to create a new amplitude definition; select **Edit** from the same menu to make changes to an existing definition. Either command opens the amplitude editor, which allows you to select the options and provide the data needed to define your amplitude.

You can also plot amplitude data in much the same way that you can plot *X*–*Y* data. For more information about using the **Amplitude Plotter** plug-in, see *Plotting amplitude data*.

Additional information

- *Amplitude Curves*
- *Response Spectrum Analysis*

Understanding the amplitude editors

You create amplitudes by entering data in the amplitude editor; you can type the data in using the keyboard or you can read it in from a file. (For more information, see *Entering tabular data*.)

The top panel of the editor displays the name of the amplitude and the amplitude type. The format of the rest of the editor depends on the type of amplitude you are creating. For example, the editor for creating a periodic amplitude is shown in *Figure 1*.

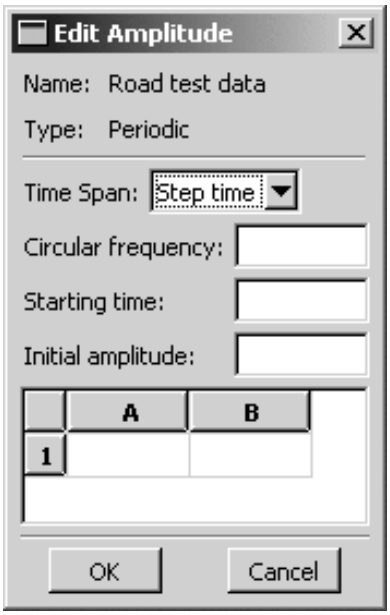


Figure 1: The amplitude editor.

Additional information

- *Amplitude Curves*
- *Response Spectrum Analysis*
- *Entering tabular data*

Selecting an amplitude type to define

Select **Tools->Amplitude->Create** from the main menu bar to create an amplitude. For detailed information on amplitudes, see *Amplitude Curves* and *Specifying a Spectrum*.

1. From the main menu bar, select **Tools->Amplitude->Create**.

   Tip: You can also create an amplitude by clicking mouse button 3 on the **Amplitudes** container in the Model Tree or by clicking **Create** in the **Amplitude Manager**.

   The **Create Amplitude** dialog box appears.

2. In the **Name** field, enter a name for the amplitude. For information on naming objects, see *Using basic dialog box components*.
3. Choose the **Type** of amplitude that you want to create:

   - Choose **Tabular** to define the amplitude curve as a table of values at convenient points on the time scale. Abaqus interpolates linearly between these values, as needed. For more information, see *Defining Tabular Data*.
   - Choose **Equally spaced** to give a list of amplitude values at fixed time intervals beginning at a specified value of time. Abaqus interpolates linearly between each time interval. For more information, see *Defining Equally Spaced Data*.
   - Choose **Periodic** to define the amplitude, a, as a Fourier series:

     $$a = A_0 + \sum_{n=1}^{N} \left[A_n \cos n\omega \left(t - t_0 \right) + B_n \sin n\omega \left(t - t_0 \right) \right] \text{ for } t \geq t_0,$$
     $$a = A_0 \quad \text{for } t < t_0,$$

     where t_0, N, ω, A_0, A_n, and B_n, $n = 1, 2 \ldots N$, are user-defined constants. For more information, see *Defining Periodic Data*.
   - Choose **Modulated** to define the amplitude, a, as

     $$a = A_0 + A \sin \omega_1 \left(t - t_0 \right) \sin \omega_2 \left(t - t_0 \right) \quad \text{for } t > t_0,$$
     $$a = A_0 \quad \text{for } t \leq t_0,$$

     where A_0, A, t_0, ω_1, and ω_2 are user-defined constants. For more information, see *Defining Modulated Data*.
   - Choose **Decay** to define the amplitude, a, as

     $$a = A_0 + A \exp \left(- \left(t - t_0 \right) / t_d \right) \quad \text{for } t \geq t_0,$$
     $$a = A_0 \quad \text{for } t < t_0,$$

     where A_0, A, t_0, and t_d are user-defined constants. For more information, see *Defining Exponential Decay*.
   - Choose **Solution dependent** to calculate amplitude values based on a solution-dependent variable. For more information, see *Defining a Solution-Dependent Amplitude for Superplastic Forming Analysis*.
   - Choose **Smooth step** to define the amplitude, a, between two consecutive data points (t_i, A_i) and (t_{i+1}, A_{i+1}) as

$$a = A_i \quad + \quad (A_{i+1} - A_i)\,\xi^3\left(10 - 15\xi + 6\xi^2\right) \quad \text{for } t_i \le t \le t_{i+1},$$

where $\xi = (t - t_i) / (t_{i+1} - t_i)$. For more information, see *Defining Smooth Step Data*.

- Choose **Actuator** to import the current value of an actuator amplitude at any given time from a co-simulation with a logical modeling program. For more information, see *Defining an Actuator Amplitude via Co-Simulation*. No additional data is required to define the amplitude curve.
- Choose **Spectrum** to define a spectrum to be used in a response spectrum analysis. For more information, see *Specifying a Spectrum*.
- Choose **User** to define the amplitude curve in user subroutine *UAMP* (Abaqus/Standard) or *VUAMP* (Abaqus/Explicit). For more information, see *Defining an Amplitude via a User Subroutine*.
- Choose **PSD definition** to define a frequency function that defines the frequency dependence of the random loading in a random response analysis step. This amplitude curve represents the power spectral density function for the random noise source. The PSD amplitude can be referenced in the correlation definition of a base motion boundary condition in a random response step. For more information, see *Defining the Frequency Functions*.

4. Click **Continue**.

   The **Edit Amplitude** dialog box appears in which you can enter all of the data necessary to define the amplitude curve. See the following sections for detailed instructions:

   - *Using tabular data to define an amplitude curve*
   - *Using equally spaced data to define an amplitude curve*
   - *Using periodic data to define an amplitude curve*
   - *Using modulated data to define an amplitude curve*
   - *Using exponential decay data to define an amplitude curve*
   - *Defining a solution-dependent amplitude curve*
   - *Using smooth step data to define an amplitude curve*
   - *Defining a spectrum*
   - *Defining a user-defined amplitude curve*
   - *Defining a PSD definition*

Additional information

- *Amplitude Curves*
- *Understanding the amplitude editors*
- *Entering tabular data*

Using tabular data to define an amplitude curve

Use the tabular definition method to define the amplitude curve as a table of values at convenient points on the time scale. Abaqus interpolates linearly between these values, as needed. For more information, see *Defining Tabular Data*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. Indicate how you want to define **Smoothing**:
   - Choose **Use solver default** to accept a default value of 0.25 in Abaqus/Standard and 0.0 in Abaqus/Explicit.
   - Choose **Specify** to enter a value for the smoothing parameter in the adjacent field. A value of 0.05 is suggested for amplitude definitions that contain large time intervals to avoid severe deviation from the specified definition.

   The **Smoothing** parameter is the fraction of the time interval before and after each time point during which the piecewise linear time variation is replaced by a smooth quadratic time variation. This parameter is applicable only when time derivatives are needed (for displacement or velocity boundary conditions in a direct integration dynamic analysis) and is ignored for all other uses of this option.
4. Display the **Amplitude Data** tabbed page, and enter the tabular data. For detailed information on how to enter data, see *Entering tabular data*.
5. If desired, display the **Baseline Correction** tabbed page.

   When you use an amplitude definition to define an acceleration history in the time domain (a seismic record of an earthquake, for example), the integration of the acceleration record through time may result in a relatively large displacement at the end of the event. This behavior typically occurs because of instrumentation errors or a sampling frequency that is not sufficient to capture the actual acceleration history. In Abaqus/Standard it is possible to compensate for it by using "baseline correction." For more information, see *Baseline Correction in Abaqus/Standard*.
6. Click the arrow to the right of the **Correction** field, and select one of the following options:
   - Select **None** for no baseline correction.
   - Select **Single interval** to treat the entire time of the amplitude definition as a single correction interval.
   - Select **Multiple intervals** to treat the entire time of the amplitude definition as multiple correction intervals. If you select this option, enter the time points defining the different correction intervals (i.e. in the first row enter the time point defining the end of the first correction interval and the beginning of the section correction interval; in the second row enter the time point defining the end of the second correction interval and the beginning of the third correction interval, etc.).
7. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Using equally spaced data to define an amplitude curve

Use the equally spaced definition method to provide a list of amplitude values at fixed time intervals beginning at a specified value of time. Abaqus interpolates linearly between each time interval. For more information, see *Defining Equally Spaced Data*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. Indicate how you want to define **Smoothing**:
   - Choose **Use solver default** to accept a default value of 0.25 in Abaqus/Standard and 0.0 in Abaqus/Explicit.
   - Choose **Specify** to enter a value for the smoothing parameter in the adjacent field. A value of 0.05 is suggested for amplitude definitions that contain large time intervals to avoid severe deviation from the specified definition.

   The **Smoothing** parameter is the fraction of the time interval before and after each time point during which the piecewise linear time variation is replaced by a smooth quadratic time variation. This parameter is applicable only when time derivatives are needed (for displacement or velocity boundary conditions in a direct integration dynamic analysis) and is ignored for all other uses of this option.
4. Display the **Amplitude Data** tabbed page.
5. In the **Fixed interval** field, enter the fixed time or frequency interval at which you will provide the amplitude data.
6. In the first row of the **Time/Frequency** column in the data table, enter the time (or lowest frequency) at which you will enter the first amplitude.
7. Enter the **Amplitude** values for each time or frequency. For detailed information on how to enter data, see *Entering tabular data*.
8. If desired, display the **Baseline Correction** tabbed page.

   When you use an amplitude definition to define an acceleration history in the time domain (a seismic record of an earthquake, for example), the integration of the acceleration record through time may result in a relatively large displacement at the end of the event. This behavior typically occurs because of instrumentation errors or a sampling frequency that is not sufficient to capture the actual acceleration history. In Abaqus/Standard it is possible to compensate for it by using "baseline correction." For more information, see *Baseline Correction in Abaqus/Standard*.
9. Click the arrow to the right of the **Correction** field, and select one of the following options:
   - Select **None** for no baseline correction.
   - Select **Single interval** to treat the entire time of the amplitude definition as a single correction interval.
   - Select **Multiple intervals** to treat the entire time of the amplitude definition as multiple correction intervals. If you select this option, enter the time points defining the different correction intervals (i.e. in the first row enter the time point defining the end of the first correction interval and the beginning of the section correction interval; in the second row enter the time point defining the end of the second correction interval and the beginning of the third correction interval, etc.).
10. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Using periodic data to define an amplitude curve

Select **Periodic** to define the amplitude, a, as a Fourier series:

$$a = A_0 + \sum_{n=1}^{N} [A_n \cos n\omega (t - t_0) + B_n \sin n\omega (t - t_0)] \text{ for } t \geq t_0,$$

$$a = A_0 \quad \text{for } t < t_0,$$

where t_0, N, ω, A_0, A_n, and B_n, $n = 1, 2 \ldots N$, are user-defined constants. For more information, see *Defining Periodic Data*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. In the **Circular frequency** field, enter ω, the circular frequency, in radians per time.
4. In the **Starting time** field, enter t_0.
5. In the **Initial amplitude** field, enter A_0, the constant term in the Fourier series.
6. In the data table, enter values for **A**, the coefficient of the cosine terms, and **B**, the coefficient of the sine terms. For detailed information on how to enter data, see *Entering tabular data*.
7. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Using modulated data to define an amplitude curve

Use the modulated definition method to define the amplitude, a, as

$$a = A_0 + A \sin \omega_1 (t - t_0) \sin \omega_2 (t - t_0) \quad \text{for } t > t_0,$$
$$a = A_0 \quad \text{for } t \le t_0,$$

where A_0, A, t_0, ω_1, and ω_2 are user-defined constants. For more information, see *Defining Modulated Data*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. In the **Initial amplitude** field, enter A_0.
4. In the **Amplitude** field, enter A.
5. In the **Starting time** field, enter t_0.
6. In the **Circular frequency 1** field, enter ω_1.
7. In the **Circular frequency 2** field, enter ω_2.
8. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Using exponential decay data to define an amplitude curve

Use the exponential decay definition method to define the amplitude, a, as

$$a = A_0 + A\exp\left(-\left(t - t_0\right)/t_d\right) \quad \text{for } t \geq t_0,$$
$$a = A_0 \quad \text{for } t < t_0,$$

where A_0, A, t_0, and t_d are user-defined constants. For more information, see *Defining Exponential Decay*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. In the **Initial amplitude** field, enter A_0.
4. In the **Amplitude** field, enter A.
5. In the **Starting time** field, enter the start time of the exponential function, t_0.
6. In the **Decay time** field, enter the decay time of the exponential function, t_d.
7. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Defining a solution-dependent amplitude curve

Use the solution-dependent definition method to calculate amplitude values based on a solution-dependent variable. For more information, see *Defining a Solution-Dependent Amplitude for Superplastic Forming Analysis*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. In the **Initial amplitude** field, enter the initial amplitude value. The amplitude starts with the initial value and is then modified based on the progress of the solution.
4. In the **Min amplitude** field, enter the minimum amplitude value.
5. In the **Max amplitude** field, enter the maximum amplitude value. The maximum value is typically the controlling mechanism used to end the analysis.
6. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Using smooth step data to define an amplitude curve

Use the smooth step definition method to define the amplitude, *a*, between two consecutive data points (t_i, A_i) and (t_{i+1}, A_{i+1}) as

$$a = A_i \quad + \quad (A_{i+1} - A_i)\,\xi^3\left(10 - 15\xi + 6\xi^2\right) \quad \text{for } t_i \le t \le t_{i+1},$$

where $\xi = (t - t_i) / (t_{i+1} - t_i)$. This definition is intended to ramp up or down smoothly from one amplitude value to another. For more information, see *Defining Smooth Step Data*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Time span** field, and specify how you want to define the amplitude as a function of time:
   - Select **Step time** for time that is measured from the beginning of each step.
   - Select **Total time** for total time accumulated over all non-perturbation analysis steps.
3. In the data table, enter the tabular data. For detailed information on how to enter data, see *Entering tabular data*.
4. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Defining a spectrum

Use the spectrum method to define a spectrum to be used in a response spectrum analysis. For more information, see *Specifying a Spectrum*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Specification units** field, and specify the units in which you want to define the spectrum.
   - Select **Displacement**, **Velocity**, or **Acceleration** to define the spectrum in displacement units, velocity units, or acceleration units, respectively.
   - Select **Gravity** to specify an acceleration spectrum in *g*-units.
3. If you selected **Gravity**, enter the value of the acceleration of gravity.
4. In the data table, enter the **Magnitude** (magnitude of the spectrum), **Frequency** (frequency, in cycles per time, at which this magnitude is used), and **Damping** (associated damping, given as a ratio of critical damping) values. For detailed information on how to enter data, see *Entering tabular data*.
5. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Defining a user-defined amplitude curve

Use the user definition method to define the amplitude curve in user subroutine *UAMP* (Abaqus/Standard) or *VUAMP* (Abaqus/Explicit). For more information, see *Defining an Amplitude via a User Subroutine*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. In the **Number of variables** field, enter the number of solution-dependent state variables that must be stored with this amplitude definition.
3. Define the amplitude in user subroutine *UAMP* (for Abaqus/Standard) or *VUAMP* (for Abaqus/Explicit). See the following sections for more information:
   - *Specifying general job settings*
   - *UAMP*
   - *VUAMP*
4. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

Defining a PSD definition

Use the PSD definition method to define the frequency dependence of the random loading in a random response analysis step. For more information, see *Defining the Frequency Functions*.

1. Display the **Edit Amplitude** dialog box as described in *Selecting an amplitude type to define*.
2. Click the arrow to the right of the **Specification units** field, and specify the units in which you want to define the curve.
   - Select **Power** to define the frequency function directly in power units.
   - Select **Decibel** to define the frequency function in decibel units.
   - Select **Gravity (base motion)** if the frequency function will be used to define a base motion in *g*-units. If you choose these units, you must define the acceleration of gravity.
3. If you selected **Decibel** units, enter a value for the **Reference power**.
4. If you selected **Gravity** units, enter a value for the **Reference gravity**.
5. In the data table, enter or import the data values for the function:
   - **Real** and **Imaginary** parts of the function, in decibels or in units^2 per frequency.
   - **Frequency**, in cycles/time, or frequency band number (for **Decibel** units).

   For detailed information on how to enter data, see *Entering tabular data*
6. Toggle on **Specify data in an external user subroutine** if you will provide the PSD function in user subroutine *UPSD*.
7. Click **OK** to save your amplitude definition and to close the **Edit Amplitude** dialog box.

The Analytical Field toolset

Abaqus/CAE provides two types of analytical fields: expression fields and mapped fields.
You can use analytical fields to define spatially varying values for selected properties, loads, interactions, and predefined fields, such as the variation of a pressure over a region in a pressure load. The Analytical Field toolset allows you to create and manage analytical fields in the Property module, Interaction module, or Load module.

In this section:

- *Using the Analytical Field toolset*
- *Using analytical expression fields*
- *Using analytical mapped fields*
- *Displaying symbols for interactions and prescribed conditions that use analytical fields*
- *Displaying symbols to visualize mapping source data*
- *Creating expression fields*
- *Creating mapped fields*

Using the Analytical Field toolset

The Analytical Field toolset allows you to create and manage analytical fields. Analytical fields defined using mathematical expressions are called expression fields. Analytical fields defined using an external data source, such as point cloud data, are called mapped fields. Analytical fields indicate points in space using coordinates of the global coordinate system or of a local coordinate system.

Select **Tools->Analytical Field->Create** from the main menu bar in the Property module, Interaction module, or Load module to create a new analytical field. Select **Edit** from the same menu to change an existing analytical field.

Using analytical expression fields

Analytical expression fields define analytical functions—special types of mathematical functions.

You can use expression fields in many prescribed conditions to define spatially varying parameters. For more information, see the following sections:

- *Using the interaction editors*
- *Using the load editors*
- *Using the boundary condition editors*
- *Using the predefined field editors*

In this section:

- *Building valid expressions*
- *Operations and functions in expressions*
- *Evaluating expression fields*

Building valid expressions

Expression fields describe the variation of a parameter for selected interactions and prescribed conditions (such as a pressure magnitude) at a point in space. You create an expression field by building a mathematical expression in the expression field editor. To locate the expression field editor, select **Tools->Analytical Field->Create** from the main menu bar.

Expression fields indicate points in space using coordinates of the global coordinate system or of a local coordinate system. In the expression field editor, these coordinates are listed as **Parameter Names**. By default, the expression is built using the coordinates of the global coordinate system (**X**, **Y**, and **Z** parameter names). The parameter names that are available depend on the type of coordinate system that you select, as shown in *Table 1*. For cylindrical and spherical coordinate systems, the values for **Th** and **P** are evaluated in radians in the range from $-\pi$ to π.

Table 1: Relationship between coordinate system types and parameter names in the expression field editor.

Coordinate system type	Parameter	Parameter names
Rectangular or Global (default)	*X*-, *Y*-, and *Z*-axes	**X**, **Y**, **Z**
Cylindrical	*R*-, θ-, and *Z*-axes	**R**, **Th**, **Z**
Spherical	*R*-, θ-, and ϕ-axes	**R**, **Th**, **P**

An expression is composed using one or more parameter names and one or more operators. You can build the mathematical expression in the expression window of the editor by selecting the parameter names and operators from lists or by typing the parameter names and operators directly into the expression window. Parameter names and operators are case sensitive. Expressions use the syntax required by Python; entries containing errors will generate standard Python errors. See *Operations and functions in expressions* for information on supported operators.

The following examples demonstrate valid expressions.

Example 1

To define a spatial variation with linear distance along a face, type:

```
4.5 * X + 2.75
```

in the expression window of the editor. `X` is the parameter name representing the distance along the face.

Example 2

To define a more complex spatial variation in a cylindrical coordinate system, type:

```
R + sin(Th*pi/2)
```

in the expression window of the editor.

For detailed information, see *Creating expression fields*.

Operations and functions in expressions

This section lists the operations and functions available in the expression field editor.

Arguments to trigonometric functions are assumed to be in radians in the range from $-\pi$ to π. Proper inline Python statements returning a value, including functions in the math module such as those listed, can be used.

For more information, see the documentation for the math module accessible from the official Python home page (*http://www.python.org*). Select **Help->About Abaqus** from the main menu bar to obtain the Python version used by Abaqus/CAE.

Mathematical operations:

+	Add
–	Subtract
*****	Multiply
/	Divide
%	Return the remainder of integer division.
1/A	Return the reciprocal of **A**.
ceil(A)	Return the ceiling of **A**, the smallest integer value greater than or equal to **A**.
fabs(A)	Return the absolute value.
floor(A)	Return the floor of **A**, the largest integer value less than or equal to **A**.
fmod(A,B)	Return fmod(**A**, **B**), as defined by the platform C library. The intent of the C standard is that fmod(**A**, **B**) be exactly (mathematically; to infinite precision) equal to **A** – *n* * **B** for some integer *n* such that the result has the same sign as **A** and magnitude less than abs(**B**).
frexp(A)[]	Return the mantissa and exponent of **A** as the pair (*m*, *e*). *m* is a float and *e* is an integer such that **A** = *m* * 2 ** *e* exactly. Use the brackets to specify the index of the return value to use in the expression evaluation.
modf(A)[]	Return the fractional and integer parts of **A**. Use the brackets to specify the index of the return value to use in the expression evaluation.

Trigonometric functions:

acos(A)	Return the arccosine.
asin(A)	Return the arcsine.
atan(A)	Return the arctangent.
cos(A)	Return the cosine.
cosh(A)	Return the hyperbolic cosine.
hypot(A,B)	Return the Euclidean norm, sqrt(**A***A + **B***B). This is the length of the vector from the origin to point (**A**, **B**).
sin(A)	Return the sine.
sinh(A)	Return the hyperbolic sine.
tan(A)	Return the tangent.
tanh(A)	Return the hyperbolic tangent.

Power and logarithmic functions:

exp(A)	Return the natural exponential to the power **A**, e^{A}.
ldexp(A,B)	Return **A** * (2^{B}).
log(A)	Return the natural logarithm.
log10(A)	Return the base 10 logarithm.
pow(A,B)	Raise a variable to a power.
sqrt(A)	Return the square root.

Constants:

pi	Mathematical constant .
e	Mathematical constant *e*.

Additional information

- *Building valid expressions*
- *Creating expression fields*

Evaluating expression fields

If you specify an expression field for an interaction or prescribed condition, Abaqus/CAE must evaluate the expression field as the first step in determining the values to submit for the analysis. Expressions are evaluated as Python input; entries containing errors will generate standard Python errors. For cylindrical and spherical coordinate systems, the values for **Th** and **P** are evaluated in radians in the range from $-\pi$ to π. Abaqus/CAE evaluates the expression field when the input file is written.

Abaqus/CAE evaluates the expression field at different locations on the model depending on the type of region that you selected when you created the interaction or prescribed condition. *Table 1* indicates the locations on the model that Abaqus/CAE uses for evaluation of the expression field.

Table 1: Expression field evaluation locations.

Region type	Location of expression field evaluation
Node or vertex	At the node or vertex
Edge	At the midpoint of the edge of each element
Surface or face	Centroid of each element face contained in the region
Cell	Centroid of the element

Next, Abaqus/CAE multiplies the magnitude that you specify for the spatially varying parameter, such as a pressure magnitude, by the evaluated expression at each element or node to determine the final values that are submitted to the analysis. The expression field is applied to magnitudes in the interaction or prescribed condition, including the real and imaginary parts of complex magnitudes. Beam and shell gradient values, such as gradients in temperature, are not affected by the expression field. During the analysis, Abaqus applies any amplitude that you have specified for the interaction or prescribed condition.

Analytical field and datum coordinate system

In Abaqus/CAE, when an analytical field uses a local coordinate system, the field is evaluated based on the transformations with respect to the assembly global coordinate system. The instance assembly transformations are evaluated if the analytical field is created in the Load module and an instance level datum coordinate system (for example, Part-1-1.Datum-1) is selected. The instance assembly transformations are not evaluated if the analytical field is created in the Property module and a part-level datum coordinate system (for example, Datum-1) is selected.

Using analytical mapped fields

Analytical mapped fields allow you to define spatially varying parameter values from an external data source.

In this section:

- *About analytical mapped fields*
- *Point cloud data file formats for mapping*
- *Coordinate system for point cloud mapped fields*
- *Mesh-to-mesh mapping from an output database file*
- *Supported elements for mapping*

About analytical mapped fields

In Abaqus/CAE a mapped field is used to provide the values of selected loads, interactions, properties, etc., at different points in space.
For example, you can define a spatially varying shell thickness or pressure load by providing the thickness or pressure values at different coordinates. Parameter values can be read in from a point cloud data file generated by a third-party CAE application or from an Abaqus output database (`.odb`) file.

Mapped fields allow you to import discrete and discontinuous parameter data and apply them to your Abaqus/CAE model. Abaqus/CAE applies the values to the current model, mapping the input *X*-, *Y*-, and *Z*-coordinates to locations in the model. Abaqus/CAE maps the source data onto the target model, and Abaqus computes the distributed parameter values to be used during the analysis. The parameter values are also called field values, or field data; for example, pressure values at different points on a surface.

When you create a mapped field from point cloud data, you can assign a local coordinate system to the source data region to simplify the three-dimensional definition of points in space. The local coordinate system can be rectangular, cylindrical, or spherical. For example, with a rectangular coordinate system, the *X*-, *Y*-, and *Z*-coordinates are interpreted in that local coordinate system.

Only scalar data values can be used in mapped fields. Each field data value is mapped from the source to the target region as a scalar value. The Abaqus/CAE mapping algorithm is purely geometric, with no physical considerations such as conservative mapping.

Mapped fields can be used to define the properties and attributes shown in *Table 1*.

Table 1: Attributes and properties that support mapped fields.

Category	Attribute/Property
Loads	Body concentration flux
	Body heat flux
	Pressure
	Surface concentration flux
	Surface heat flux
	Surface pore fluid flow
Boundary conditions	Acoustic pressure
	Electrical potential
	Normalized concentration
	Mass flow
	Pore pressure
	Temperature
Predefined fields	Nodal temperature
	Pore pressure
	Saturation
	Void ratio
Interactions	Concentrated film condition
	Surface film condition
	Surface radiation
Other	Density (material density distributions)

Category	Attribute/Property
	Shell thicknesses (element distribution or nodal distribution in shell sections)

The magnitude you specify in the property, load, predefined field, or interaction is used as a multiplier for the mapped field data values. You can also scale the source data coordinates; for example, to account for a mismatch of units (i.e., meters to millimeters).

When you apply a load, interaction, or predefined field using a mapped field, you can display symbols in the viewport to visualize the locations and magnitudes of the field values. However, you must mesh the model first to be able to see these symbols.

Abaqus/CAE provides a set of mapping tolerance controls that allow you to adjust how far source data points may lie from the target points. Depending on how far away each source point is from the nearest node on the meshed model target, Abaqus/CAE must decide whether to use or discard each source point.

Output request frequency time points are not supported in mapped fields.



Note: When a mapped field is used to apply a pressure load, you must request output for the field output variable P to be able to visualize the mapped pressure values in the Visualization module. The output variable name P is automatically changed to PDLOAD during the analysis; see *Abaqus/Standard Output Variable Identifiers*, or *Abaqus/Explicit Output Variable Identifiers*.

Point cloud data file formats for mapping

Point cloud data files must be plain text files in one of two formats: **XYZ** or **Grid**.
A point cloud data file in **XYZ** format must contain the desired field values at a set of coordinates. For a rectangular coordinate system, the points must be given by *X*-, *Y*-, and *Z*-coordinates. If you use a cylindrical or spherical local coordinate system, the appropriate coordinates must be used. The **Grid** format contains field values at points in a three-dimensional grid. Abaqus/CAE interpolates to fill in any missing field values in your grid data files.

In this section:

- *XYZ format*
- *Grid format*

XYZ format

A point cloud data file in **XYZ** format must contain rows of data. For a rectangular coordinate system, each row must consist of *X*-, *Y*-, and *Z*-coordinates followed by the field value at that point. The values in each row can be separated by any combination of spaces, tabs, or commas; each space, tab, or comma is considered a single field delimiter. Comma-separated values (CSV), shown in the following example, is a commonly used format:

```
X1,Y1,Z1,value
X2,Y2,Z2,value
X3,Y3,Z3,value
etc.
```

If you use a cylindrical or spherical local coordinate system, the appropriate coordinates must be given in the data file; see *Coordinate system for point cloud mapped fields*.

Figure 1 shows an example of point cloud data in **XYZ** format that have been imported into the **Edit Mapped Field** dialog box.

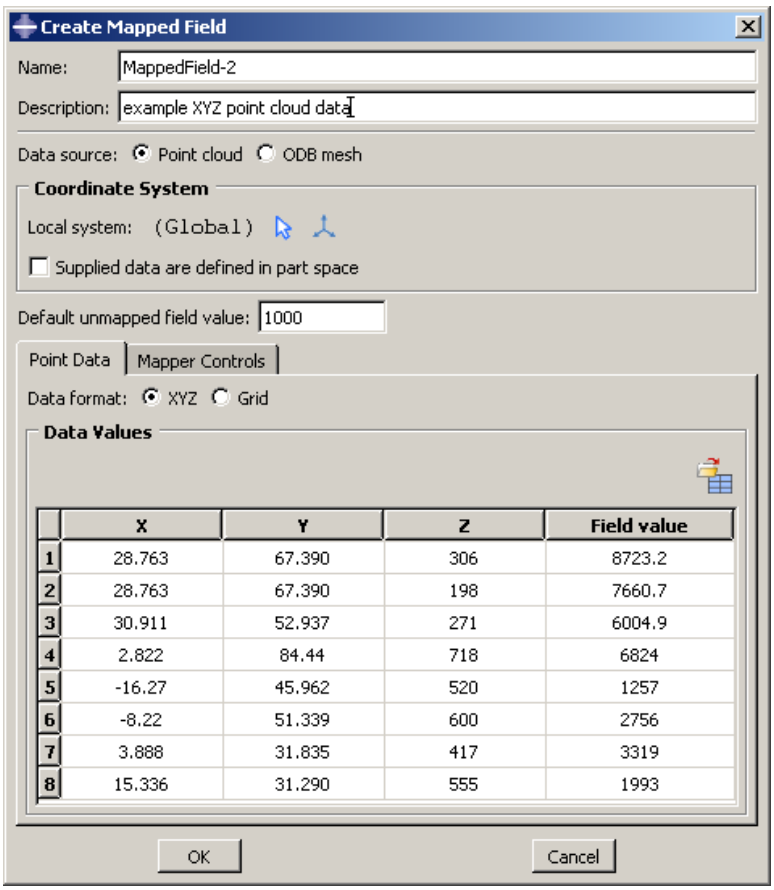
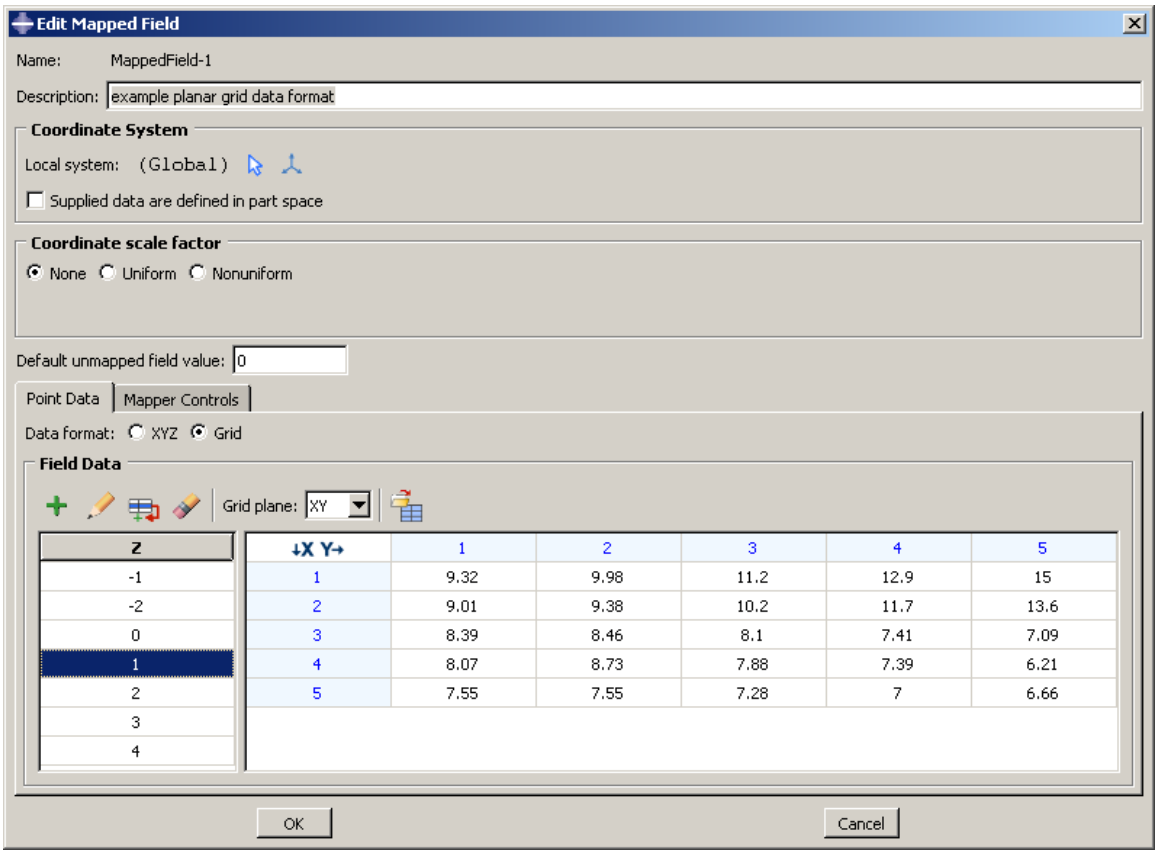


Figure 1: Point cloud data in XYZ format.

Grid format

A point cloud data file in **Grid** format defines the field values at points in a three-dimensional grid. For a rectangular coordinate system, this is a planar or cubic grid. *Figure 1* shows a simple example of point cloud data in **Grid** format that have been imported into the **Edit Mapped Field** dialog box.



Z	↓X Y→	1	2	3	4	5
-1	1	9.32	9.98	11.2	12.9	15
-2	2	9.01	9.38	10.2	11.7	13.6
0	3	8.39	8.46	8.1	7.41	7.09
1	4	8.07	8.73	7.88	7.39	6.21
2	5	7.55	7.55	7.28	7	6.66
3						
4						

Figure 1: Point cloud data in Grid format.

Grid data consist of a set of files, each containing a single plane. For example, you could have one file for the *Z*=3 plane. This file would contain a grid of *X*- and *Y*-coordinates and the field value at each point; for example:

```
a,  -2,     -1,     0,      1,      2
-2, 0.146, 0.141, 0.139, 0.137, 0.131
-1, 0.141, 0.121, 0.116, 0.111, 0.100
 0, 0.139, 0.116, 0.105, 0.101, 0.094
 1, 0.133, 0.129, 0.122, 0.114, 0.107
 2, 0.128, 0.120, 0.111, 0.102, 0.090
```

The *X*-coordinates appear in the first (left) column in this file, and the *Y*-coordinates are in the first (top) row, starting in the second position. The first position contains a dummy value that is ignored by Abaqus/CAE when you import the file into the data table in the **Create Mapped Field** dialog box. In the example file above, the dummy value is the character `a`; you can use any value in this position since it will be ignored.

The values in each row of a grid data file can be separated by any combination of spaces, tabs, or commas; each space, tab, or comma is considered a single field delimiter.

Your complete set of grid data files must include one file for each plane; for example, the *X*–*Y* planes at *Z*=0, *Z*=1, *Z*=2, *Z*=3. The *Y*–*Z*, *X*–*Z*, *Y*–*X*, *Z*–*Y*, and *Z*–*X* planes can also be used. You choose between the different grid planes in the **Create Mapped Field** dialog box.

If your grid data files have any missing field values, Abaqus/CAE interpolates to fill in the missing points; you do not need to fill in these missing values in the **Create Mapped Field** dialog box.

If you use a cylindrical or spherical local coordinate system, the appropriate coordinates must be given in the grid data files; see *Coordinate system for point cloud mapped fields*.

Coordinate system for point cloud mapped fields

Mapped fields indicate points in space using coordinates of the global coordinate system or a local coordinate system. For a point cloud data source, different coordinate systems can be involved when the mapped data coordinates are interpreted and applied to the property, load, predefined field, or interaction. When analytical field values (point cloud data in grid format) referring to a cylindrical or spherical coordinate system are used to map a field variable, Abaqus/CAE converts the values in the cylindrical or spherical coordinate system into corresponding discrete values in the Cartesian coordinate system.

Source data local coordinate system

The three-dimensional coordinates and field values are all defined in the local coordinate system. You can specify the coordinate system to use in the **Create Mapped Field** dialog box. This is an assembly-level (not part-level) coordinate system.

If you use a rectangular coordinate system (default), Abaqus/CAE uses *X*-, *Y*-, and *Z*-coordinates to interpret the source data points. If you specify a cylindrical or spherical coordinate system, Abaqus/CAE interprets the coordinates as shown in *Table 1*. The columns shown in the data table of the **Create Mapped Field** dialog box depend on the type of local coordinate system that you choose. For cylindrical and spherical coordinate systems, the values for **Th** and **P** are entered in degrees and during the analysis are converted to radians in the range from $-\pi$ to π.

Table 1: Coordinate system types and coordinate names for mapped fields.

Coordinate system type	Coordinates	Columns shown in data table
Rectangular or Global (default)	*X*-, *Y*-, and *Z*-axes	**X**, **Y**, **Z**
Cylindrical	*R*-, θ-, and *Z*-axes	**R**, **Th**, **Z**
Spherical	*R*-, θ-, and ϕ-axes	**R**, **Th**, **P**

Source data defined in part space

You can choose this option by toggling on **Supplied data are defined in part space** in the **Create Mapped Field** dialog box. The source data and its local coordinate system are both interpreted in the part-level coordinate system of the target model region.

Mesh-to-mesh mapping from an output database file

When the data source is an Abaqus output database (`.odb`) file, the field output variable values must be displayed in a viewport. This technique is called mesh-to-mesh mapping. The field output variable values can be located at nodes, element centroids, or integration points. In this type of mapping the geometric region of the source data is the mesh connectivity data such as element faces or element sets.

To select the exact source data you want to use, open the desired output database file and display the undeformed contour plot in the Visualization module. You can also use the deformed plot, but the data are still mapped from the undeformed mesh. Choose the field output variable and the step and frame of the analysis for the value you want to map. When you create the new mapped field in your target model in the model database (`.cae`) file, you specify the displayed viewport as the data source. The current state of the viewport is taken as a snapshot of the set of values to be mapped onto the target model; namely, the specific field output variable displayed at a specific step and frame of the analysis.

As an example, you might use mesh-to-mesh mapping in the following workflow:

1. Set up and run a thermal analysis in Abaqus to generate an output database containing nodal temperatures (output variable NT).
2. Open the output database (`.odb`) file in the Visualization module, and display output variable NT at Step-3, Frame 0 of the analysis.
3. Open the model database (`.cae`) file containing your main target model. Select **Tools->Analytical Field->Create** from the main menu bar in the Load module to create a mapped field from the nodal temperature values. In the **Create Mapped Field** dialog box, select the viewport containing the output database as the source data.
4. Mesh (or remesh) your main model.
5. Create a temperature predefined field, and select the mapped analytical field to define the temperature distribution. Abaqus/CAE maps the source data points and their associated temperatures onto points in the target model.
6. Set up and run the subsequent analysis.

The source data on the output database mesh can be located at nodes, integration points, or whole elements. Dissimilar meshes are supported.

The current settings in the selected viewport are used in the mapping. These settings are as follows:

- Primary field output variable
- Step/increment
- Averaging options selected in the **Result Options** dialog box
- Section points (top or bottom)
- Etc.

The following viewport settings are not supported for mesh-to-mesh mapping:

- Display groups
- Transformations
- Averaging by element sets or display groups in the **Result Options** dialog box
- Element face data
- User data in the current session (but data saved in the output database file is supported)

When you create a mapped field from output database data and close the **Create Mapped field** dialog box, the source data are saved to the mapped field object, but the viewport itself is not saved. Any changes made in the viewport after creating the mapped field are not reflected in the mapping data. After you have created a mapped field using

mesh-to-mesh mapping, you cannot edit the source data mesh specifications; you can edit an existing mapped field only to change the default field value or mapper control options.

Volumetric mesh-to-mesh mapping can be performed only with nodal temperatures and material densities. When the mapping target is a mesh-based nodal region, you must specify whether the target is a surface or a volumetric region. Abaqus/CAE uses different mapping algorithms for a surface versus a volumetric region.

Supported elements for mapping

Mapped fields can be used only in models in which supported element types are used in the mesh. The most commonly used elements are supported, including shells.

Supported element classes are listed below, according to the topology of the representative element type: shape, number of nodes, and number of integration points. Any similar element with the same topology is supported for mapping. The supported representative elements are as follows:

- S3, S4, S4R, S8R, STRI65
- C3D4, C3D6, C3D8, C3D8R, C3D10, C3D15, C3D20, C3D20R
- CPE3, CPE4, CPE6, CPE8
- CAX3, CAX4, CAX6, CAX8

The following classes of elements are not supported and cannot be used in a target model mesh or output database source mesh:

- Surface elements
- Rigid elements
- Analytical rigid surface elements
- Cohesive elements
- Composite solid elements
- Continuum shell elements
- Cylindrical elements
- Eulerian elements
- Gasket elements
- Internal elements
- Elements that use modified second-order interpolation
- Elements with twist

Displaying symbols for interactions and prescribed conditions that use analytical fields

When you apply an interaction or a prescribed condition to a region, you can display symbols in the viewport that represent the interaction or prescribed condition.

By default, the symbols for interactions and prescribed conditions that use analytical field distributions are scaled based on the calculated value. For symbols other than arrows, a plus sign (+) or a minus sign (−) is displayed inside each symbol to indicate whether the magnitude of the interaction or prescribed condition is positive or negative at that location. You can turn off the symbol scaling using the **Attribute** display options in the **Assembly Display Options** dialog box. For more information, see *Controlling the display of attributes*.

Abaqus/CAE displays scaled-down symbols for interactions and prescribed conditions when an analytical field evaluates to zero for a portion of its region. These scaled-down symbols are noticeably smaller than the default symbol size. For more information, see *Understanding prescribed condition symbol type, color, and size*.

In this section:

- *Expression field display symbol details*
- *Mapped field display symbol details*

Expression field display symbol details

The points used for symbol display on meshes coincide with the locations where the expression field is evaluated during the analysis. On geometry, when the symbols appear equally spaced along an edge or over the face of a model, Abaqus/CAE selects a random set of points to use for symbol display. The points selected for symbol display may or may not coincide with the locations where the expression field is evaluated.

In some cases the expression field cannot be evaluated at a given point due to an error; for example, division by zero. When this situation occurs, a message appears in the message area indicating that an error occurred while evaluating the expression field. As a visual cue to indicate that there is an error, the symbols displayed in the viewport are no longer scaled, even though the setting to scale symbols based on the expression field value is turned on. You can attempt to eliminate the evaluation error by changing the symbol density settings, which prompts Abaqus/CAE to select a different set of points to use for symbol display. For more information on changing the symbol density settings, see *Controlling the display of attributes*.

Mapped field display symbol details

To be able to see the symbols representing a mapped field, you must first mesh your model before applying the load, interaction, or predefined field.

The points used for symbol display on mapped fields are the native or orphan mesh nodes or elements that will be used in the analysis. If a native mesh is not present, the symbols will be displayed unscaled. In addition, if the mapping fails due to an error (for example, unsupported element types or incorrect tolerances), the symbols will also be displayed unscaled.

In these cases a message appears in the message area indicating an error in the evaluation. As a visual cue to indicate that there is an error, the symbols displayed in the viewport are no longer scaled and are displayed at the points they would be when scaling is turned off. You can try to fix the mapping by remeshing or by changing the symbol density settings, which prompts Abaqus/CAE to select a different set of points to use for symbol display. For more information on changing the symbol density settings, see *Controlling the display of attributes*.

Displaying symbols to visualize mapping source data

You can plot your mapping source data separately from the load, interaction, or predefined field where it is applied. Plotting of source data is supported only for analytical mapped fields created from point cloud data in **XYZ** format.

1. From the Model Tree, expand the **Fields** container and the **Analytical Fields** container to display the mapped field object.
2. Click mouse button 3 on the mapped field object, and select **Plot source data** from the menu that appears.

The displayed symbols represent the locations and relative magnitudes (field values) of the source data points. The magnitudes are color coded using a rainbow color spectrum from red (for the maximum value) to blue (for the minimum value), as shown in *Figure 1*.

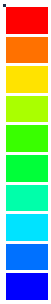


Figure 1: Color coding used for visualizing mapping source data.

Creating expression fields

Select **Tools->Analytical Field->Create** from the main menu bar in the Property module, Interaction module, or Load module to create an analytical field using an expression. Analytical fields defined using mathematical expressions are referred to as expression fields.

1. Open the **Create Expression Field** dialog box using one of the following methods:
   - From the main menu bar, select **Tools->Analytical Field->Create**.

     Tip: You can also click **Create** in the **Analytical Field Manager**.

   - From the interaction editor in the Interaction module, click $f(x)$ next to the **Definition** or **Emissivity distribution** field.
   - From the load, boundary condition, or predefined field editor in the Load module, click $f(x)$ next to the **Distribution** field.
2. Choose **Expression field** as the **Type**, and click **Continue**.
3. In the **Name** text field, enter a name for the expression field. For information on naming objects, see *Using basic dialog box components*.
4. In the **Description** text field, enter a description for the expression field.
5. If you want to change the local coordinate system for the expression field, click **Edit** and use one of the following methods:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.
6. Build your expression in the expression window of the editor using a combination of the following procedures:
   - Select a parameter name from the list on the left side of the editor. The parameter names that are available depend on the type of coordinate system that you select:
     - **Rectangular**: **X**, **Y**, and **Z** are the values on the *X*-, *Y*-, and *Z*-axes, respectively.
     - **Cylindrical**: **R**, **Th**, and **Z** are the values on the *R*-, θ-, and *Z*-axes, respectively.
     - **Spherical**: **R**, **Th**, and **P** are the values on the *R*-, θ-, and ϕ-axes, respectively.

     The parameter name appears in the expression window.
   - Select an operation, function, or constant from the **Operators** list on the right side of the editor. For more information, see *Operations and functions in expressions*.

     The selected operation, function, or constant appears in the expression window.
   - Use standard mouse and keyboard editing techniques in the expression window to position the cursor and configure your expression. Parameter names and operators are case sensitive.
   - Adjust *the syntax of your expression* if necessary; parentheses may be needed.

- Click **Clear Expression** to clear the entire expression in the expression window.

7. Click **OK** to create the expression field and to exit the editor.

Additional information

- *Using the interaction editors*
- *Using the load editors*
- *Using the boundary condition editors*
- *Using the predefined field editors*
- *The Analytical Field toolset*

Creating mapped fields

Select **Tools->Analytical Field->Create** from the main menu bar in the Property module, Interaction module, or Load module to create an analytical mapped field.
Analytical fields defined from an external data source are referred to as mapped fields. The external data source can be either point cloud data files or an Abaqus output database (`.odb`) file. See *Using analytical mapped fields* for an overview of mapped fields.

In this section:

- *Creating mapped fields from point cloud data*
- *Creating mapped fields from output database mesh data*
- *Search controls for mapped fields*

Creating mapped fields from point cloud data

You can create a mapped field by reading parameter values in from a point cloud data file generated by a third-party CAE application. Before creating a mapped field using the procedure below, you must have your point cloud data files prepared and ready to import.

1. Open the **Create Mapped Field** dialog box using one of the following methods:
   - From the main menu bar, select **Tools->Analytical Field->Create**.

     Tip: You can also click **Create** in the **Analytical Field Manager**.

   - From the interaction editor in the Interaction module, click $f(x)$ next to the **Definition** or **Emissivity distribution** field.
   - From the load or predefined field editor in the Load module, click $f(x)$ next to the **Distribution** field.
2. Choose **Mapped field** as the **Type**, and click **Continue**.
3. In the **Name** text field, enter a name for the mapped field. For information on naming objects, see *Using basic dialog box components*. In the **Description** text field, enter a description for the mapped field.
4. Choose **Point cloud** as the **Data source** to import point cloud data files that you have generated from another CAE application. These files must contain your field values and coordinates in either **XYZ** format or **Grid** format, as described in *Point cloud data file formats for mapping*.
5. For a point cloud data source you can optionally assign a local coordinate system to the source data region to simplify the three-dimensional definition of points in space. Click and use one of the following methods:
   - Select an existing datum coordinate system in the viewport.
   - Select an existing datum coordinate system by name.
     1. From the prompt area, click **Datum CSYS List** to display a list of datum coordinate systems.
     2. Select a name from the list, and click **OK**.
   - Click **Use Global CSYS** from the prompt area to revert to the global coordinate system.

   Click to create a new datum coordinate system.

   When you assign a local coordinate system, the source data coordinates will be defined in that coordinate system, which is an assembly-level (not part-level) coordinate system. See *Coordinate system for point cloud mapped fields*, for more information.
6. If desired, toggle on **Supplied data are defined in part space** to indicate that the source data and its local coordinate system are both defined in the part-level coordinate system of the target model region.
7. If desired, choose one of the following to enter a **Scale factor** and to scale the source data coordinates; for example, to account for a mismatch of units (i.e., meters to millimeters):
   - Choose **Uniform**, and enter a single scale factor.
   - Choose **Nonuniform**, and enter a scale factor for the *X*-, *Y*-, and *Z*-coordinates.

8. Enter a value for **Default unmapped field value**, or accept the default of zero. Abaqus/CAE assigns this value at any target points for which it cannot find a source point from which to map. See *Search controls for mapped fields*.
9. On the **Point Data** tabbed page, choose one of the following as the data format:
   - Choose **XYZ** if your data file contains rectangular *X*-, *Y*-, and *Z*-coordinates with corresponding field values. See *XYZ format*, for details about this format. Click  to browse and select your data file to import. For more information about reading data from a file, see *Entering tabular data*.

     If your local coordinate system is cylindrical or spherical rather than rectangular, Abaqus/CAE expects your data file to contain the appropriate coordinate types. See *Coordinate system for point cloud mapped fields*.
   - Choose **Grid** if your data files contain field values defined in a three-dimensional grid of points. See *Grid format*, for details about this format.

     In the **Field Data** table, specify the planes and then import the field values as follows:

     1. Select the grid plane in which your data are organized, such as **XY**, **YZ**, **XZ**, **YX**, **ZY**, or **ZX** for a rectangular coordinate system.
     2. Click  to add a plane in the left frame. Enter the plane height, click **OK**, and enter the field values in the table or click  to import from a file. For example, add a plane at *Z*=2 and then import the field values at various *X*- and *Y*-coordinates within this plane.
     3. Repeat for other plane heights within the same grid plane; for example, at *Z*=3, *Z*=4, etc.

     You can use the following editing tools in the **Data Values** table:

	**Add Plane**
	**Edit Plane**
	**Copy Plane**
	**Delete Plane**

     For more information about reading data from a file, see *Entering tabular data*.

10. On the **Mapper Controls** tabbed page, you can adjust the search tolerance parameters. See *Search controls for mapped fields*.
11. Click **OK** to create the mapped field and to close the dialog box.

When you later apply a load, interaction, or predefined field using a mapped field, you can display symbols in the viewport to visualize the locations and magnitudes of the field values. However, you must mesh the model first to be able to see these symbols.

Additional information

- *Using the interaction editors*
- *Using the load editors*
- *Using the boundary condition editors*

- *The Analytical Field toolset*

Creating mapped fields from output database mesh data

You can create a mapped field by reading parameter values in from an Abaqus output database (`.odb`) file. When the mapping data source is an Abaqus output database, the field values must be displayed in a contour plot in the Visualization module. See *Mesh-to-mesh mapping from an output database file*, for an overview of mesh-to-mesh mapping.

1. When creating a mapped field from an output database, you must have two files open and displayed simultaneously in separate viewports:
   - the model database (`.cae`) file containing your main target model for the mapping, and
   - the output database (`.odb`) file containing field output variable data from a previous analysis run on the same (or similar) model.
2. Select the source data you want to use.
   a. In the Visualization module, open the output database file that contains the source data.
   b. Display the undeformed contour plot in a viewport. You can also use the deformed plot, but the data are still read from the undeformed mesh.
   c. Choose the field output variable for the values you want to map, at a specific step and frame of the analysis. See *Producing a contour plot*, for details about contour plots.
3. In the model database containing the main (target) model, open the **Create Mapped Field** dialog box using one of the following methods:
   - From the main menu bar, select **Tools->Analytical Field->Create**.

     Tip: You can also click **Create** in the **Analytical Field Manager**.

   - From the interaction editor in the Interaction module, click $f(x)$ next to the **Definition** or **Emissivity distribution** field.
   - From the load, boundary condition, or predefined field editor in the Load module, click $f(x)$ next to the **Distribution** field.
4. Choose **Mapped field** as the **Type**, and click **Continue**.
5. In the **Name** text field, enter a name for the mapped field. For information on naming objects, see *Using basic dialog box components*. In the **Description** text field, enter a description for the mapped field.
6. Choose **ODB mesh** as the **Data source**.
7. If desired, choose one of the following to enter a **Scale factor** and to scale the source data coordinates; for example, to account for a mismatch of units (i.e., meters to millimeters):
   - Choose **Uniform**, and enter a single scale factor.
   - Choose **Nonuniform**, and enter a scale factor for the *X*-, *Y*-, and *Z*-coordinates.
8. Enter a value for **Default unmapped field value**, or accept the default of zero. Abaqus/CAE assigns this value at any target points for which it cannot find a source point from which to map. See *Search controls for mapped fields*.
9. On the **ODB Mesh Data** tabbed page, select the **Viewport to map** from the list of open viewports.

Abaqus/CAE uses the current plot state of the selected viewport as the set of values to be mapped onto the target model in your model database (`.cae`) file; namely, the specific field output variable displayed at a specific step and frame of the analysis.

10. On the **Mapper Controls** tabbed page, you can adjust the search tolerance parameters. See *Search controls for mapped fields*.
11. Click **OK** to create the mapped field and to close the dialog box.

After you have created a mapped field using mesh-to-mesh mapping, you cannot edit the source data mesh specifications; you can edit an existing mapped field only to change the default field value or the mapper controls.

When you later apply a load, interaction, or predefined field using a mapped field, you can display symbols in the viewport to visualize the locations and magnitudes of the field values. However, you must mesh the model first to be able to see these symbols.

Additional information

- *Using the interaction editors*
- *Using the load editors*
- *Using the boundary condition editors*
- *The Analytical Field toolset*

Search controls for mapped fields

On the **Mapper Controls** tabbed page of the **Create Mapped Field** dialog box, you can adjust the search tolerance parameters. The search controls are slightly different for point cloud data sources versus output database mesh sources.

Abaqus/CAE attempts to map all of your source data points and their associated field values onto points in the target model. Depending on how far away each source point lies from the nearest node on the meshed model target, Abaqus/CAE must decide whether to use or discard each source point. These decisions are based on the search tolerance distance values you choose on the **Mapper Controls** tabbed page.

Search controls for point cloud mapped fields

The positive and negative normal tolerance values define how far away a source data point can lie from the exterior surface of the target model. The target model surface is interpolated in the undeformed finite element model. By default, each source data point must lie within a distance calculated by multiplying the average element characteristic size in the target model by 0.05. This distance is measured along the positive normal vector of the target surface (see *Figure 1*).

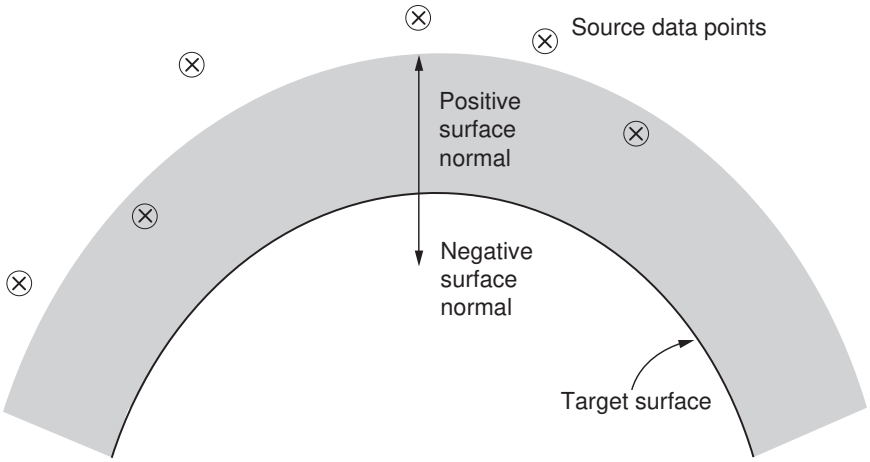


Figure 1: Positive normal search distance.

You can change the tolerance distance values to control which source data points will be included or ignored. You can define the tolerance values as a relative fraction of the average element size in the meshed model or as an absolute distance measured in the length units being used in your model.

For any target points at which Abaqus/CAE cannot map a source point, it will substitute the **Default unmapped field value** entered in the **Create Mapped Field** dialog box. The default is zero. The default field value is applied for both point cloud data sources and output database mesh sources.

For a point cloud data source, the following options and search tolerance values are available on the **Mapper Controls** tabbed page:

Tolerance type

If you choose **Relative** (default), the tolerance values are interpreted as a fraction (not percentage) of the average element characteristic size in the meshed model. If you choose **Absolute**, the tolerance values are taken as exact distances measured in the length units used in your model.

Positive normal search distance tolerance

Any source points within this distance, as measured along the positive normal vector of the underlying geometry of the target surface, are mapped and included in the analysis (see *Figure 1*). This tolerance applies only to

surface (not volumetric) target mapping. The default is 0.05 of the average element characteristic size in the meshed model.

Negative normal search distance tolerance

Any source points within this distance, as measured along the negative normal vector of the underlying geometry of the target surface, are mapped and included in the analysis. This tolerance applies only to surface (not volumetric) target mapping. The default is 0.15 of the average element characteristic size in the meshed model.

Neighborhood search distance tolerance

Any source data points that lie outside the neighborhood search tolerance are ignored and are not used in the analysis. This tolerance applies to both surface and volumetric target mapping.

Abaqus/CAE uses a distance weighting algorithm to interpolate the field data values on the meshed target model. Abaqus/CAE always tries to interpolate the source values on the target. If Abaqus/CAE finds any points for which interpolation is not possible, a distance weighting algorithm is used that will take nodes within the neighborhood search distance. Distance weighting does not consider any other existing tolerances (positive/negative normal search tolerances or boundary search tolerance). You can deactivate distance weighting by using a very small neighborhood search distance, in which case Abaqus/CAE applies the **Default unmapped field value** at these nodes.

Abaqus/CAE makes three passes when attempting to map source points onto target points. Abaqus/CAE:

1. Tries to interpolate field values on the target. The neighborhood search tolerance is ignored on the first pass.
2. Uses the distance weighting algorithm to take source points within the neighborhood search tolerance.
3. For any target points that remain unmapped, Abaqus/CAE applies the **Default unmapped field value**.

Search controls for output database fields

The normal tolerance value defines how far away a source data point can lie from the exterior surface of the target model. By default, each source data point must lie within a distance calculated by multiplying the average element characteristic size in the target model by 0.05. This distance is measured along the normal vector of the target surface. The tolerance values are interpreted as a fraction (not percentage) of the average element characteristic size in the meshed model

You can change the default tolerance values to determine which source data points will be included or discarded.

For any target points at which Abaqus/CAE cannot map a source point, it will substitute the **Default unmapped field value** entered in the **Create Mapped Field** dialog box. The default is zero. The default field value is applied for both point cloud data sources and output database mesh sources.

For an output database source, the following options and search tolerance values are available on the **Mapper Controls** tabbed page:

Normal search distance tolerance

Any source points within this distance, as measured along the positive normal vector of the underlying geometry of the target surface, are mapped and included in the analysis. This tolerance applies to both surface and volumetric target mapping. The default is 0.05 of the average element characteristic size in the meshed model.

Boundary search distance tolerance

This tolerance specifies the in-plane distance within which a source data point must lie, outside the region of the elements of the meshed target model (see *Figure 2*). Increasing this tolerance from the default of 0.01 (of the

average element size) effectively allows you to expand each target element face for the purpose of mapping. Increasing this tolerance may be helpful when the source data are a coarser mesh than the target model mesh.

The field values are interpolated when they are mapped onto the target model. The boundary search tolerance applies to both surface mapping and volumetric mapping.

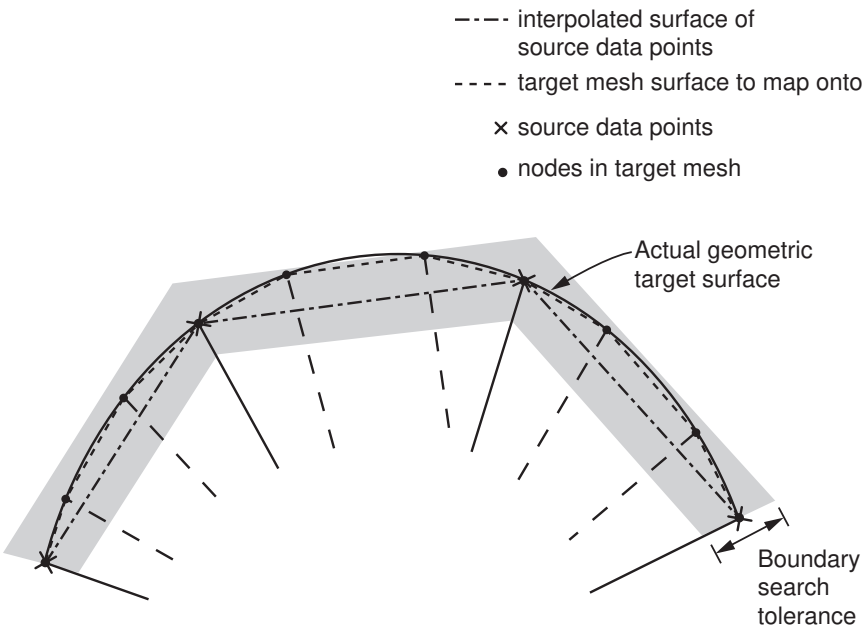


Figure 2: Boundary search distance.

Volume-to-surface mapping control

The **Mapper Controls** tabbed page of the **Create Mapped Field** dialog box includes the **Mapping algorithm for target surface** option, which lets you choose **Surface** or **Volumetric** mapping.

Your target region must be one of three types: a volume, a surface, or a set of nodes in a mesh. If the target is a volume, Abaqus/CAE always performs volumetric mapping. However, if the target is a surface or a set of nodes, you must choose the algorithm that you want Abaqus/CAE to use to apply the source data. Choosing **Surface** causes Abaqus/CAE to use a surface projection algorithm in which it projects the source data to target surface centroids or nodes. Choosing **Volumetric** causes Abaqus/CAE to use a volumetric interpolation algorithm in which it interpolates the source data to target element centroids or nodes. The distinction between the two algorithms only affects how field values from a source volume are applied to a target surface.

For the purpose of the mapping algorithm, a target surface or volume is defined in the context of either a geometric model or a mesh in Abaqus/CAE:

- a volume in a geometric model is a three-dimensional cell
- a volume in a mesh consists of elements
- a surface in a three-dimensional geometric model is a geometric face
- a “surface” in a mesh consists of element faces

Your choice of surface versus volumetric algorithm also applies when you are mapping onto nodes and the target nodes are defined by a node set (mesh-based). However, if the target nodes are defined from model geometry, Abaqus/CAE always uses surface mapping for geometric faces and uses volumetric mapping for geometric cells.

The target region is where you apply the load, interaction, boundary condition, or predefined field in your main model. If you apply the same mapped field source in two different attributes, they may have different behaviors if the target region type is different.

Volume-to-surface control for point cloud data sources

Table 1 describes the cases in which Abaqus/CAE performs volumetric or surface mapping for a point cloud data source and when you must choose between the two.

Table 1: Volumetric vs. surface mapping cases for a point cloud data source.

Point cloud data source	Target region type		
	Geometric cell (3D elements or nodes)	Shell or surface/faces[1]	Node set
Point cloud data	Volumetric	Volumetric or surface	Volumetric or surface
[1] For the purpose of the mapping algorithm, a surface can be a face of a three-dimensional element.			

Volume-to-surface control for output database sources

Table 2 describes the cases in which Abaqus/CAE performs volumetric or surface mapping for an output database source and when you must choose between the two.

Table 2: Volumetric vs. surface mapping cases for an output database source.

Output database source	Target region type		
	Geometric cell (3D elements or nodes)	Shell or surface/faces[1]	Node set
Nodal data	Volumetric	Volumetric or surface	Volumetric or surface
Element-based data[2]	Volumetric	Volumetric or surface	Volumetric or surface
Surface-based data	N/A (error)	Surface	Surface
[1] For the purpose of the mapping algorithm, a surface can be a face of a three-dimensional element.			
[2] Output database averaging controls in the Visualization module affect the continuity of element-based source data.			

If the target region is a shell or surface, the volumetric algorithm interpolates field values from an entire source volume (in the output database) onto the target surface region. Alternatively, the surface algorithm projects only the field values from the surface of the source volume onto the target surface region.

The surface algorithm projects output database source data onto target surface centroids or nodes when the target surface can reference three-dimensional element faces. The volumetric algorithm interpolates source data onto target element centroids or nodes.

The Attachment toolset

The Attachment toolset allows you to create attachment points and attachment lines that can be used to define fasteners and other components in your model.
The Attachment toolset is available in the Part, Property, Assembly, and Interaction modules. This chapter describes the Attachment toolset and editing features created using the toolset.

In this section:

- *Understanding attachment points and lines*
- *Understanding the projection methods*
- *Creating attachment points by picking or reading from a file*
- *Creating attachment points by choosing a direction and a spacing*
- *Creating patterns of attachment points based on edges*
- *Creating attachment lines by projecting points*
- *Editing attachment points and lines*
- *Removing attachment points and lines*

Understanding attachment points and lines

You use the Attachment toolset to add attachment points and lines to your model. You use attachment points to define point-based fasteners, inertia, springs, dashpots, loads or boundary conditions, or connector points for a connector definition in your model; and you use attachment lines to define discrete fasteners. Fasteners model point-to-point connections (such as spot welds, rivets, and bolts) and are described in *About fasteners*. You can create attachment points on the assembly or on a part. You can create attachment lines on only the assembly.

You can use the Attachment toolset to create attachment points and lines by selecting **Tools->Attachment** from the main menu bar and selecting one of the available menu options. In the Interaction module you can use the tools in the module toolbox to create attachment points and lines. The three attachment point tools are also available from the Property module and Part module toolboxes. The Attachment toolset provides the following tools for creating attachment points:

Create attachment points by picking or reading from a file

This tool allows you to create attachment points by picking each point from the viewport or reading the coordinates of the points from a file. For detailed instructions, see *Creating attachment points by picking or reading from a file*.

Create attachment points by choosing a direction and a spacing

This tool allows you to create attachment points by defining a line and specifying the number of points along the line or specifying the spacing of the points along the line. For detailed instructions, see *Creating attachment points by choosing a direction and a spacing*.

Create attachment points by choosing edges and offsets

This tool allows you to create simple patterns of attachment points on edges, on faces, or along directions by selecting a single edge or connected edges and then defining pattern parameters. For detailed instructions, see *Creating patterns of attachment points based on edges*.

If desired, you can move the points to a specified face, as described in *Understanding the projection methods*.

The Attachment toolset provides the following tool for creating attachment lines:

Create attachment lines by projecting points

This tool allows you to create attachment lines by specifying points and then projecting them through multiple faces. Attachment lines connect one or more faces. You can create attachment lines only on the assembly; you cannot create attachment lines on a part. Creating attachment lines is a three-step process:

1. Select the points to project.
2. Project the points onto a source face along the normal to the surface or along a specified direction.
3. From the intersection of the projected points with the source face, project the attachment lines to the target faces along the normal to the source face for a specified distance.

For detailed instructions, see *Creating attachment lines by projecting points*.

Attachment points and lines are features, and they regenerate when you change the model; for example, after you add a cut feature. The regeneration may affect the projection of attachment points and lines and change the total number

of fastening points and lines created by the attachment feature. In addition, the sets that contain attachments may be affected by the regeneration. Therefore, you should determine that your attachment points and lines are behaving as expected after you change the model. You should also check that any prescribed conditions that you have applied to your attachment points and lines, such as loads and boundary conditions, have not changed.

Understanding the projection methods

The Attachment toolset provides three tools for creating attachment points. The tools provide convenient methods for producing points, such as following an edge, reading the coordinates of each point from a file, or spacing points along a line. However, in some cases it may be easier to define the attachment points in one location and project the points onto a desired face. Therefore, Abaqus/CAE provides projection options as a method of positioning attachment points on a face.

The Attachment toolset provides the following techniques to create attachment points by projecting points onto the desired face:

- Project each point to its closest face.
- Choose the start point and end point of a projection vector. Abaqus/CAE projects each point along the vector and creates an attachment point where it first intersects with a face.

Figure 1 illustrates the two techniques for projecting points onto a selected face.

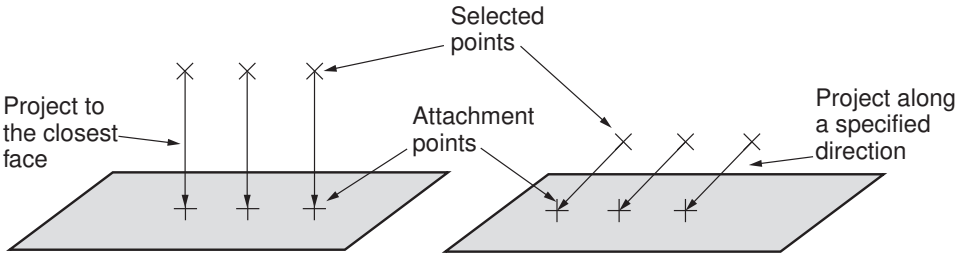


Figure 1: Projecting points onto a selected face.

For more information about how fasteners use attachment points, see *About fasteners*.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Creating attachment points by picking or reading from a file

You can position attachment points by picking each point from the viewport or by reading the coordinates of the points from a file. If desired, you can create the attachment points by projecting the specified points onto a selected face. You can use attachment points to define the location of point-based fasteners. For more information, see *About fasteners*.

1. From the main menu bar, select **Tools->Attachment->Points From File/By Picking**.

   Abaqus/CAE displays the **Create Attachment Points** dialog box.

   Tip: You can also create attachment points by picking or reading from a file using the tool, located with the attachment tools in the Interaction module toolbox or the Property module toolbox. For a diagram of the attachment tools in the toolbox, see *Understanding attachment points and lines*.

2. From the dialog box, use one of the following methods to enter the coordinates of each attachment point in the table:

   - Select to pick a point from the part or assembly in the viewport. You can select any existing point from the part, including vertices, datum points, interesting points, reference points, and orphan mesh nodes. You can also enter the *X*-, *Y*-, and *Z*-coordinates of the point in the text field that appears in the prompt area.
   - Click to read an ASCII file containing the *X*-, *Y*-, and *Z*-coordinates of each point.
   - Click to delete the selected row from the table.

   For more information, see *Entering tabular data*.

   If the attachment points are created on a part, the coordinates you provide are relative to the part's coordinate system. If the attachment points are created on the assembly, the coordinates are relative to the assembly's global coordinate system.

3. To project the specified points onto a selected face, display the **Projection** tabbed page, and do the following:

   a. Toggle on **Project onto faces**.

   b. Click , and select the faces onto which the points will be projected. The faces can be planar or nonplanar.

   c. Select the method for projecting the points.

      - Choose **Proximity** to allow Abaqus/CAE to draw a vector from each point to the closest point on the selected faces.
      - Choose **Direction** to define the vector along which to project the points. Click to choose the **Start point** and **End point** of the vector.

   For more information, see *Understanding the projection methods*.

4. By default, Abaqus/CAE creates a set that will contain the attachment points. If desired, you can modify the set name. Toggle off **Create set with name** if you do not want to create a set containing the attachment points.
5. Click **OK** to create the attachment points.

Abaqus/CAE displays the attachment points. You cannot modify attachment points; you must delete the points and create new points.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Creating attachment points by choosing a direction and a spacing

You can position attachment points by choosing a start point and a direction and by specifying the spacing of the points. If desired, you can create the attachment points by projecting the specified points onto a selected face. You can use attachment points to define the location of point-based fasteners. For more information, see *About fasteners*.

1. From the main menu bar, select **Tools->Attachment->Points Along Direction**.

   Tip: You can also create attachment points by choosing a direction and a spacing using the  tool, located with the attachment tools in the Interaction module toolbox or the Property module toolbox. For a diagram of the attachment tools in the toolbox, see *Understanding attachment points and lines*.

2. From the current viewport, select a point representing the starting point. You can also enter the *X*-, *Y*-, and *Z*-coordinates of the point in the text field that appears in the prompt area.

   Abaqus/CAE displays the **Create Attachment Points Along Direction** dialog box.

3. From the dialog box, use either of the following methods to select a method for specifying the direction along which the points will be created:

   - Choose **End point** and click  to select a point specifying the end point of the direction vector.
   - Choose **Straight line** and click  to select a straight line specifying the direction vector.

4. Do either of the following to specify the spacing of the points along the direction vector:

   - Choose **Number of points between start and end points**, and enter the desired number of points. (This option is available only when you select an end point to specify the direction vector.)
   - Choose **Spacing**, and enter the distance between each attachment point.

     Do either of the following to specify the number of points to create:

     - Choose **Number of points along direction**, and enter the desired number of points. The attachment points can extend beyond the end point of the direction vector. Click  to reverse the direction vector.
     - Choose **Auto-fit points between start and end points** to allow Abaqus/CAE to determine the number of points that can be created between the start and end points with the specified spacing.

5. By default, Abaqus/CAE creates additional attachment points at both ends of the direction vector. If desired, toggle off **At start point** and/or **At end point** to prevent Abaqus/CAE from creating the additional points.
6. To project the specified points onto a selected face, display the **Projection** tabbed page, and do the following:
   a. Toggle on **Project onto faces**.
   b. Click , and select the faces onto which the points will be projected. The faces can be planar or nonplanar.
   c. Select the method for projecting the points.

- Choose **Proximity** to allow Abaqus/CAE to draw a vector from each point to the closest point on the selected faces.

- Choose **Direction** to define the vector along which to project the points. Click  to choose the **Start point** and **End point** of the vector.

For more information, see *Understanding the projection methods*.

7. By default, Abaqus/CAE creates a set that will contain the attachment points. If desired, you can modify the set name. Toggle off **Create set with name** if you do not want to create a set containing the attachment points.
8. Click **OK** to create the attachment points.

   Abaqus/CAE displays the attachment points. For information on editing attachment points, see *Editing attachment points created by choosing a direction and a spacing*.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Creating patterns of attachment points based on edges

You can position attachment points in a simple pattern along a single edge or connected edges. You can then position the points in a pattern of rows on an adjacent face or along a specified direction. If desired, you can project the specified points onto a selected face. You can use the attachment points to define the location of point-based fasteners. For more information, see *About fasteners*.

1. From the main menu bar, select **Tools->Attachment->Points Offset From Edges**.

   Tip: You can also create attachment points by choosing edges and offsets using the  tool, located with the attachment tools in the Property module or Interaction module toolbox. For more information on the attachment tools in the toolbox, see *Understanding attachment points and lines*.

2. From the current viewport, select the edges on which to assign the attachment points. You can select multiple edges as long as the edges are connected and do not branch. You can click **Sets** from the right side of the prompt area to use a previously defined set.

   If the edge selection forms a closed loop, select the starting vertex for point generation from the viewport.

   A red arrow appears to indicate the direction of point generation from the start point.

3. In the prompt area, click **Flip** to reverse the direction of point generation, if necessary, and click **Yes**.

   Abaqus/CAE displays the **Create Attachment Points Offset From Edges** dialog box.

4. From the dialog box, specify the points along the edges.

   - Select **by number** to define the number of attachment points along the edge selection. Abaqus/CAE automatically spaces the points along the edges.

     1. Enter the number of points along the edges.
     2. Enter the desired offset of the first attachment point from the start point.
     3. Enter the desired offset of the last attachment point from the end point.

   - Select **by spacing** to define the distance between the attachment points.

     1. Enter the spacing between the points.
     2. Enter the desired offset of the first attachment point from the start point.
     3. Specify the number of points along the edges, or select **Auto-fit** to have Abaqus/CAE determine the number of attachment points. If you specify the number of points, Abaqus/CAE will create only the points between the start point and the end point; any points extending past the end point will not be created.

5. If desired, define the pattern of rows of the attachment points by specifying the following:

   - Enter the number of rows in the pattern. The default value of 1 creates a single row of attachment points.
   - Enter a value for spacing to define the distance between the rows of the pattern.
   - Enter the offset distance of the first row of the pattern from the edges. The default value of 0 will create the first row of the pattern on the edges.

- Select **Orthogonal to faces** as the offset method, and click  to select an adjacent face on which to position the pattern.
- Select **Along direction** as the offset method, and click  to specify the start and end points of the direction vector.

6. To project the specified points onto a selected face, display the **Projection** tabbed page and do the following:
   a. Toggle on **Project onto faces**.
   b. Click , and select the faces onto which the points will be projected. The faces can be planar or nonplanar.
   c. Select the method for projecting the points.
      - Choose **Proximity** to allow Abaqus/CAE to draw a vector from each point to the closest point on the selected faces.
      - Choose **Direction** to define the vector along which to project the points. Click  to choose the **Start point** and **End point** of the vector.

   For more information, see *Understanding the projection methods*.
7. By default, Abaqus/CAE creates a set that contains the attachment points. If desired, you can modify the set name. Toggle off **Create set with name** if you do not want to create a set containing the attachment points.
8. Click **OK** to create the attachment points.

   Abaqus/CAE displays the attachment points. For information on editing attachment points, see *Editing attachment points created by a pattern based on edges*.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Creating attachment lines by projecting points

Positioning attachment lines is a three-step process:

1. Select the points to project.
2. Project the points onto a source face along a specified direction.
3. From the intersection of the projected points with the source face, project the attachment lines to the target faces along the normal to the source face for a specified distance or to a specified number of target faces.

You can use attachment lines to define the location of discrete fasteners. For more information, see *About fasteners*.

1. From the main menu bar, select **Tools->Attachment->Lines by Projecting Points**.

   Tip: You can also create attachment lines by projecting points using the tool, located with the attachment tools in the Interaction module toolbox. For a diagram of the attachment tools in the toolbox, see *Understanding attachment points and lines*.

2. From the current viewport, select the points to project and click **Done** in the prompt area. You can select vertices, reference points, or attachment points.
3. From the current viewport, select the source faces onto which the points will be projected and click **Done** in the prompt area. An attachment line starts where the direction vector intersects a source surface.
4. From the current viewport, select the target faces that will be intersected by the attachment lines, and click **Done** in the prompt area. Attachment lines are drawn from the start point normal to the source face and end at the furthermost target face.

   Abaqus/CAE displays the **Create Attachment Lines By Projecting Points** dialog box. If desired, you can use the dialog box to edit the points and the source and target faces.

5. From the dialog box, select the method for projecting the points:

   - Choose **Proximity** to allow Abaqus/CAE to draw a vector from each point to the closest point on the selected faces.
   - Choose **Direction** to define the vector along which to project the points. Click to choose the **Start point** and **End point** of the vector.

6. To change the target faces, display the **Target** tabbed page, and do the following:
   a. Abaqus/CAE displays an arrow indicating the direction in which the points on the source faces will be projected to the target faces. Click to reverse the direction of the arrow.
   b. Select the method for determining the length of the projection vector:
      - Choose **Maximum projections along direction**, and enter the maximum number of target faces to be intersected by the attachment lines.
      - Choose **Maximum length of projection vector**, and enter the maximum length of the attachment lines.

7. By default, Abaqus/CAE creates a set that will contain the end points of the attachment lines. If desired, you can modify the set name. Toggle off **Create set with name** if you do not want to create a set containing the end points.
8. Click **OK** to create the attachment lines.

Abaqus/CAE displays the attachment lines. For information on editing attachment lines, see *Editing attachment lines created by projecting points*.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Editing attachment points and lines

You can edit several parameters for attachment points created by choosing a direction and spacing, attachment points created in patterns along or offset from edges, and attachment lines created by projecting points.
You cannot edit attachment points created by picking or reading from a file. Attachment features created in the Part module or Property module can be edited only from those two modules; attachment features created in the Assembly module or Interaction module can be edited only from those two modules. By default, Abaqus/CAE regenerates the model when the edit procedure is finished.

In this section:

- *Editing attachment points created by choosing a direction and a spacing*
- *Editing attachment points created by a pattern based on edges*
- *Editing attachment lines created by projecting points*

Editing attachment points created by choosing a direction and a spacing

You can edit attachment points created by choosing a direction and a spacing.

1. From the main menu bar, select **Feature->Edit**.

    **Tip:** You can also edit a feature by clicking the  tool located in the module toolbox.

2. Select the attachment point feature to modify. You can select the feature either directly from the current viewport or from the Model Tree.

   The **Edit Attachment Points Along Direction Feature** dialog box appears.

3. Select the method for specifying the spacing of the points along the direction vector:

   - Choose **Number of points between start and end points**, and enter the desired number of points. (This option is available only when you select an end point to specify the direction vector.)
   - Choose **Spacing**, and enter the distance between each attachment point.

     Do either of the following to specify the number of points to create:

     - Choose **Number of points along direction**, and enter the desired number of points. The attachment points can extend beyond the end point of the direction vector.
     - Choose **Auto-fit points between start and end points** to allow Abaqus/CAE to determine the number of points that can be created between the start and end points with the specified spacing. (This option is available only when you select an end point to specify the direction vector.)

4. Toggle **At start point** and/or **At end point** to add or remove attachment points at the start and end points. The **At end point** option is available only when you have selected an end point to specify the direction vector.
5. If desired, toggle **Regenerate on OK** to control the model regeneration when the edit procedure is finished.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Editing attachment points created by a pattern based on edges

You can edit attachment points created by a pattern based on edges.

1. From the main menu bar, select **Feature->Edit**.

   Tip: You can also edit a feature by clicking the tool located in the module toolbox.

2. Select the attachment point feature to modify. You can select the feature either directly from the current viewport or from the Model Tree.

   The **Edit Attachment Points Offset From Edges Feature** dialog box appears.

3. Do either of the following to specify the points along the edges:

   - Select **by number** to define the number of attachment points along the edge selection. Abaqus/CAE automatically spaces the points along the edges.

     1. Enter the number of points along the edges.
     2. Enter the desired offset of the first attachment point from the start point.
     3. Enter the desired offset of the last attachment point from the end point.

   - Select **by spacing** to define the distance between the attachment points.

     1. Enter the spacing between the points.
     2. Enter the desired offset of the first attachment point from the start point.
     3. Specify the number of points along the edges, or select **Auto-fit** to have Abaqus/CAE determine the number of attachment points. If you specify the number of points, Abaqus/CAE will create only the points between the start point and the end point; any points extending past the end point will not be created.

4. If desired, define the pattern of rows of the attachment points by specifying the following:

   - Enter the number of rows in the pattern. A value of 1 creates a single row of attachment points.
   - Enter a value for spacing to define the distance between the rows of the pattern.
   - Enter the offset distance of the first row of the pattern from the edges. A value of 0 creates the first row of the pattern on the edges.

5. If desired, toggle **Regenerate on OK** to control the model regeneration when the edit procedure is finished.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Editing attachment lines created by projecting points

You can edit attachment lines created by projecting points.

1. From the main menu bar, select **Feature->Edit**.

    **Tip:** You can also edit a feature by clicking the  tool located in the module toolbox.

2. Select the attachment line feature to modify. You can select the feature either directly from the current viewport or from the Model Tree.

   The **Edit Attachment Lines Feature** dialog box appears.

3. Select the method for determining the length of the projection vector.

   - Choose **Maximum projections along direction**, and enter the maximum number of target faces to be intersected by the attachment lines.
   - Choose **Maximum length of projection vector**, and enter the maximum length of the attachment lines.

4. If desired, toggle **Regenerate on OK** to control the model regeneration when the edit procedure is finished.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

Removing attachment points and lines

You can remove attachment points and lines from your model. The removed attachment points or lines appear in the Model Tree as a **Remove Attachments** feature in the **Features** container under the part or assembly, depending on the module in which the attachment points or lines were created. You can suppress or delete the **Remove Attachments** feature to restore the removed attachment points or lines. Attachment points created in the Part module or Property module can be removed only from these two modules; attachment points and lines created in the Assembly module or Interaction module can be removed only from these two modules.

1. From the main menu bar, select **Tools->Attachment->Remove attachments**.
2. From the current viewport, select the attachment points or lines to remove. You can click **Sets** from the right side of the prompt area to remove a previously defined set.
3. If you are in the Part module or Property module, click **Done** in the prompt area to indicate that you have finished selecting attachment points to remove.

   The selected attachment points are removed from the model and a **Remove Attachments** feature is created in the Model Tree under the part.
4. If you are in the Interaction module or Assembly module, do the following:
   a. In the prompt area, click **Done** to indicate that you have finished selecting attachment points or lines to remove.
   b. In the dialog box that appears, do one of the following:
      - Click **Yes** to continue and remove the selected attachment points or lines from the model.

        A **Remove Attachments** feature is created in the Model Tree under the assembly.
      - Click **No** to return to the viewport to modify the selected attachment points or lines to remove.
      - Click **Cancel** to exit the procedure without removing any attachment points or lines.
5. If desired, select additional attachment features to remove.
6. Click mouse button 2 to exit the remove attachments procedure.

Additional information

- *Understanding attachment points and lines*
- *About fasteners*

The CAD Connection toolset

The CAD Connection toolset is used by the optional add-on associative interfaces for Abaqus/CAE to create a connection from Abaqus/CAE to another CAD system.
You can use the CAD system to modify a model or to change its position, and you can use the established connection to update the model quickly in Abaqus/CAE. You can also modify some geometric features in Abaqus/CAE and use the CAD connection to update the original CAD system model.

In this section:

- *Creating a CAD connection*
- *Updating geometry parameters in an imported model*

Creating a CAD connection

You can use the CAD Connection toolset to create a connection from Abaqus/CAE to another CAD system running an associative interface plug-in.

You can use the connection to export the model from the CAD system to the Assembly module in Abaqus/CAE. You can also use the connection to export certain geometry modifications from an imported model in Abaqus/CAE back to the CAD system (see *Updating geometry parameters in an imported model*). Associative interface plug-ins are available for the following CAD systems:

- CATIA V6
- CATIA V5
- SOLIDWORKS
- Pro/ENGINEER
- NX (Unigraphics)

Figure 1 shows concurrent CATIA V5 and Abaqus/CAE sessions and illustrates the communication between the applications.

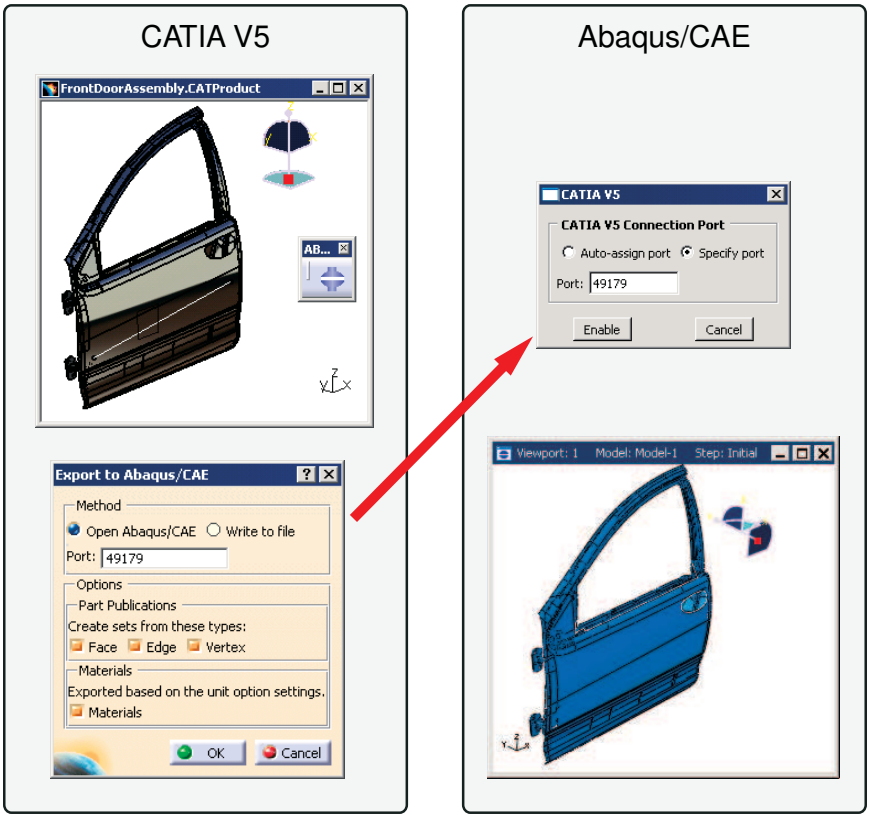


Figure 1: Associative import using the CATIA V5 Associative Interface.

Using a CAD connection to transfer a model from a CAD system running an associative interface plug-in to Abaqus/CAE is called associative import. For further information about the individual associative interface plug-ins, including the CAD packages that support associative import along with the supported version numbers, see the following resources:

- SIMULIA Installation Guide
- *https://www.3ds.com/products-services/simulia/products/abaqus/add-ons/cad-associative-interfaces*
- Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*

1. From the main menu bar in the Assembly module, select **Tools->CAD Interfaces->*CAD System***. ***CAD System*** is the name of the CAD system to which you are connecting; you can establish connections with CATIA V6, CATIA V5, SOLIDWORKS, NX, and Pro/ENGINEER.
2. From the dialog box that appears, do one of the following:

Auto-assign port

Choose **Auto-assign port**, and click **Enable** to open a port from Abaqus/CAE.

Abaqus/CAE displays the port number that it assigned in the message area. When you export a model from the CAD system, the data are transferred through this connection port number.

Specify port

Choose **Specify port**, and enter a port number to specify a port. The number must be between 1025 and 65535. If you are reopening a connection, you can use the port number displayed by Abaqus/CAE when it first assigned the port.

Disable port

If a connection has already been enabled, you can click **Disable** from the dialog box to disable the connection.

3. After you have used the CAD Connection toolset to create a connection from Abaqus/CAE to another CAD system running an associative interface plug-in, you can do the following to export the assembly to the Abaqus/CAE session:

CATIA V6

CATIA V6R2013x or earlier

Select **Tools->Associative Export->Export to Abaqus/CAE** from the main menu bar.

CATIA V6R2014 or later

Select **Share->Export->Associative Export->Export to Abaqus/CAE**.

More detailed instructions are available in the SIMULIA Associative Interface for Abaqus/CAE User's Guide; this guide is available after you install the SIMULIA Associative Interface for Abaqus/CAE app.

CATIA V5

Select **Abaqus->Export to Abaqus/CAE** from the CATIA V5 main menu bar. In the **Export to Abaqus/CAE** dialog box that appears, select **Open Abaqus/CAE**, and click **OK**. More detailed instructions are available in the CATIA V5 Associative Interface User's Guide; you can find information about this guide in the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

SOLIDWORKS

Select **Abaqus->Export to Abaqus/CAE** from the SOLIDWORKS main menu bar. In the **Export to Abaqus/CAE** dialog box that appears, toggle on **Open in Abaqus/CAE**, and click the green check mark. More detailed instructions are available in the SOLIDWORKS Associative

Interface User's Guide; you can find information about this guide in the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

Pro/ENGINEER

Select **Abaqus->Open in CAE** from the Pro/ENGINEER main menu bar. More detailed instructions are available in the Pro/ENGINEER Associative Interface User's Guide; you can find information about this guide in the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

NX

Select **Abaqus->Open in /CAE** from the NX main menu bar. More detailed instructions are available in the Abaqus/CAE Associative Interface for NX User's Guide; you can download this guide from the Elysium, Inc. website.

For more information on associative import, see *What can I do with the associative interfaces?*.

Updating geometry parameters in an imported model

The parameter update capability allows a bidirectional associative import, in which you modify the dimensions of geometric features in a model that has been imported into Abaqus/CAE, then propagate the new dimensions back to the model files from the original CAD system. This bidirectional import feature is currently available only with the CATIA V5 Associative Interface, the CATIA V6 Associative Interface, and the Pro/ENGINEER Associative Interface.

To use the parameter update capability, you must first use the CAD system (CATIA V5, CATIA V6, or Pro/ENGINEER) to indicate which model dimensions can be modified within Abaqus/CAE. The modifiable dimensions must be defined in terms of feature-level parameters, which are imported into Abaqus/CAE along with the model. For example, a parameter may be associated with the radius of a hole in the model; modifying the value of that parameter also modifies the radius of the hole. Each parameter name must begin with the string `ABQ_` to be imported into Abaqus/CAE. In addition, to avoid overwriting an existing parameter with the same name, each parameter name must be unique.

The CATIA V5 Associative Interface, the CATIA V6 Associative Interface, and the Pro/ENGINEER Associative Interface use a special parameters (`.par_abq`) file to pass parameters to Abaqus/CAE. For more information about defining modifiable parameters in an imported CATIA V5, CATIA V6, or Pro/ENGINEER model, refer to the CATIA V5 Associative Interface, the CATIA V6, or the Pro/ENGINEER Associative Interface User's Guides. These guides are available from the Dassault Systèmes Knowledge Base at *http://support.3ds.com/knowledge-base/*.

You can use the **CAD Parameters** dialog box in Abaqus/CAE to assign new values to the imported parameters. When you change a parameter value, the associated geometry feature is regenerated both in the Abaqus/CAE model and in the saved CAD model. This dialog box also enables you to keep the CAD software running in the background after you make a change to CAD parameters. Having the CAD software running in the background improves performance for bidirectional parameter update, but this option occupies a license for your CAD software for the remainder of your session.

1. Do either of the following:
   - Import a model from CATIA V5 using the CATIA V5 Associative Interface.
   - Import a model from CATIA V6 using the CATIA V6 Associative Interface.
   - Import a model from Pro/ENGINEER using the Pro/ENGINEER Associative Interface.
2. Open the **CAD Parameters** dialog box:
   - In the Part module, select **Tools->CAD Parameters**.
   - In the Assembly module, select **Tools->CAD Interfaces->CAD Parameters**.

   The **CAD Parameters** dialog box displays all of the parameters that can be modified.
3. If you are importing parameters from CATIA V5 or Pro/ENGINEER, click on a parameter name to highlight the portion of the model that is impacted by that parameter in the viewport.
4. To modify a parameter value, click the appropriate cell in the **Value** column, and enter a new value.
5. If you are importing parameters from Pro/ENGINEER, you can toggle on **Keep CAD software running in the background after parameter update**. This option improves performance if you perform bidirectional import or make additional updates.
6. When you have changed all of the necessary parameter values, do one of the following:
   - Click **Update** to regenerate the model in Abaqus/CAE based on the new parameter values. The geometry in the saved Pro/ENGINEER part (`.prt`) files is updated as well. The original model must not be open in Pro/ENGINEER when you perform the update.
   - Click mouse button 3, and select **Write to File** to create an updated parameters file without updating the model geometry. This file can be used to manually update parameters using the Abaqus Scripting

Interface. For more information about the parameters file, refer to the Pro/ENGINEER Associative Interface User's Guide.

- Click **Defaults** to reset all of the parameters in the **CAD Parameters** table to the values in the current Abaqus/CAE model.

Additional information

- *What can I do with the associative interfaces?*
- *Creating a CAD connection*

The Customize toolset

The Customize toolset allows you to change the behavior and appearance of some aspects of Abaqus/CAE.

In this section:

- *Configuring the visibility of toolbars*
- *Configuring keyboard shortcuts*
- *Custom toolbars*
- *Configuring icons in custom toolbars*
- *Using the Customize toolset*

Configuring the visibility of toolbars

Using the Customize toolset, you can remove individual toolbars from the Abaqus/CAE display. Hiding toolbars that you use infrequently reduces clutter on the screen. You can always access the functionality in a hidden toolbar using an alternate method (such as the main menu bar, module-specific toolboxes, or keyboard shortcuts).

There are three ways to change the visibility of a toolbar:

Closing a floating toolbar

You can hide a floating toolbar by clicking the “X” in the toolbar's title bar. For an explanation of floating versus docked toolbars, see *Components of the toolbars*.

To resume the display of a floating toolbar, you must use one of the techniques described below.

Using the main menu bar

To hide a visible toolbar, select **View->Toolbars->*Toolbar Name*** from the main menu bar. Use the same procedure to resume the display of a hidden toolbar. Toolbars that are currently being displayed have a check mark next to their names in the menu listing.

Using the Customize dialog box

Using the **Customize** dialog box, you can change the visibility of multiple toolbars at once. Open the **Customize** dialog box by selecting **Tools->Customize** from the main menu bar. On the **Toolbars** tabbed page, toggle off any toolbars you want to hide; toggle on any toolbars you want to make visible.

Configuring keyboard shortcuts

You can use the Customize toolset to edit default keyboard shortcuts, to add new shortcuts for other functions in Abaqus/CAE, and to display a list of all currently defined shortcuts.
A keyboard shortcut is a single keystroke or combination of keys that you can use to launch a function in Abaqus/CAE; for example, **[Ctrl] + `S`** saves the current model.

Several keyboard shortcuts are defined by default for common functions. Select **Tools->Customize** from the main menu bar; display the **Functions** tabbed page of the dialog box that appears to access the **Keyboard** editor.



Tip: Keyboard shortcuts are active only when their function is allowed in the current module and only when the application's focus is not in the Model Tree, Results Tree, message area, or command line interface. If a shortcut does not appear to work, confirm that the function is allowed in the current module or shift focus to the viewport by clicking the viewport title bar.

Abaqus/CAE allows the following keys and key combinations to be specified as keyboard shortcuts:

- Any function key except the `F1` key,
- **[Alt] + [Shift] + `any key`**, or
- **[Ctrl] + `any key`** (except **[Ctrl] + `A`**, which is limited to the select all option in a selection procedure). You can also add **[Alt]** or **[Shift]** to any keyboard shortcut that includes the **[Ctrl]** key.

However, not all keys are eligible for inclusion in a keyboard combination; you cannot specify a keyboard combination that includes a function key or a key from the alphanumeric keypad.

• *Filtering your selection based on the type of object*

Custom toolbars

The Customize toolset allows you to create custom toolbars containing shortcuts to most of the defined functions in Abaqus/CAE. A custom toolbar can be used to collect commonly used tools in a single location, create tools for functions that do not have an existing shortcut, or provide easily accessible controls for a common procedure (for example, the create part, instance part, and mesh part functions could all be part of the same toolbar). Custom toolbars are visible in all modules. If a tool acts as a shortcut to a module-specific function, Abaqus/CAE automatically switches to the appropriate module when that tool is invoked.

The contents and locations of custom toolbars are saved between sessions.

To create a custom toolbar, select **Tools->Customize** from the main menu bar. For detailed instructions on the creation of custom toolbars, see *Creating and modifying custom toolbars*.

Configuring icons in custom toolbars

You can change the default icon that is assigned to tools in custom toolbars. If you create a tool that does not have a default icon assignment, you can assign a new icon to that tool. If a tool appears in more than one custom toolbar, the icon assignment for that tool applies in all custom toolbars. The icons in default Abaqus/CAE toolbars and toolboxes cannot be changed and are not affected by custom icon assignments.

To change or add an icon assignment, select **Tools->Customize** from the main menu bar. On the **Functions** tabbed page of the dialog box that appears, select a tool's function and click **Select** to assign a new icon to that tool.

When assigning an icon, you can select from any of the icons currently used in Abaqus/CAE. You can also import user icons. User icons must be in one of the following formats: bitmap (`.bmp`), GIF (`.gif`), PNG (`.png`), or XPM (`.xpm`). Icon images should be sized appropriately—the standard size for toolbar icons in Abaqus/CAE is 24 × 24 pixels. You cannot scale the size with which user icons are displayed in Abaqus/CAE; if you increase the icon size scale factor (as described in *Scaling the size of icons*), only the Abaqus/CAE icons will increase in size.

Abaqus/CAE creates a directory named `abaqus_icons` in your home directory to store imported user icon images. If you delete an image file from this directory, Abaqus/CAE reverts to the default icon assignment for any tools that use the deleted image file.

The same icon can be assigned to multiple tools. To distinguish the functions of tools using the same icon, place the cursor over a tool for a moment; a small box, or “tooltip,” containing a description of the tool's function will appear.

Using the Customize toolset

All of the functionality of the Customize toolset is accessible through the **Customize** dialog box. Select **Tools**->**Customize** or **View**->**Toolbars**->**Customize** from the main menu bar in any module to open the **Customize** dialog box.

This section explains how to use the features of the **Customize** dialog box.

In this section:

- *Hiding toolbars*
- *Creating, modifying, and removing keyboard shortcuts*
- *Displaying existing keyboard shortcuts*
- *Creating and modifying custom toolbars*
- *Changing icon assignments*
- *Restoring toolbars to their default settings*

Hiding toolbars

Select **Tools->Customize** from the main menu bar to set the visibility of toolbars in Abaqus/CAE. For other methods of hiding toolbars, see *Configuring the visibility of toolbars*.

1. From the main menu bar in any module, select **Tools->Customize**.

   The **Customize** dialog box appears.
2. On the **Toolbars** tabbed page, toggle on the name of any toolbar you want to display; toggle off the name of any toolbar you want to hide.
3. When you have set the visibility for all toolbars as desired, click **Dismiss** to close the **Customize** dialog box.

Additional information

- *Components of the toolbars*
- *Configuring the visibility of toolbars*
- *Using the Customize toolset*

Creating, modifying, and removing keyboard shortcuts

You can assign keyboard shortcuts to particular functions in Abaqus/CAE.
For information on acceptable keystroke combinations, see *Configuring keyboard shortcuts*.

1. From the main menu bar in any module, select **Tools->Customize**.
2. From the **Customize** dialog box that opens, click the **Functions** tab.
3. From the **Module/Toolset** list, select the Abaqus/CAE module or toolset to which your function belongs.

   Abaqus/CAE populates the **Functions** list with the available functions in the selected module or toolset.
4. Select the function for which you want to create, modify, or remove a keyboard shortcut.
5. Perform one of the following steps:

To create or modify a keyboard shortcut:

Enter the function key or key combination that you want to use, then click **Assign**.

The new keyboard shortcut appears in the **Current shortcut** field.

To remove a keyboard shortcut:

Click **Remove**.

Abaqus/CAE deletes the keyboard shortcut assignment and clears the **Current shortcut** field.

Additional information

- *Configuring keyboard shortcuts*
- *Using the Customize toolset*

Displaying existing keyboard shortcuts

The **Shortcut Listing** dialog box displays a listing of all keyboard shortcuts currently assigned in Abaqus/CAE and the functions to which they map.

1. From the main menu bar in any module, select **Tools->Customize**.
2. From the **Customize** dialog box that opens, click the **Functions** tab.
3. To display a list of existing keyboard shortcuts, click **Show all assignments**.

   From the **Shortcut Listing** dialog box that appears, you can review all currently defined keyboard shortcuts.

Additional information

- *Configuring keyboard shortcuts*
- *Creating, modifying, and removing keyboard shortcuts*
- *Using the Customize toolset*

Creating and modifying custom toolbars

You can create new toolbars, change the contents of existing custom toolbars, rename custom toolbars, or delete custom toolbars.
Select **Tools->Customize** to edit custom toolbars in Abaqus/CAE.

Additional information

- *Components of the toolbars*
- *Custom toolbars*
- *Using the Customize toolset*

Create or modify a custom toolbar

1. From the main menu bar in any module, select **Tools->Customize**.
   The **Customize** dialog box appears.
2. To create a new custom toolbar:
   a. On the **Toolbars** tabbed page, click **Create**.
   b. In the dialog box that appears, enter a **Name** for the new toolbar, and click **OK**.
   Abaqus/CAE creates an empty floating toolbar in the main window.
3. Display the **Functions** tabbed page.
4. From the **Module/Toolset** list, select the Abaqus/CAE module or toolset to which your desired function belongs.
   In a custom toolbar, you should use the render style **Functions** available when **Visualization Display Options** is selected in the **Module/Toolset** list instead of those available for **View Manipulation**.
   Abaqus/CAE populates the **Functions** list with the available functions in the selected module or toolset.
5. In the **Functions** list, locate the function that you want to add to the toolbar. Click and drag the function over a custom toolbar; the cursor changes to a plus symbol . Release the mouse button to add the function to the toolbar.
   You can drag functions onto new or existing custom toolbars.
6. If desired, add or modify the icon assignment for the selected function. See *Changing icon assignments*, for details.
7. To remove a function from a custom toolbar, click the icon on the toolbar and drag it off the toolbar; the cursor changes to a minus symbol . Release the mouse button to delete the icon from the toolbar.

   Note:

   You can remove icons from a custom toolbar at any time using this procedure. The **Customize** dialog box does not need to be open.

8. Repeat Steps 4 to 7 until the toolbar contains all of the desired functions.
9. Click **Dismiss** to close the **Customize** dialog box.

Rename a custom toolbar

1. From the main menu bar in any module, select **Tools->Customize**.
   The **Customize** dialog box appears.

2. Display the **Toolbars** tabbed page.
3. Select the toolbar whose name you want to change, and click **Rename**.
4. In the dialog box that appears, enter a new name for the toolbar and click **OK**.
5. Click **Dismiss** to close the **Customize** dialog box.

Delete a custom toolbar

1. From the main menu bar in any module, select **Tools->Customize**.
   The **Customize** dialog box appears.
2. Display the **Toolbars** tabbed page.
3. Select the name of the toolbar that you want to delete, and click **Delete**.
4. Click **Dismiss** to close the **Customize** dialog box.

Changing icon assignments

Select **Tools->Customize** to change the icon that is associated with a particular tool or function on a custom toolbar.

1. From the main menu bar in any module, select **Tools->Customize**.

   The **Customize** dialog box appears.
2. Display the **Functions** tabbed page.
3. From the **Module/Toolset** list, select the Abaqus/CAE module or toolset to which your desired function belongs.

   Abaqus/CAE populates the **Functions** list with the available functions in the selected module or toolset.
4. From the **Functions** list, select the function whose icon you want to change.
5. In the **Icon** field at the bottom right of the dialog box, click **Select**.

   The **Select Icon** dialog box appears.
6. To use an existing Abaqus/CAE icon, select the icon image from the display on the **Abaqus Icons** tabbed page and click **OK**.

   Abaqus/CAE updates the icon for the selected function in all custom toolbars.
7. To use a custom icon:
   a. Display the **User Icons** tabbed page.
   b. To remove a custom icon from the selection display on the **User Icons** tabbed page, select the icon image and click **Delete**.
   c. If the icon you want to use is in the display, skip to step "e."

      If the icon you want is not in the display, click **Import** to import the icon image.
   d. In the **Select an Icon** dialog box that appears, select an image file to use as an icon. See *Configuring icons in custom toolbars*, for information on acceptable icon image files. For information on using the **Select an Icon** dialog box, see *Using file selection dialog boxes*.
   e. Select the icon image from the display on the **User Icons** tabbed page, and click **OK**.

      Abaqus/CAE updates the icon for the selected function in all custom toolbars.

      Note:

      You cannot modify the icons in default Abaqus/CAE toolbars and toolboxes. Nondefault icon assignments apply only to tools in custom toolbars.
8. Click **Dismiss** to close the **Customize** dialog box.

Additional information

- *Custom toolbars*
- *Configuring icons in custom toolbars*
- *Using the Customize toolset*

Restoring toolbars to their default settings

The **Reset** button in the **Customize** dialog box restores all of the toolbars to their original settings. Resetting the toolbars:

- returns all default Abaqus/CAE toolbars to their original visibility settings;
- relocates all default Abaqus/CAE toolbars to their original positions below the main menu bar; and
- deletes all custom toolbars.

These setting do not take effect immediately; you must restart Abaqus/CAE to see the changes. Any changes you make to the toolbar settings between clicking the **Reset** button and restarting Abaqus/CAE will not be saved.

1. From the main menu bar in any module, select **Tools->Customize**.
   The **Customize** dialog box appears.
2. On the **Toolbars** tabbed page, click **Reset**.
3. Restart Abaqus/CAE.

Additional information

• *Components of the toolbars*
• *Using the Customize toolset*

The Datum toolset

A datum is a construction aid that helps you progress through the modeling process when the model itself does not contain the desired geometry. You use the Datum toolset to create these construction aids.
This section explains how you use the Datum toolset to create and position datum points, axes, planes, and coordinate systems.

In this section:

- *Understanding the role of datum geometry*
- *Using the Datum toolset*
- *Why are datum coordinate systems so important?*
- *Understanding a datum as a feature*
- *Datum creation techniques*
- *Creating datum points*
- *Creating datum axes*
- *Creating datum planes*
- *Creating datum coordinate systems*

Understanding the role of datum geometry

As you proceed through the modeling process, you may find that you need a particular piece of geometry, such as a vertex or an edge, that does not exist in your model. You can use the Datum toolset to create the required piece of geometry, which is called a datum.

You can create a datum point, a datum axis, a datum plane, or a datum coordinate system.

- When prompted to select a point, you can select a point from a part, a datum point, or the origin of a datum coordinate system.
- When prompted to select an edge, you can select an edge from a part, a datum axis, or one of the axes of a datum coordinate system.
- When prompted to select a face, you can select a face from a part or a datum plane.
- When prompted to select a coordinate system, you must select a datum coordinate system.

An example of how you might use a datum plane as a sketching plane involves the part shown in *Figure 1*.

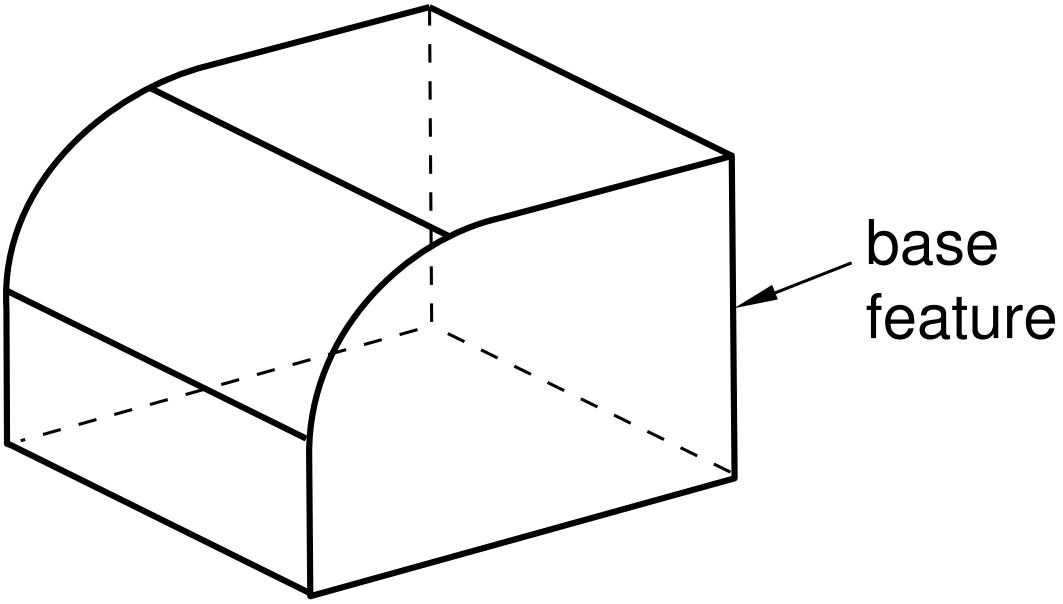


Figure 1: Creating a sketch plane.

To extrude a blind cut into the curved surface, you need to sketch the profile on a plane that is tangent to the curved surface. The desired plane does not exist, but you can create one with the Datum toolset and sketch on the resulting plane. The datum plane and the resulting cut feature are shown in *Figure 2*.

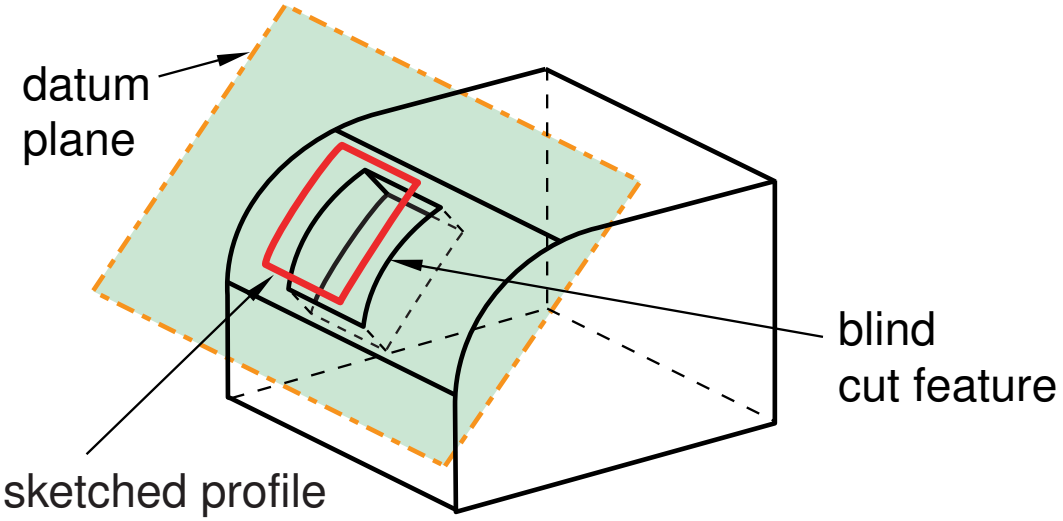


Figure 2: The resulting feature.

Abaqus/CAE does not generate meshes on datum geometry, and datum geometry has no effect on the analysis of your model; it is simply a convenience to help you construct complex geometry. You should use the Partition toolset if you want to add geometry to the model itself, such as a vertex along an edge or a plane through a cell.

You create a datum by defining it with respect to existing geometry (such as vertices, planes, and edges) or to other datum geometry. You can create a datum on a part in the Part or Property module or on an assembly in other modules. For example, you can define a datum axis that passes through two selected points of a part in the Part module, and you can use the axis to align instances of the part in the Assembly module. You can create a datum as a dependent or independent feature; and, like all features, it can be edited, deleted, suppressed, and resumed. A dependent datum is regenerated when the assembly or part is regenerated. You can also use display groups to show or hide datum geometry in a viewport.

Additional information

- *Understanding a datum as a feature*
- *Using display groups to display subsets of your model*

Using the Datum toolset

You can access the Datum toolset in any module from the main menu bar or from the module toolbox.

Select **Tools->Datum** from the main menu bar to display the **Create Datum** dialog box, and choose the type of datum to create—point, axis, plane, or coordinate system (CSYS)—from the **Type** options at the top of the dialog box. The **Method** list changes to reflect the datum you create. Select the desired datum tool from the **Method** list, and follow the prompts in the prompt area to create the datum.

You can also access the Datum toolset from the module toolbox; *Figure 1* shows the hidden icons for all the datum tools in the module toolboxes.

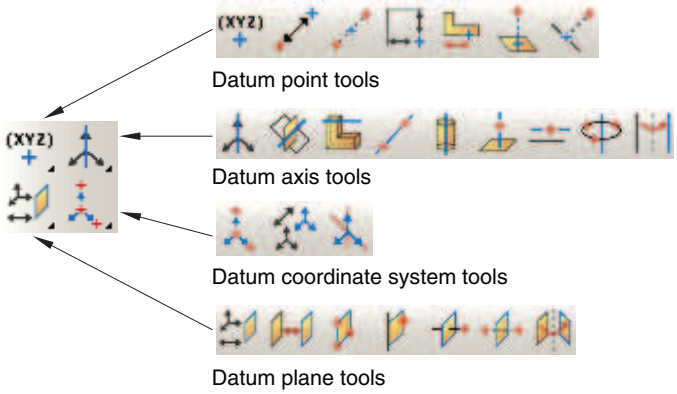


Figure 1: The datum tools.

To see a brief tooltip containing a definition of each datum tool, hold the mouse over the tool for a moment. For information on using toolboxes and selecting hidden icons, see *Using toolboxes and toolbars that contain hidden icons*.

- *Datum creation techniques*
- *Using the Datum toolset in the Part module*
- *Using datum geometry in the Assembly module*
- *Controlling datum display*

Why are datum coordinate systems so important?

Datum coordinate systems are used throughout Abaqus/CAE; for example, to define material and connector orientations; to define the orientation of a coupling constraint, an inertia relief load, and boundary conditions; and to position and align part instances.

In addition, you can select the origin of a datum coordinate system when prompted to select a point, and you can select an axis of a datum coordinate system when prompted to select an edge.

Abaqus/CAE provides three methods for creating a datum coordinate system:

- Three points
- Offset from coordinate system
- Two lines

For more information, see *Methods for creating a datum coordinate system*. In contrast with other types of datums, you can name a datum coordinate system while you create it. As with all datums, you can use the Model Tree to rename a datum coordinate system.

Additional information

• *Understanding a datum as a feature*

Understanding a datum as a feature

A datum is a useful construction aid in the feature-based modeling process and is a feature itself. As a result, you can use the Feature Manipulation toolset to delete, suppress, and resume a datum.

By default, a datum is created as a dependent feature; however, you can choose to make a datum independent when you create it. For an independent datum, no parent-child relationship exists between the datum and any feature used to create it. Once created, a datum cannot be changed from dependent to independent or vice versa. Abaqus/CAE regenerates a dependent datum along with the part or assembly, taking into account any changes to underlying geometry. If you suppress or delete the parent feature of a dependent datum, the datum is also suppressed or deleted.

You can edit any numerical parameters that define a datum; for example, the *X*-, *Y*-, and *Z*-coordinates used to define a datum point. However, a datum is always defined by the same underlying geometry you selected when you created it; if you need to define the datum using different geometry, you must delete the datum and create a new one.

You can create a datum on a part in the Part or Property module or on an assembly in other modules. If you create a dependent datum on a part, the datum moves along with an instance of the part in the Assembly module. If you create a dependent datum on the assembly by selecting entities from a part instance, the datum moves along with the instance in the Assembly module. However, if you create a datum on the assembly by entering coordinates in the prompt area, the datum remains fixed when the instance moves.

When you modify a feature, you should be aware of any parent-child relationships between your dependent datum and the modified feature. For example, consider *Figure 1*, which shows a dependent datum axis passing through the midpoint of two arcs. The midpoints of the arcs that define the location of the datum axis are parents of the datum axis.

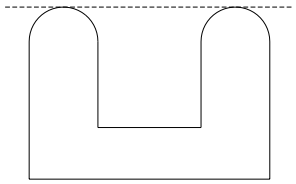


Figure 1: The original datum axis between two midpoints.

If you modify the part, Abaqus/CAE regenerates the datum axis so that it still passes through the two midpoints, as shown in *Figure 2*.

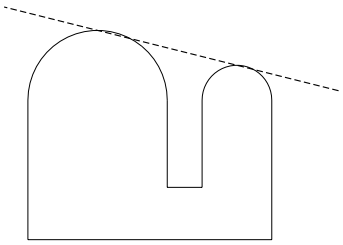


Figure 2: The datum axis after the part is modified.

Additional information

• *The Feature Manipulation toolset*

Datum creation techniques

This section provides an overview of the methods for creating each type of datum.

In this section:

- *Methods for creating a datum point*
- *Methods for creating a datum axis*
- *Methods for creating a datum plane*
- *Methods for creating a datum coordinate system*

Methods for creating a datum point

When you choose **Point** from the **Create Datum** dialog box, the **Method** list displays the methods for creating a datum point.

The following methods are available:

(XYZ) Enter coordinates

Enter the *X*-, *Y*-, and *Z*-coordinates of the datum point, as shown in *Figure 1*. For detailed instructions, see *Creating a datum point by entering its coordinates*.

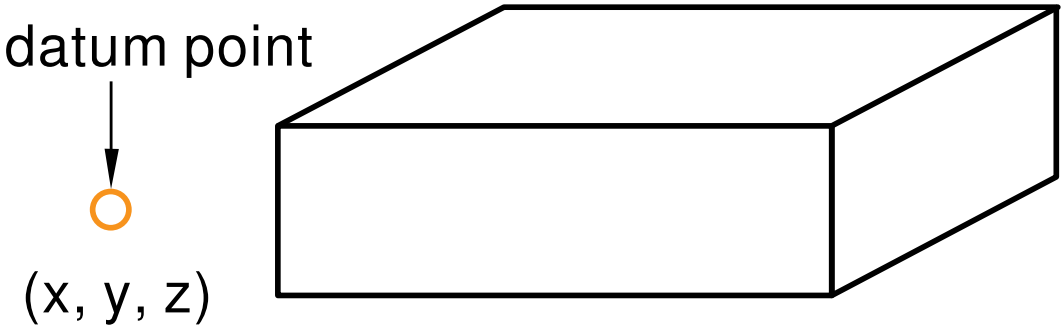


Figure 1: Creating a datum point by entering coordinates.

Offset from point

Enter the location of the datum point in the form of the *X*-, *Y*-, and *Z*-coordinates of an offset from a selected point, as shown in *Figure 2*. For detailed instructions, see *Creating a datum point at an offset from a selected point*.

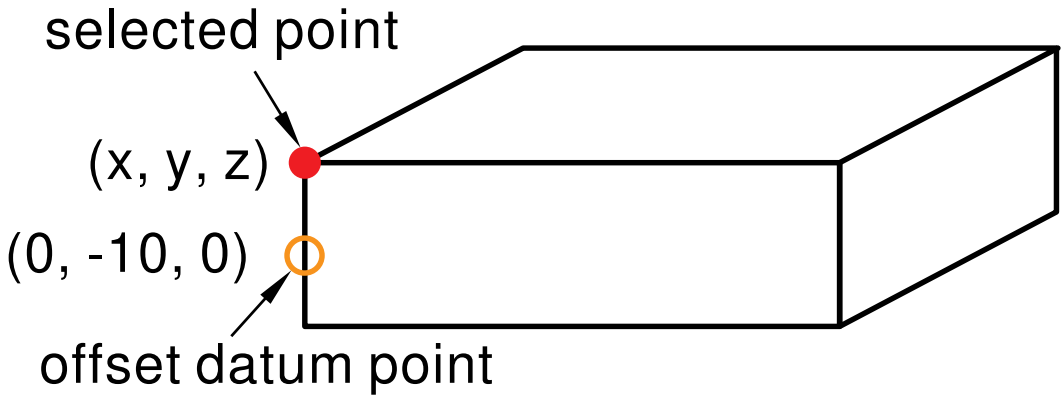


Figure 2: Creating a datum point by offsetting from a point.

Midway between 2 points

Select two points on the model; Abaqus/CAE creates the datum point midway between the two selected points, as shown in *Figure 3*. For detailed instructions, see *Creating a datum point midway between two points*.

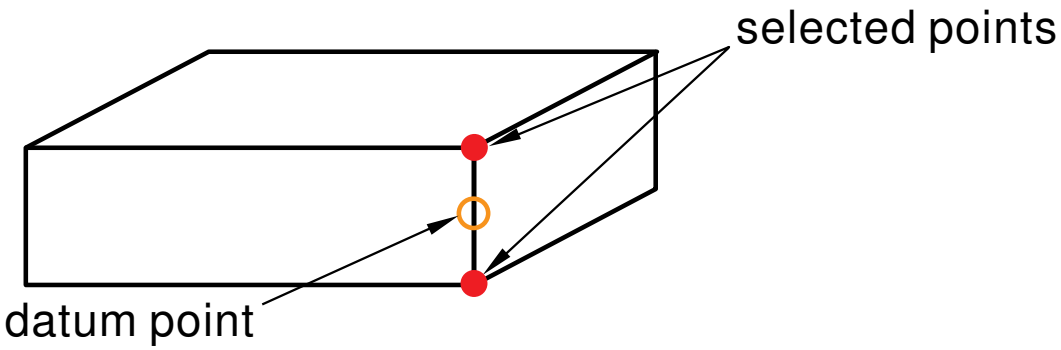


Figure 3: Creating a datum point by selecting two end points.

Offset from 2 edges

Select two edges on the model; and enter the distance from the datum point to each edge, as shown in *Figure 4*. For detailed instructions, see *Creating a datum point at a specified distance from two edges*.

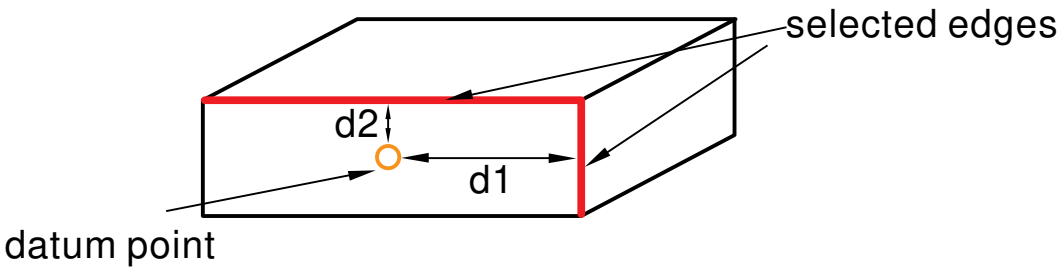


Figure 4: Positioning a datum point a specified distance from two edges.

Enter parameter

Select an edge on the model, and enter the location of the datum point in the form of a parameter value that represents a percentage of the edge length. An arrow along the edge indicates the direction of increasing parameter value from the start vertex (corresponding to an edge parameter value of zero) to the end vertex (corresponding to a value of one), as shown in *Figure 5*. For detailed instructions, see *Creating a datum point by entering an edge parameter*.

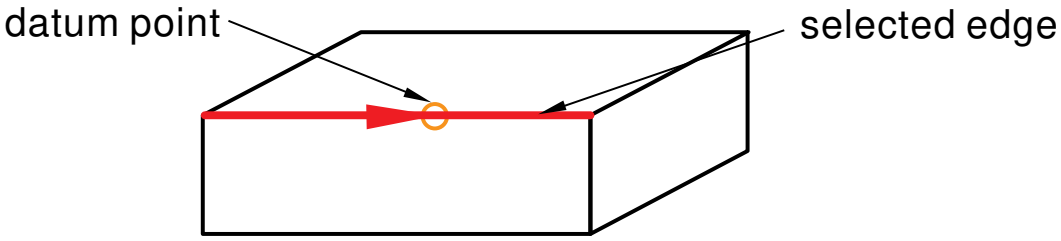


Figure 5: Positioning a datum point a specified distance along an edge.

Project point on face/plane

Select a point and a face or plane on which to project the point. Abaqus/CAE creates the datum point where the face or plane intersects a line that is normal to it and passing through the selected point, as shown in *Figure 6*. The datum point also marks the shortest distance between the selected point and the selected face or plane. For detailed instructions, see *Creating a datum point by projecting a point on a face or plane*.

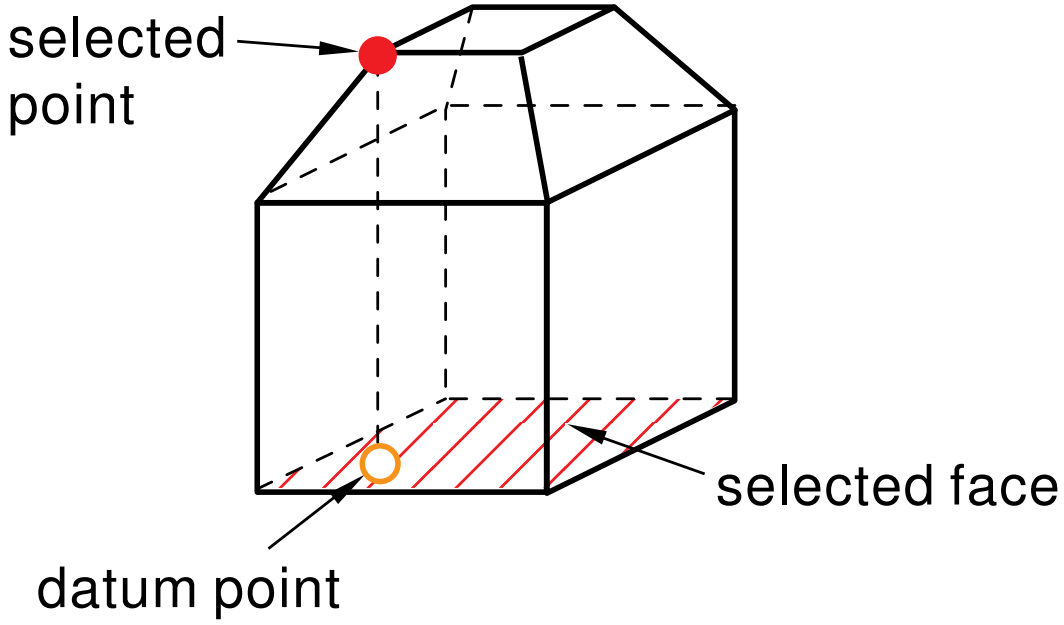


Figure 6: Creating a datum point by projecting a point onto a face.

Project point on edge/datum axis

Select a point on the model and an edge or datum axis on which to project the point. Abaqus/CAE creates the datum point where the edge or datum axis intersects a line that is normal to it and passing through the selected point, as shown in *Figure 7*. The datum point also marks the shortest distance between the selected point and the selected edge or datum axis. For detailed instructions, see *Creating a datum point by projecting a point on an edge or datum axis*.

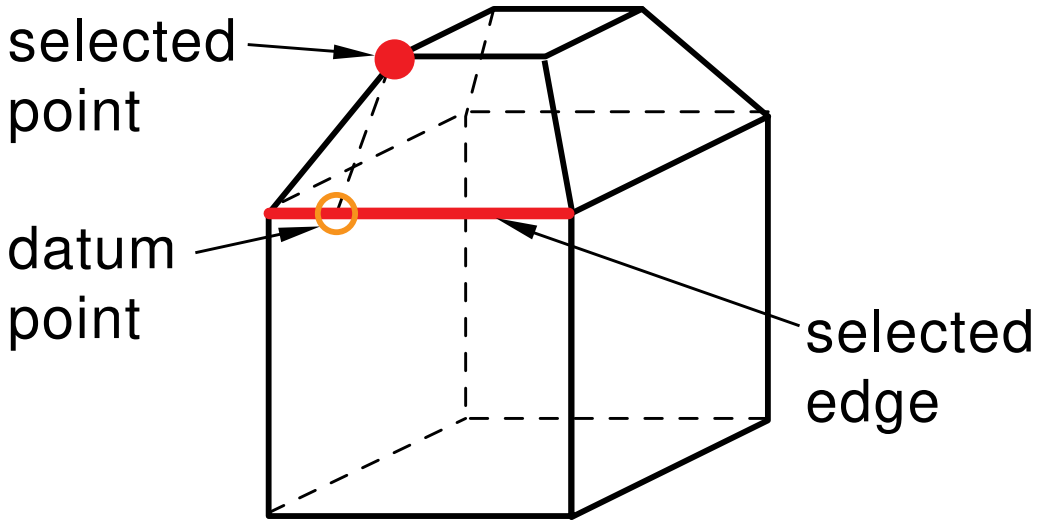


Figure 7: Creating a datum point by projecting a point onto an edge.

Additional information

- *Creating datum points*
- *Controlling datum display*

Methods for creating a datum axis

When you choose **Axis** from the **Create Datum** dialog box, the **Method** list displays the methods for creating a datum axis.

The following methods are available:

Principal axis

Select one of the three principal axes with which the datum axis must be colinear, as shown in *Figure 1*. For detailed instructions, see *Creating a datum axis along a principal axis*.

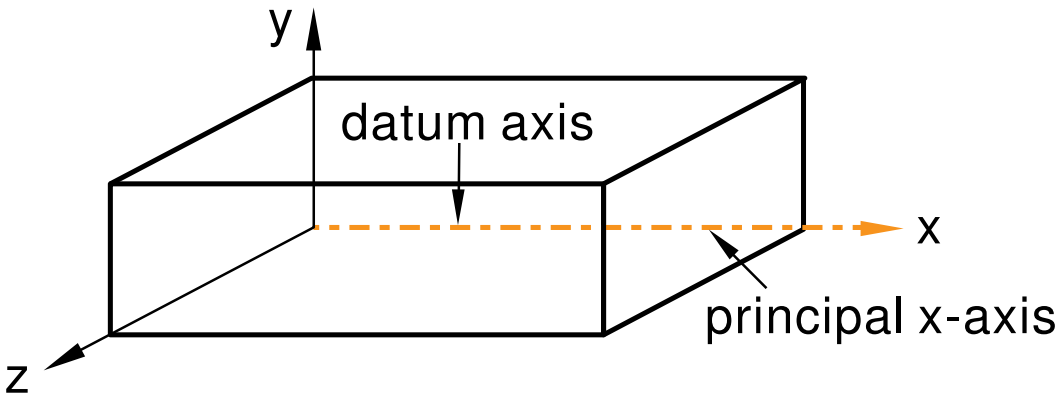


Figure 1: Defining a datum axis as one of the three principal axes.

Intersection of 2 planes

Select two non-parallel planar surfaces. Abaqus/CAE creates the datum axis where the two planes (or extensions of the two planes) intersect, as shown in *Figure 2*. For detailed instructions, see *Creating a datum axis along the intersection of two planes*.

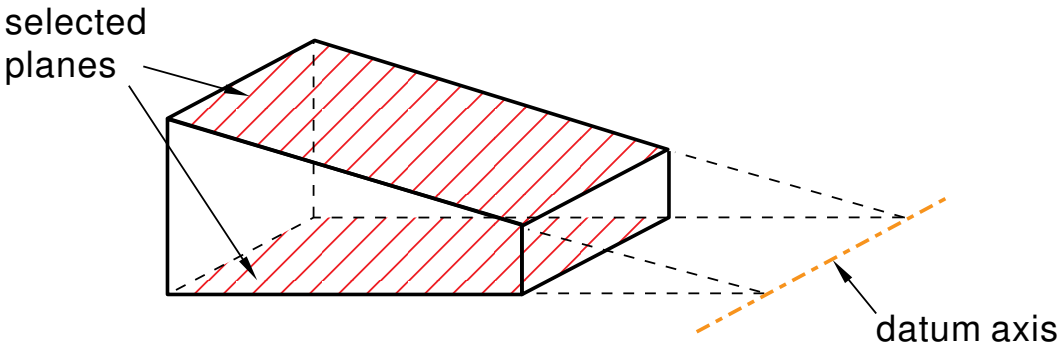


Figure 2: Defining a datum axis as the intersection of two planes.

Straight edge

Select a straight edge on the model with which the datum axis must be colinear, as shown in *Figure 3*. For detailed instructions, see *Creating a datum axis along a straight edge*.

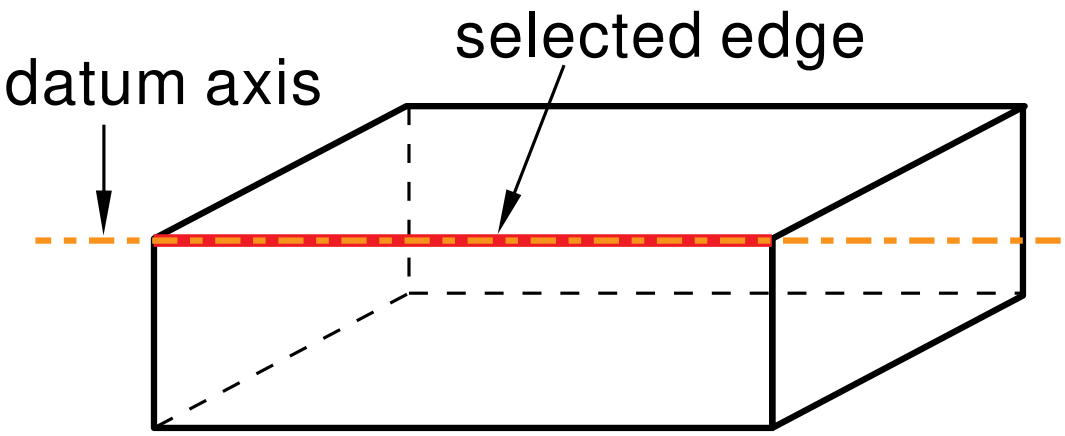


Figure 3: Defining a datum axis as a straight edge on the model.

2 points

Select any two points on the model through which the datum axis must pass, as shown in *Figure 4*. For detailed instructions, see *Creating a datum axis through two points*.

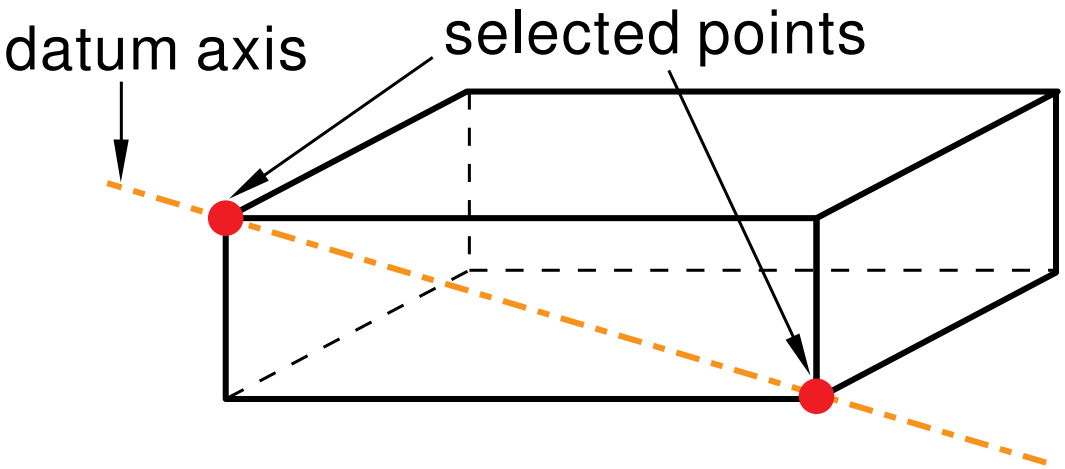


Figure 4: Defining a datum axis by selecting two points.

Axis of cylinder

Select a cylindrical face on the model. Abaqus/CAE creates a datum axis that lies along the axis of the cylindrical face, as shown in *Figure 5*. For detailed instructions, see *Creating a datum axis along the axis of a cylinder*.

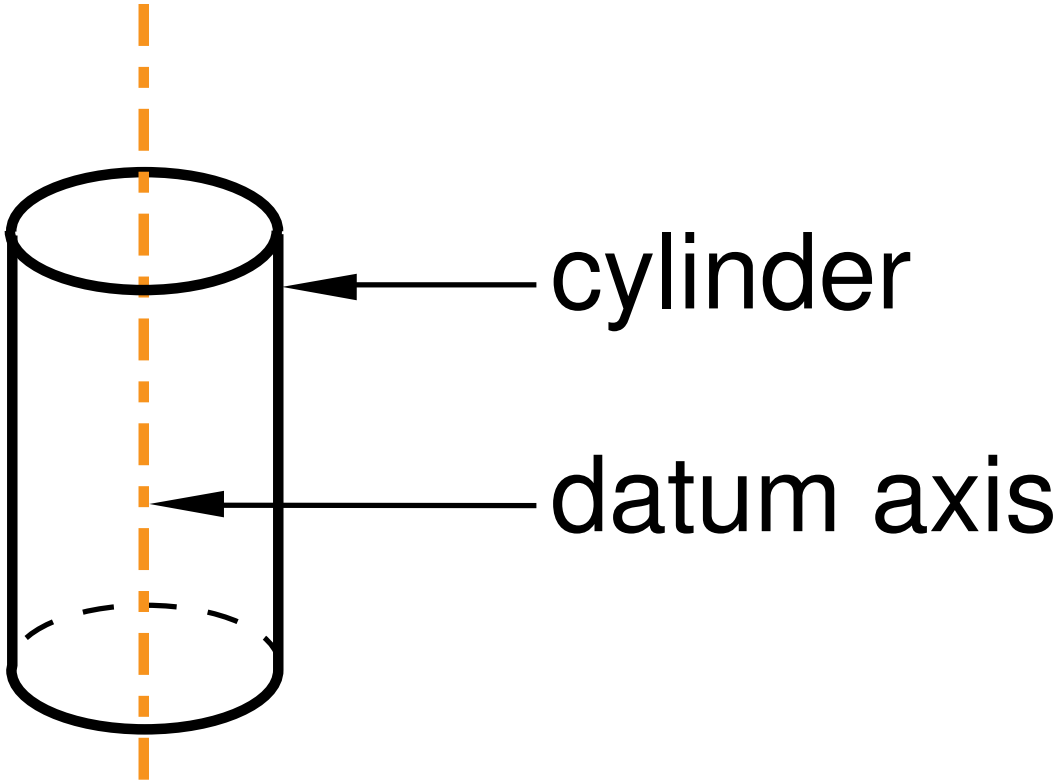


Figure 5: Defining a datum axis as the axis of a cylinder.

Normal to plane, thru point

Select a plane and a point that is not on the plane. Abaqus/CAE creates a datum axis that is normal to the plane and passes through the point, as shown in *Figure 6*. For detailed instructions, see *Creating a datum axis normal to a plane and passing through a point*.

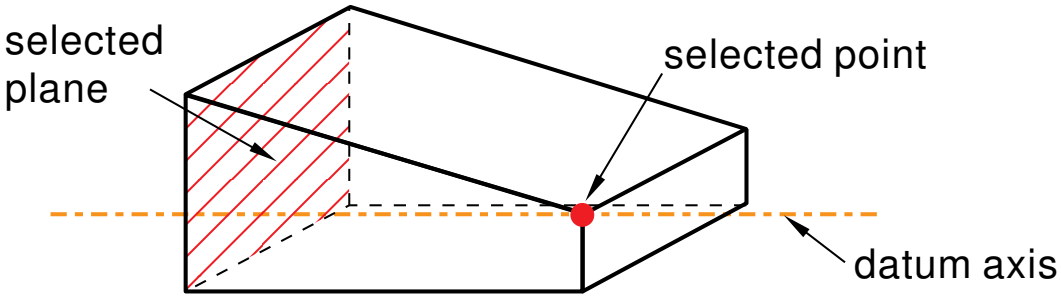


Figure 6: Defining a datum axis by selecting a point and a plane.

Parallel to line, thru point

Select an edge of the model and a point outside the edge. Abaqus/CAE creates a datum axis that is parallel to the edge and passes through the point, as shown in *Figure 7*. For detailed instructions, see *Creating a datum axis parallel to a line and passing through a point*.

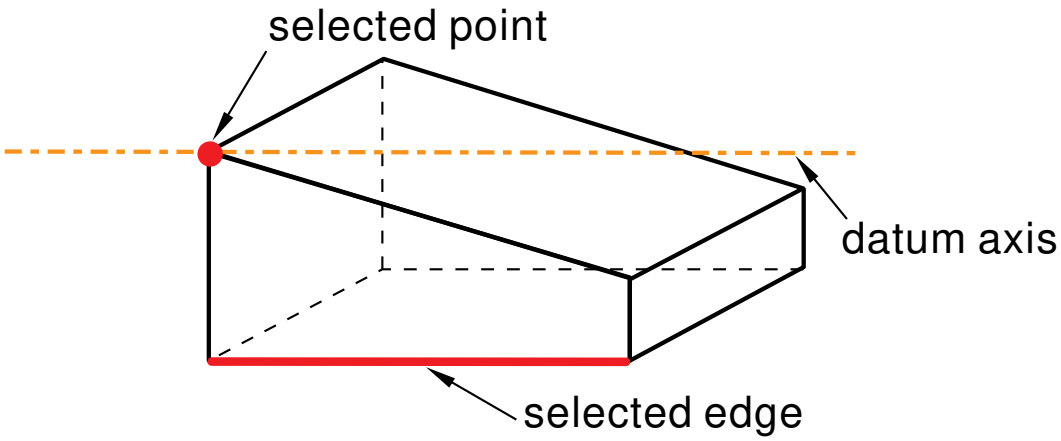


Figure 7: Defining a datum axis by selecting a point and an edge.

3 points on circle

Select three points on the model that define a circle. Abaqus/CAE creates a datum axis along the axis of the circle, as shown in *Figure 8*. For detailed instructions, see *Creating a datum axis running along the axis of a circle defined by three points*.

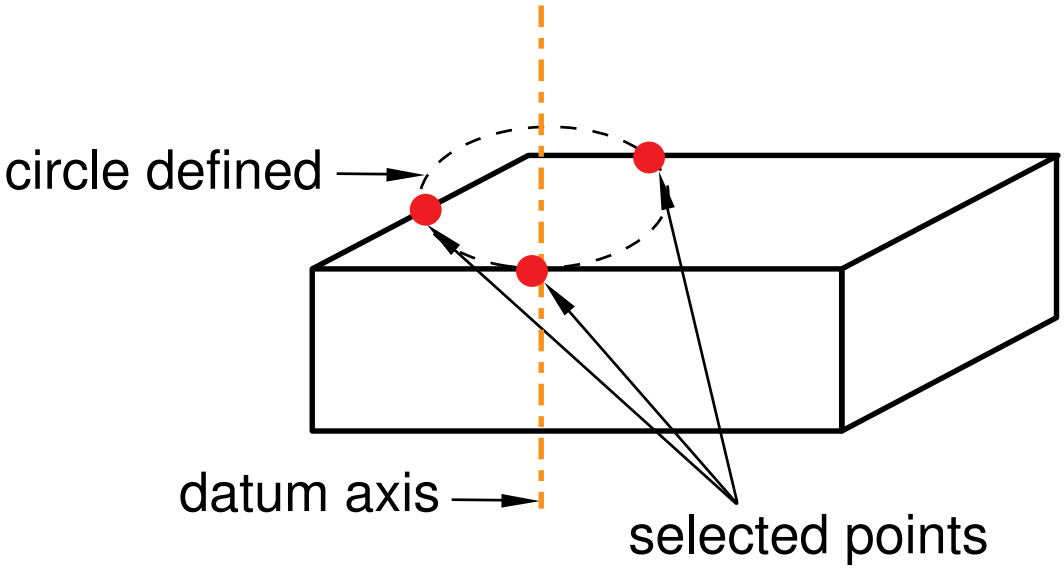


Figure 8: Defining a datum axis as the axis of a circle.

Rotate from line

Select an edge and an axis of rotation, and specify the angle through which the edge will be rotated. Abaqus/CAE creates a datum axis by rotating the edge about the axis through the specified angle, as shown in *Figure 9*. For detailed instructions, see *Creating a datum axis by rotating an existing edge through a specified angle*.

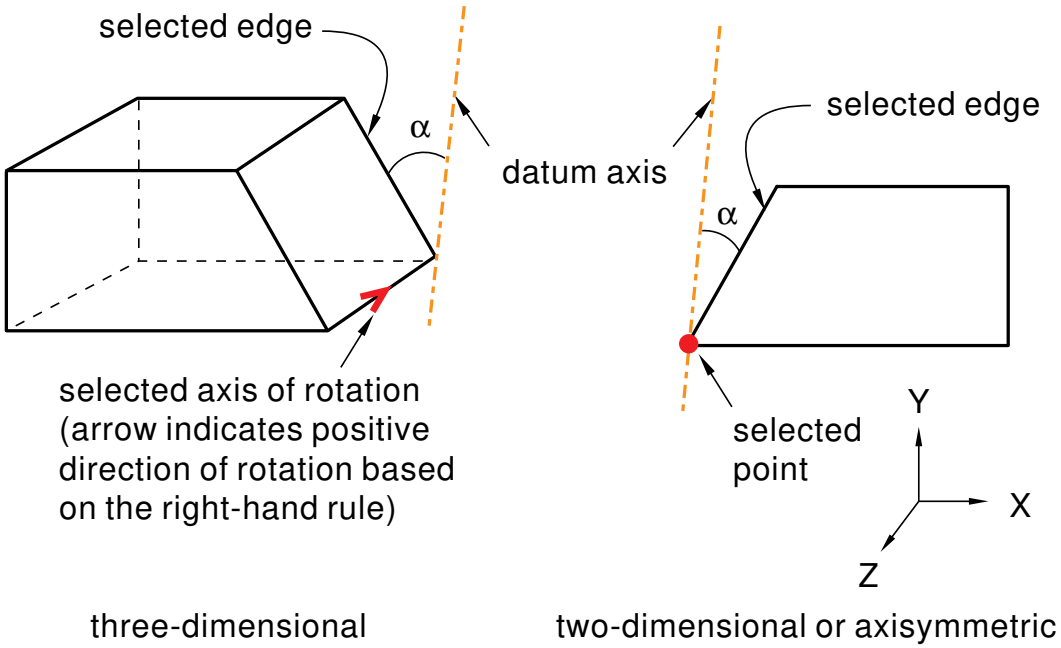


Figure 9: Defining a datum axis by rotating an edge through a specified angle.

Additional information

- *Creating datum axes*
- *Controlling datum display*

Methods for creating a datum plane

When you choose **Plane** from the **Create Datum** dialog box, the **Method** list displays the methods for creating a datum plane.

The following methods are available:

Offset from principal plane

Select one of the three principal planes; and provide the location of the datum plane in the form of an offset from the selected plane, as shown in *Figure 1*. A positive value indicates an offset in the positive direction along the axis normal to the selected plane; for example, along the *X*-axis normal to the *Y*–*Z* plane. For detailed instructions, see *Creating a datum plane offset from a principal plane*.

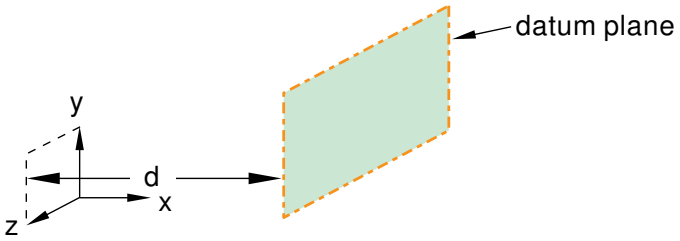


Figure 1: Creating a datum plane by offsetting from one of the three principal planes.

Offset from plane

Select any plane on the model; and provide the location of the datum plane by specifying the direction of the normal and an offset from the selected plane along the normal, as shown in *Figure 2*. You can specify the offset by entering a value or selecting a point. For detailed instructions, see *Creating a datum plane at an offset from a selected plane*.

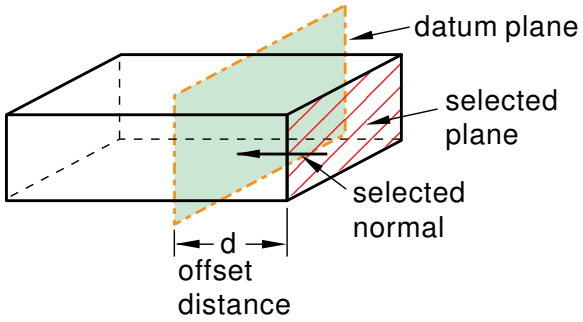


Figure 2: Creating a datum plane by offsetting from any plane.

3 points

Select three points through which the datum plane must pass, as shown in *Figure 3*. For detailed instructions, see *Creating a datum plane passing through three points*.

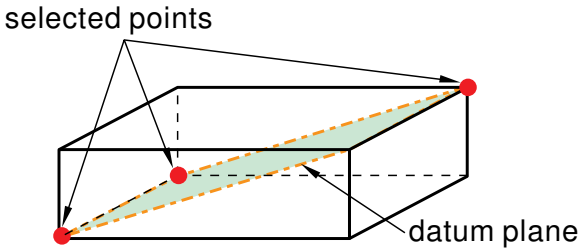


Figure 3: Creating a datum plane by selecting three points.



Line and point

Select an edge and a point through which the datum plane must pass, as shown in *Figure 4*. For detailed instructions, see *Creating a datum plane through a line and a point*.

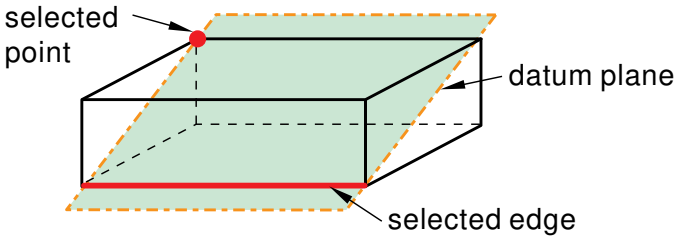


Figure 4: Creating a datum plane by selecting a point and an edge.



Point and normal

Select a point and an edge; the datum plane passes through the point and is normal to the selected edge, as shown in *Figure 5*. For detailed instructions, see *Creating a datum plane passing through a point and normal to an edge*.

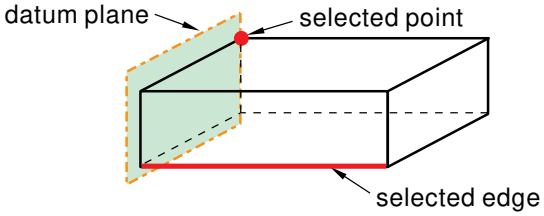


Figure 5: Creating a datum plane by selecting a point and a normal edge.



Midway between 2 points

Select two points. Abaqus/CAE creates a datum plane midway between the two selected points and normal to the line connecting them, as shown in *Figure 6*. For detailed instructions, see *Create a datum plane midway between two points and normal to the line connecting the two points*.

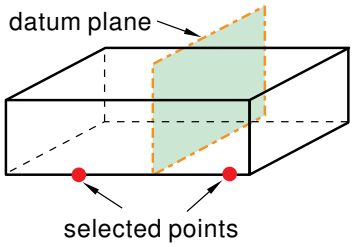


Figure 6: Positioning a datum plane midway between two points.

Rotate from plane

Select a face and an axis of rotation, and specify the angle through which the face will be rotated. Abaqus/CAE creates a datum plane by rotating the face about the axis through the specified angle, as shown in *Figure 7*. For detailed instructions, see *Creating a datum plane by rotating an existing face through a specified angle*.

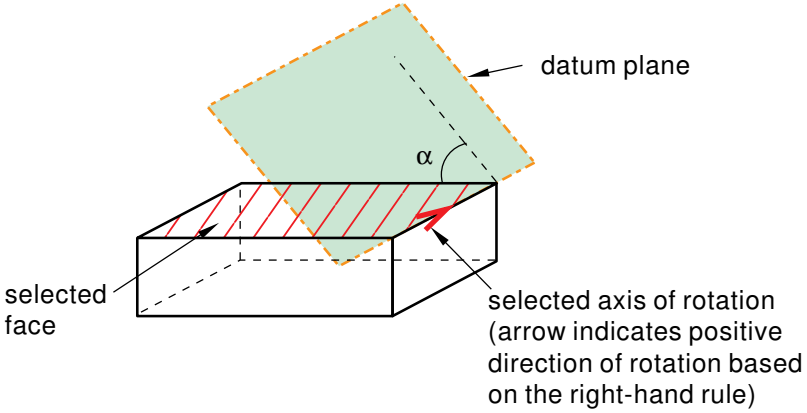


Figure 7: Defining a datum plane by rotating a face through a specified angle.

Additional information

- *Creating datum planes*

Methods for creating a datum coordinate system

When you choose **CSYS** from the **Create Datum** dialog box, the **Method** list displays the methods for creating a datum coordinate system.

Datum coordinates systems are used throughout Abaqus/CAE; for example, to define material orientations and to define connector orientations. To help you keep track of your datum coordinate systems, you can name the systems when you create them, and the name appears alongside their entry in the Model Tree.

The following methods are available:

3 points

Define a rectangular, cylindrical, or spherical coordinate system by selecting the origin and, optionally, two additional points. In the case of a rectangular system, the second point defines the *X*-axis, and the *X*–*Y* plane passes through the second and third points, as shown in *Figure 1*. This is the most versatile tool for creating a datum coordinate system, and you should use it when possible. For detailed instructions, see *Creating a datum coordinate system defined by three points*.



Figure 1: Positioning a rectangular datum coordinate system by selecting the origin and two points.

Offset from CSYS

Select a coordinate system; and provide the location of the rectangular, cylindrical, or spherical datum coordinate system by specifying an offset, as shown by the example involving a rectangular system in *Figure 2*. You can specify the offset by entering a value or by selecting a point. For detailed instructions, see *Creating a datum coordinate system at an offset from another coordinate system*.



Figure 2: Positioning a rectangular datum coordinate system by offsetting from another coordinate system.

2 lines

Select two edges that define the rectangular, cylindrical, or spherical coordinate system. In the case of a rectangular system, the first edge defines the *X*-axis and the *X*–*Y* plane passes through the second edge, as shown in *Figure 3*. For detailed instructions, see *Creating a datum coordinate system defined by two lines*.



Figure 3: Positioning a rectangular datum coordinate system by selecting two edges.

Additional information

- *Creating datum planes*
- *Controlling datum display*

Creating datum points

This section describes the tools that you can use to create a datum point.

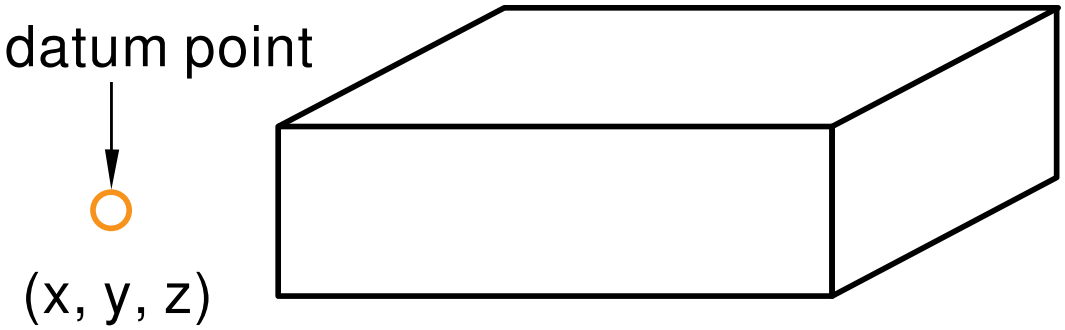
In this section:

- *Creating a datum point by entering its coordinates*
- *Creating a datum point at an offset from a selected point*
- *Creating a datum point midway between two points*
- *Creating a datum point at a specified distance from two edges*
- *Creating a datum point by entering an edge parameter*
- *Creating a datum point by projecting a point on a face or plane*
- *Creating a datum point by projecting a point on an edge or datum axis*

Creating a datum point by entering its coordinates

You can position a datum point by entering its *X*-, *Y*-, and *Z*-coordinates.

The figure below shows an example of creating a datum point by entering its coordinates.



1. From the main menu bar, select **Tools**->**Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the (XYZ) tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Enter coordinates**.
4. In the text field that appears in the prompt area, enter the *X*-, *Y*-, and *Z*-coordinates of the datum point you want to create. If the datum is associated with the assembly, the coordinates you provide are relative to the assembly's global coordinate system. If the datum is associated with a part, the coordinates are relative to the part's coordinate system. If you are unsure of the location or orientation of the part's coordinate system, first create a default coordinate system at the origin of the part. For more information, see *Creating a datum coordinate system defined by three points*.

   The datum point appears. You can modify the coordinates of the datum point by selecting **Feature**->**Edit** from the main menu bar and selecting the datum.

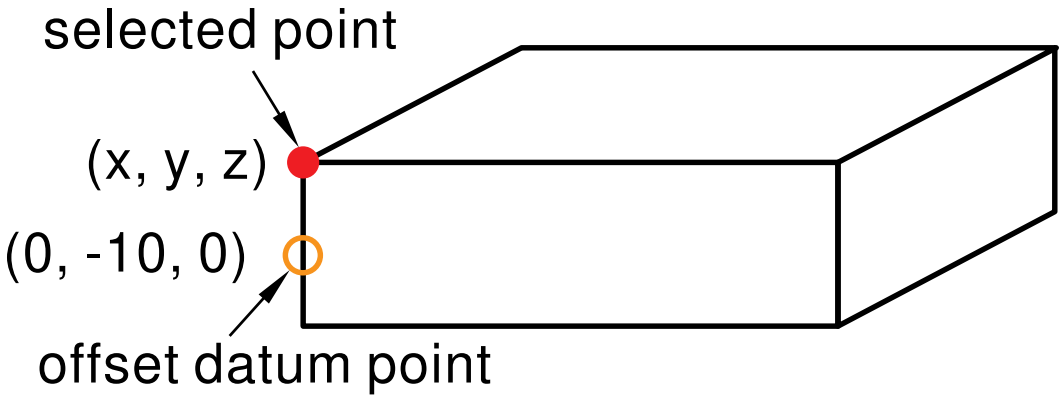
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point at an offset from a selected point

You can position a datum point by selecting a point on the model and entering the *X*-, *Y*-, and *Z*-coordinates of an offset from the point.

The figure below shows an example of creating a datum point at an offset from a selected point.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Offset from point**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a point.
6. In the text field that appears in the prompt area, enter the *X*-, *Y*-, and *Z*-coordinates of the offset from the selected point.

   The datum point appears. You can modify the coordinates of the offset by selecting **Feature->Edit** from the main menu bar and selecting the datum.

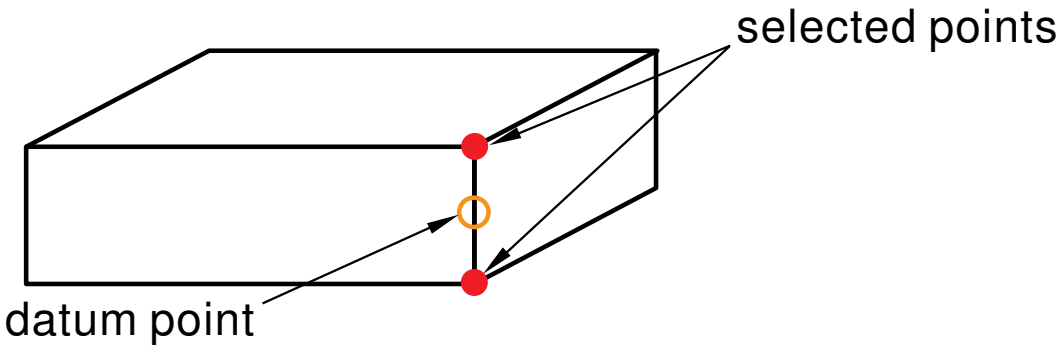
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point midway between two points

You can position a datum point by selecting two points on the model; Abaqus/CAE creates the datum point midway between the two selected points.

The figure below shows an example of creating a datum point midway between two points.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Midway between 2 points**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select two points.

   The datum point appears. You cannot modify a datum point created with this method; you must delete the old point and create a new one.

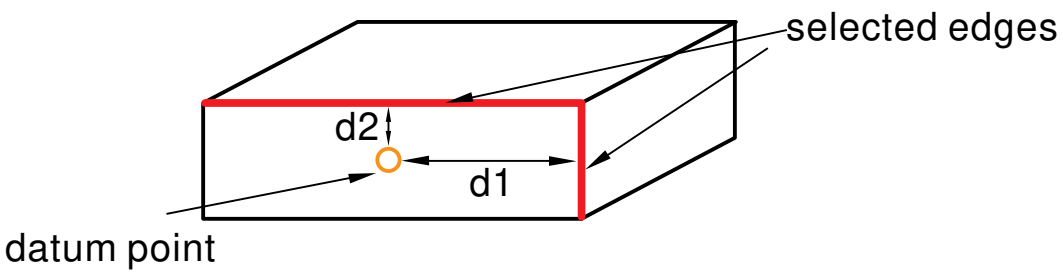
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point at a specified distance from two edges

You can position a datum point by selecting two edges on the model and entering the distance from the datum point to each edge.

The figure below shows an example of creating a datum point at a specified distance from two edges.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Offset from 2 edges**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select the plane on which to create the datum point.
6. Select a straight edge on the specified plane and enter the distance of the datum point from this edge.
7. Select a second straight edge on the plane and enter the distance of the datum point from this edge.

   The datum point appears. You can modify the position of the datum point by selecting **Feature->Edit** from the main menu bar and modifying the distance from the datum to each edge.

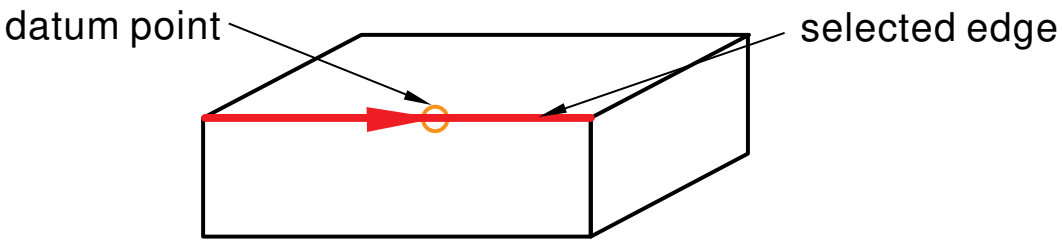
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point by entering an edge parameter

You can position a datum point by selecting an edge on the model and entering a parameter value that represents a percentage of the edge length.
An arrow along the edge indicates the direction of increasing parameter value from the start vertex (corresponding to an edge parameter value of zero) to the end vertex (corresponding to a value of one.)

The figure below shows an example of creating a datum point by entering an edge parameter.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Enter parameter**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select the edge on which the datum point will be positioned.

   An arrow appears on the selected edge indicating the direction of increasing parameter value.

6. In the text field in the prompt area, enter the value of the edge parameter.

   The datum point appears. You can modify the position of the datum point by selecting **Feature->Edit** from the main menu bar and modifying the value of the edge parameter.

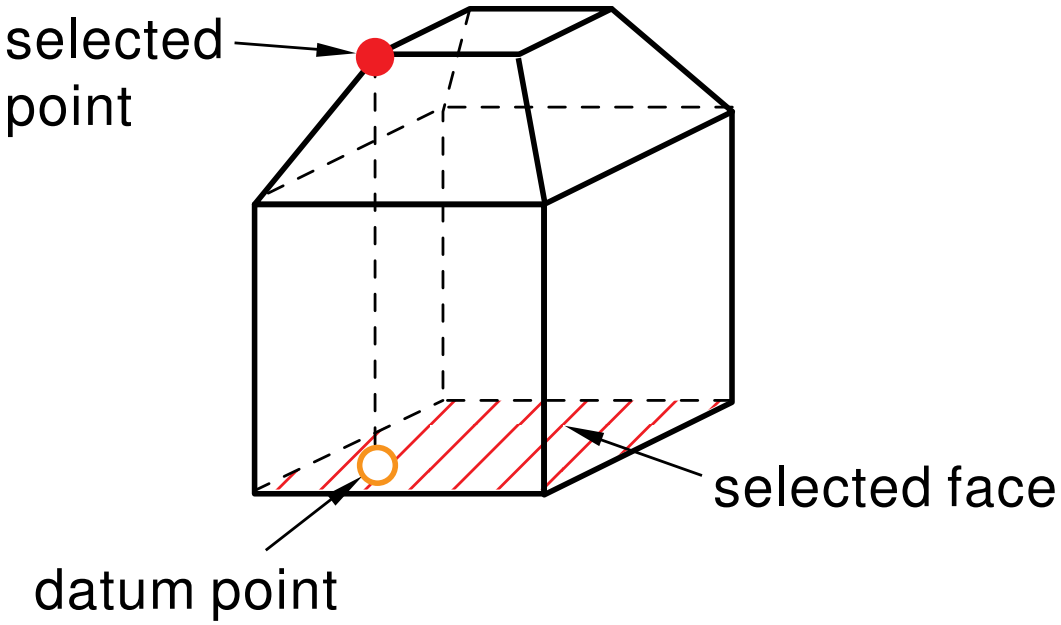
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point by projecting a point on a face or plane

You can position a datum point by selecting a point on the model and a face or plane on which to project the point. Abaqus/CAE creates the datum point where the selected face or plane intersects a line that is normal to it and passes through the selected point.

The figure below shows an example of creating a datum point by projecting a point on a face.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Project point on face/plane**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a point.
6. Select a face or plane on which to project the point. The selection can be a planar face, a nonplanar face, or a datum plane.

The datum point appears. You cannot modify a datum point created with this method; you must delete the old point and create a new one.

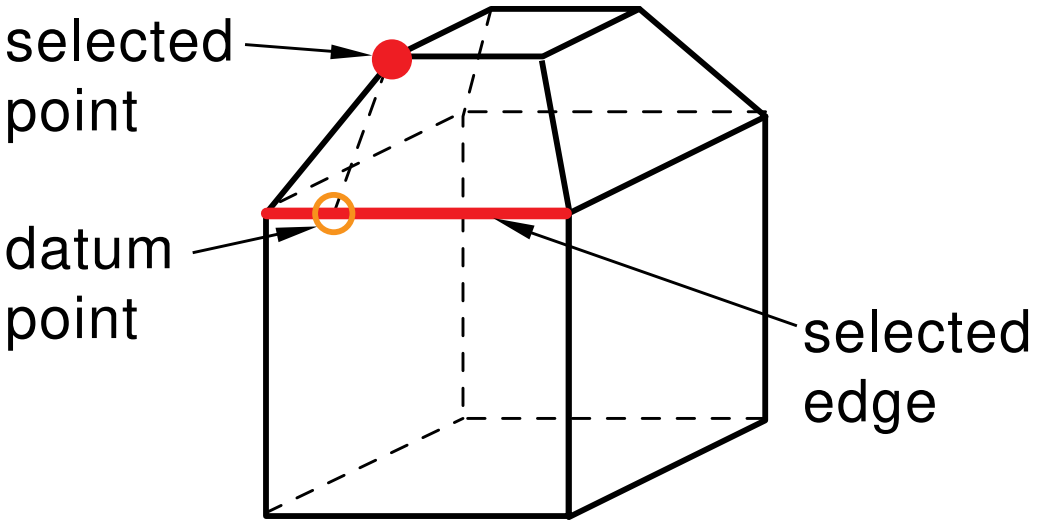
Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum point by projecting a point on an edge or datum axis

You can position a datum point by selecting a point on the model and an edge or datum axis on which to project the point.
Abaqus/CAE creates the datum point where the selected edge or datum axis intersects a line that is normal to it and passes through the selected point.

The figure below shows an example of creating a datum point by projecting a point on an edge.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum point using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Point**.

   The **Method** list indicates the methods you can use to create a datum point.

3. From the **Method** list, select **Project point on edge/datum axis**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a point.
6. Select an edge or datum axis on which to project the point.

   The datum point appears. You cannot modify a datum point created with this method; you must delete the old point and create a new one.

Additional information

- *Methods for creating a datum point*
- *Datum creation techniques*
- *Controlling datum display*

Creating datum axes

This section describes the tools that you can use to create a datum axis.

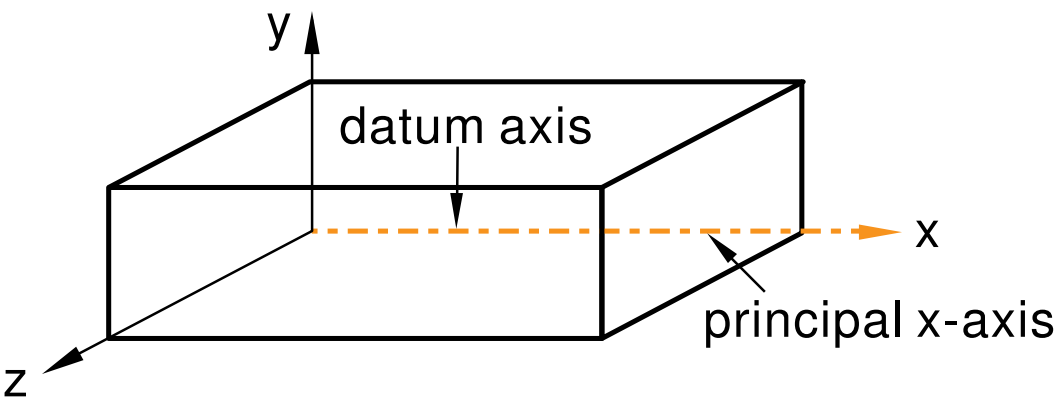
In this section:

- *Creating a datum axis along a principal axis*
- *Creating a datum axis along the intersection of two planes*
- *Creating a datum axis along a straight edge*
- *Creating a datum axis through two points*
- *Creating a datum axis along the axis of a cylinder*
- *Creating a datum axis normal to a plane and passing through a point*
- *Creating a datum axis parallel to a line and passing through a point*
- *Creating a datum axis running along the axis of a circle defined by three points*
- *Creating a datum axis by rotating an existing edge through a specified angle*

Creating a datum axis along a principal axis

You can position a datum axis by selecting one of the three principal axis to define as the datum axis.

The figure below shows an example of creating a datum axis by selecting a principal axis.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Principal axis**.
4. Select the principal *X*-, *Y*-, or *Z*-axis through which your datum axis must pass.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

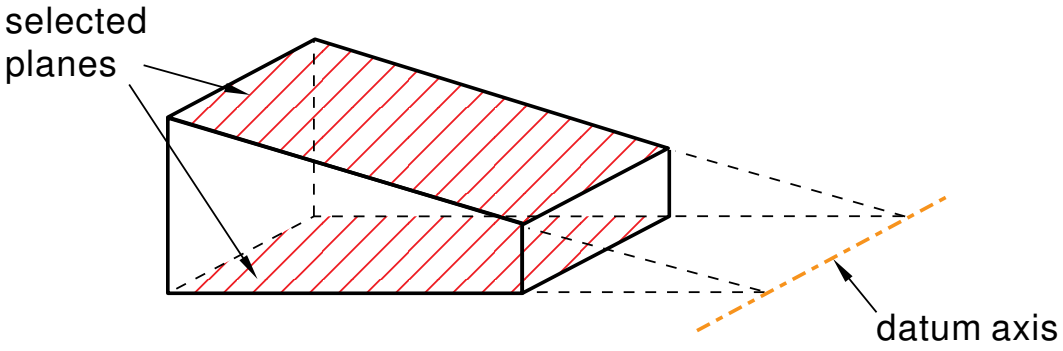
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis along the intersection of two planes

You can position a datum axis by selecting two nonparallel planar surfaces.
Abaqus/CAE creates the datum axis where the two planes, or a projection of the two planes, intersect.

The figure below shows an example of creating a datum axis at the intersection of two planes.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Intersection of 2 planes**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a plane.
6. Select a second plane that is not parallel with the first.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

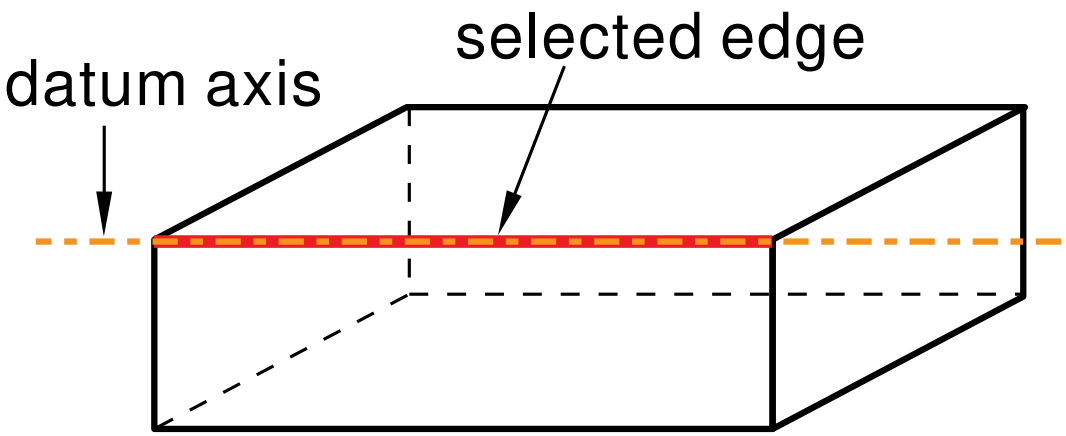
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis along a straight edge

You can position a datum axis by selecting a straight edge on the model to define as the datum axis.

The figure below shows an example of creating a datum axis along a straight edge on the model.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the  tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Straight edge**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a straight edge.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

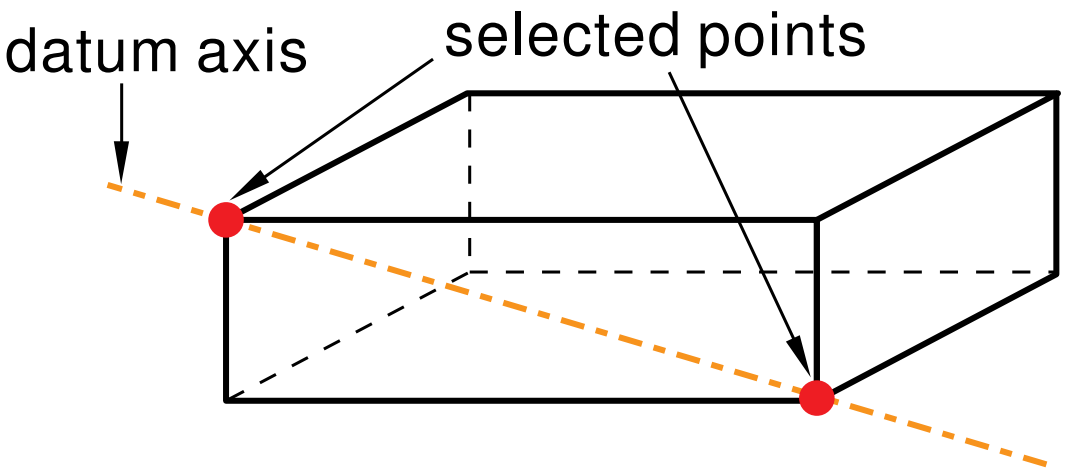
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis through two points

You can position a datum axis by selecting any two points on the model through which the datum axis passes.

The figure below shows an example of creating a datum axis through two points.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **2 points**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select two points.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

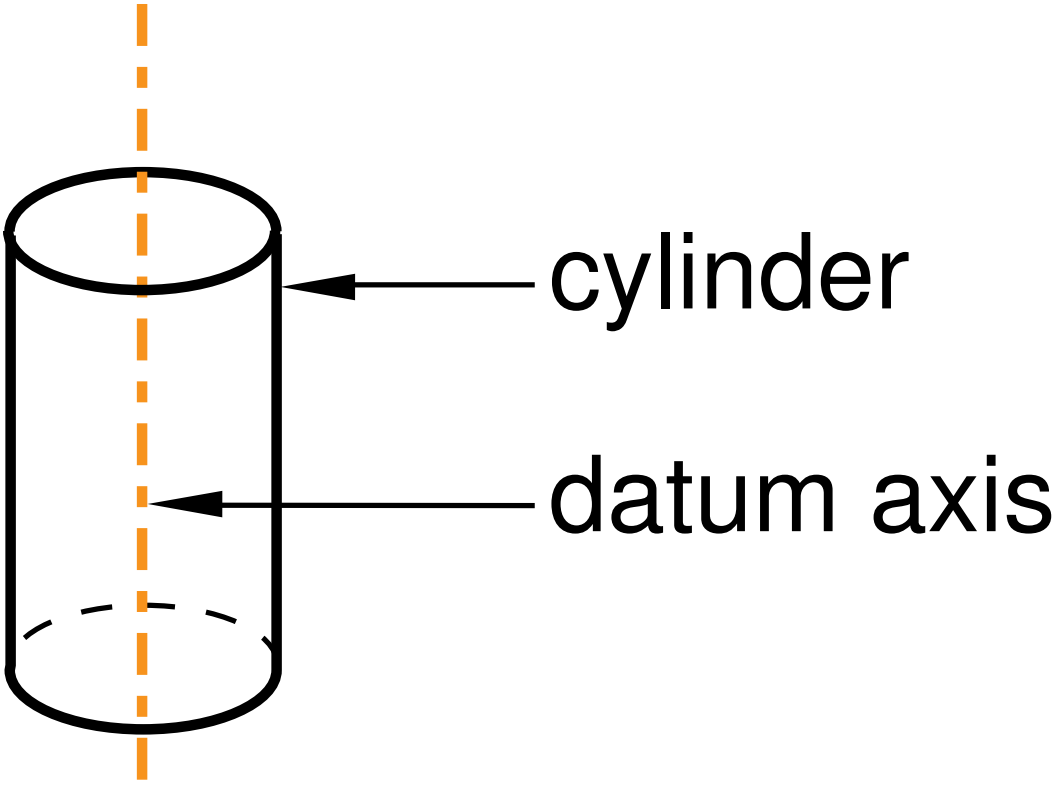
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis along the axis of a cylinder

You can position a datum axis by selecting a cylindrical or conical face on the model. Abaqus/CAE creates a datum axis that lies along the axis of the face.

The figure below shows an example of creating a datum axis along the axis of a cylinder.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Axis of cylinder**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a a cylindrical or conical face.

The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

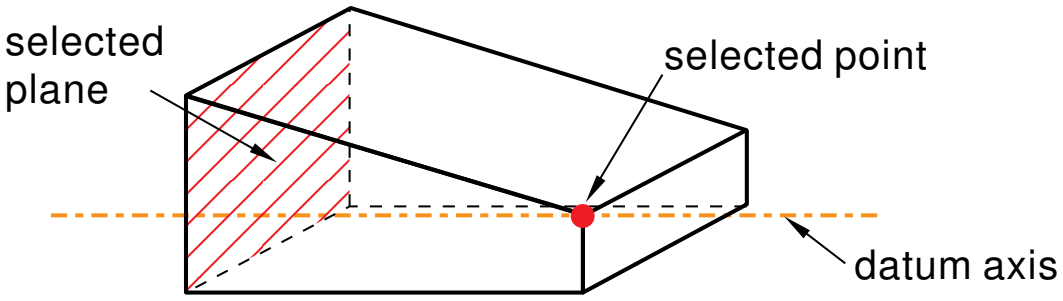
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis normal to a plane and passing through a point

You can position a datum axis by selecting a plane and a point that is not on the plane.
Abaqus/CAE creates a datum axis that is normal to the plane and passes through the point.

The figure below shows an example of creating a datum axis normal to a plane and passing through a point.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Normal to plane, thru point**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a planar face.
6. From the part or assembly in the current viewport, select a point.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

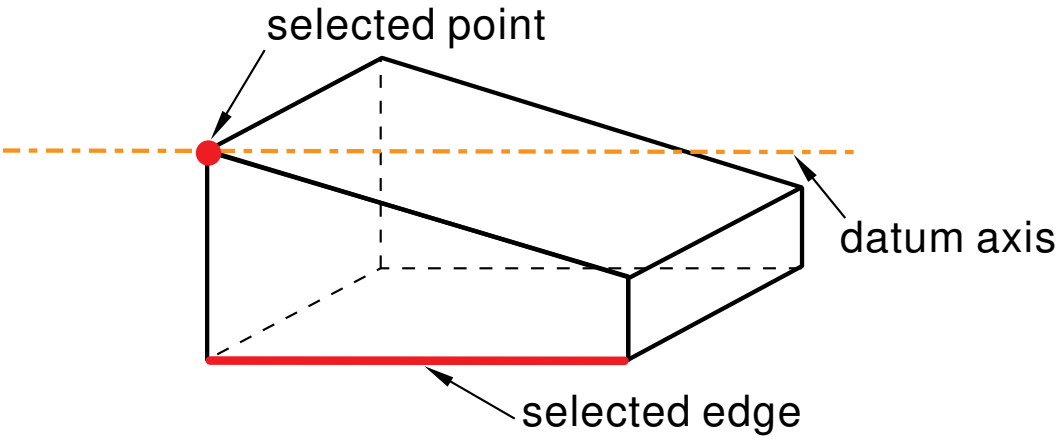
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis parallel to a line and passing through a point

You can position a datum axis by selecting an edge of the model and a point outside the edge. Abaqus/CAE creates a datum axis that is parallel to the edge and passes through the point.

The figure below shows an example of creating a datum axis parallel to a line and passing through a point.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **parallel to line, thru point**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select an edge.
6. From the part or assembly in the current viewport, select a point.

   The datum axis appears. You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

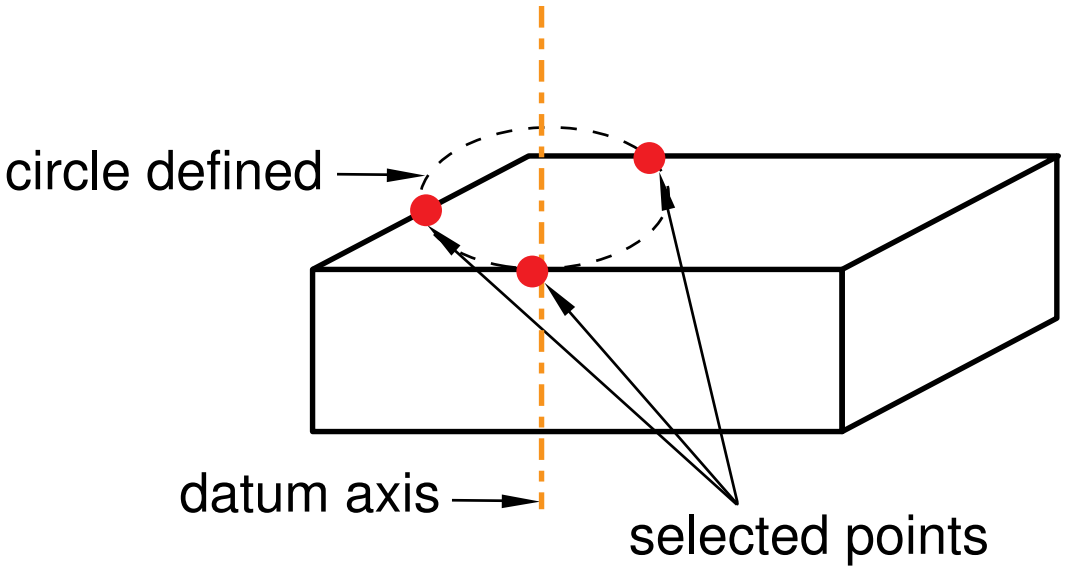
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis running along the axis of a circle defined by three points

You can position a datum axis by selecting three points on the model that define a circle. Abaqus/CAE creates a datum axis along the axis of the circle.

The figure below shows an example of creating a datum axis running along the axis of a circle defined by three points.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Three points on circle**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select three points.

   The datum axis appears.You cannot modify a datum axis created with this method; you must delete the old axis and create a new one.

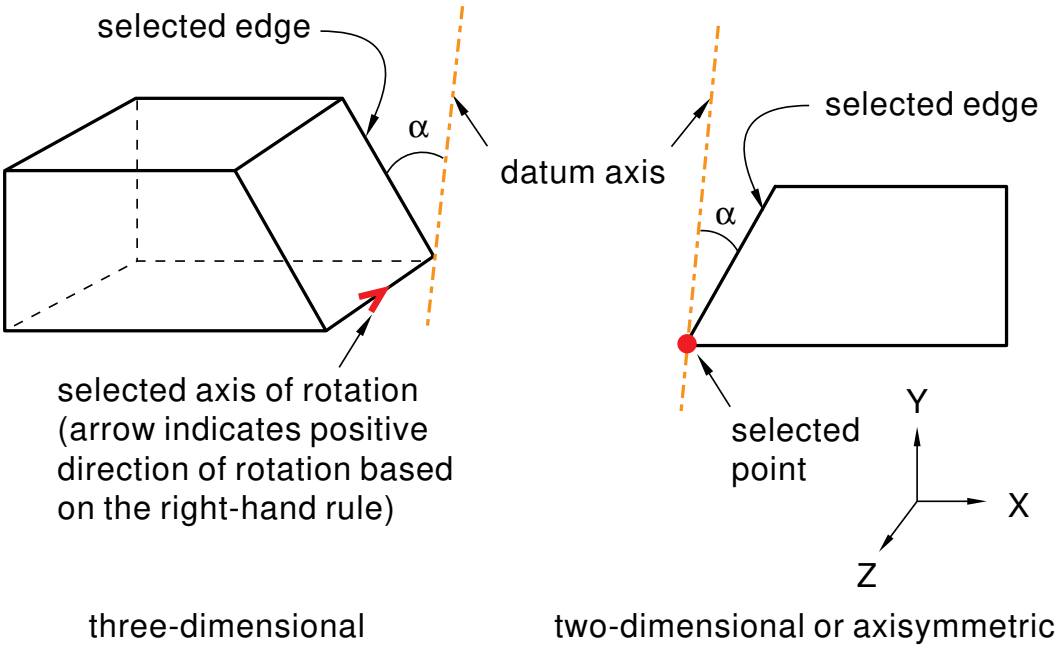
Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum axis by rotating an existing edge through a specified angle

You can define a datum axis by rotating a selected edge or datum axis about a selected axis of rotation and through a specified angle.

The figures below show examples of creating a datum axis by rotating an existing edge through a specified angle.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum axis using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Axis**.

   The **Method** list indicates the methods you can use to create a datum axis.

3. From the **Method** list, select **Rotate from line**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select an edge or a datum axis to rotate.
6. The technique for specifying the rotation depends on the modeling space of the part or assembly:

For a two-dimensional or axisymmetric part or assembly:

1. Select a vertex or datum point at the center of rotation. You can also use the text box that appears in the prompt area to enter the precise coordinates of the center of rotation.

2. In the text box in the prompt area, enter the angle of rotation. A positive angle indicates a counterclockwise rotation about the *Z*–axis.

   The datum axis appears.

For a three-dimensional part or assembly:

1. Select an edge or a datum axis to represent the axis of rotation.

   Abaqus/CAE displays an arrow along the selected edge indicating the positive direction of rotation using the right-hand rule.
2. In the text box that appears in the prompt area, type the angle of rotation.

   The datum axis appears.

You can modify the angle of rotation by selecting **Feature->Edit** from the main menu bar and selecting the datum axis. If you change the sign of the angle, Abaqus/CAE reverses the direction of rotation.

Additional information

- *Methods for creating a datum axis*
- *Datum creation techniques*
- *Controlling datum display*

Creating datum planes

This section describes the tools that you can use to create a datum plane.

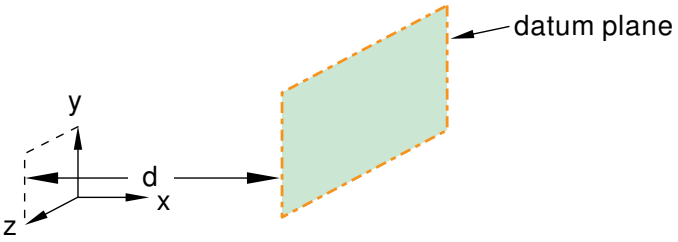
In this section:

- *Creating a datum plane offset from a principal plane*
- *Creating a datum plane at an offset from a selected plane*
- *Creating a datum plane passing through three points*
- *Creating a datum plane through a line and a point*
- *Creating a datum plane passing through a point and normal to an edge*
- *Create a datum plane midway between two points and normal to the line connecting the two points*
- *Creating a datum plane by rotating an existing face through a specified angle*

Creating a datum plane offset from a principal plane

You can position a datum plane by selecting any plane on the model (including another datum plane) and specifying an offset from the selected plane; you can specify the offset by entering a value or selecting a point.

The figure below shows an example of creating a datum plane offset from a principal plane.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Offset from principal plane**.
4. From the buttons in the prompt area, select the principal plane.
5. In the text field in the prompt area, enter the offset from the selected principal plane. The offset can be positive or negative. A positive value offsets the datum plane along the positive principal axis normal to the selected plane.

   The datum plane appears. You can modify the position of the datum plane by selecting **Feature->Edit** from the main menu bar and modifying the value of the offset.

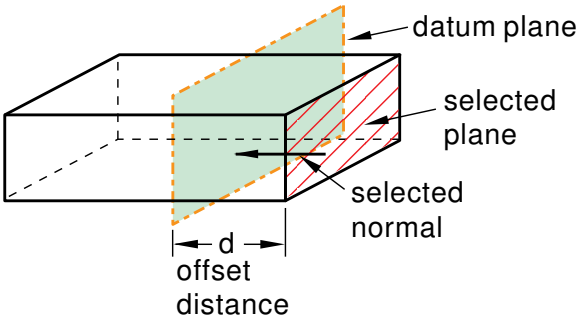
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum plane at an offset from a selected plane

You can position a datum plane by selecting any plane on the model (including another datum plane) and specifying an offset from the selected plane; you can specify the offset by entering a value or selecting a point.

The figure below shows an example of creating a datum plane at an offset from another plane.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Offset from plane**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a plane.
6. From the buttons in the prompt area, select one of the following:

Enter Value

1. An arrow appears, indicating the offset direction.
2. Click **Flip** to reverse the arrow, if necessary. Click **OK** to accept the indicated offset direction.
3. In the text field in the prompt area, enter the offset from the selected plane. The offset can be positive or negative. A positive value offsets the datum plane in the direction indicated by the arrow.

Select Point

From the part or assembly in the current viewport, select a point to define the offset from the selected plane.

The datum plane appears. You can modify the position of the datum plane by selecting **Feature->Edit** from the main menu bar and modifying the value of the offset.

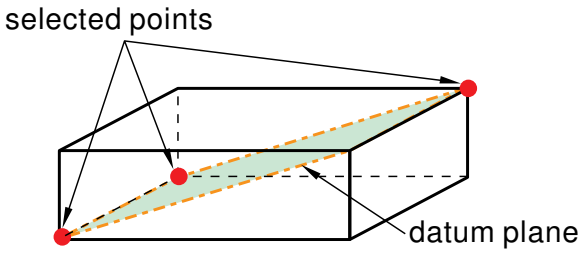
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum plane passing through three points

You can position a datum plane by selecting three points through which the datum plane must pass.

The figure below shows an example of creating a datum plane passing through three points.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **3 points**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select three points.

   The datum plane appears. You cannot modify a datum plane created with this method; you must delete the old plane and create a new one.

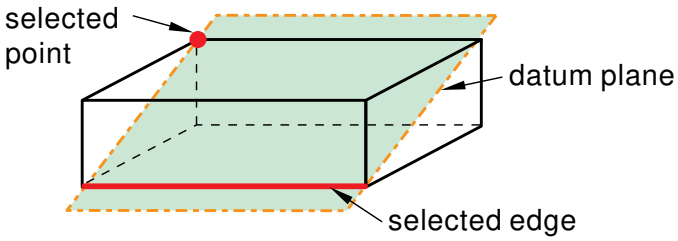
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum plane through a line and a point

You can position a datum plane by selecting an edge and a point through which the datum plane must pass.

The figure below shows an example of creating a datum plane through a line and a point.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Line and point**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a straight edge.
6. From the part or assembly in the current viewport, select a point.

   The datum plane appears. You cannot modify a datum plane created with this method; you must delete the old plane and create a new one.

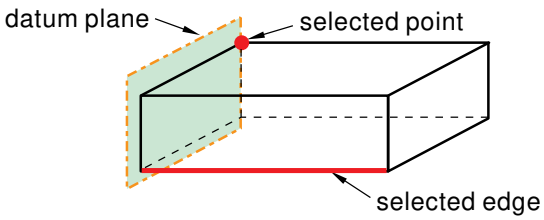
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum plane passing through a point and normal to an edge

You can position a datum plane by selecting a point and an edge; the datum plane passes through the point and is normal to the selected edge.

The figure below shows an example of creating a datum plane passing through a point and normal to an edge.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Point and normal**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a point.
6. From the part or assembly in the current viewport, select an edge.

   The datum plane appears. You cannot modify a datum plane created with this method; you must delete the old plane and create a new one.

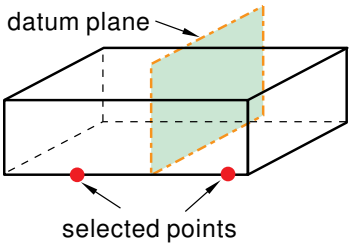
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Create a datum plane midway between two points and normal to the line connecting the two points

You can position a datum plane by selecting two points.
Abaqus/CAE creates a datum plane midway between the two selected points and normal to the line connecting them.

The figure below shows an example of creating a datum plane midway between two points and normal to the line connecting the two points.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Midway between 2 points**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select two points.

   The datum plane appears. You cannot modify a datum plane created with this method; you must delete the old plane and create a new one.

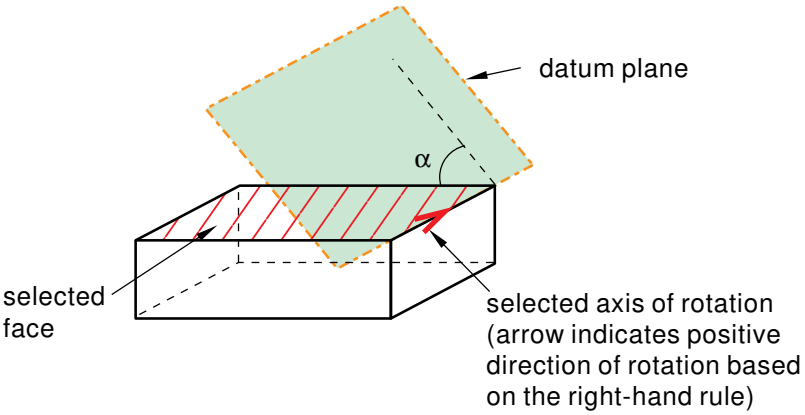
Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum plane by rotating an existing face through a specified angle

You can define a datum plane by rotating a selected face or datum plane about a selected axis of rotation and through a specified angle.

The figure below shows an example of creating a datum plane by rotating an existing face through a specified angle.



1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum plane using the tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **Plane**.

   The **Method** list indicates the methods you can use to create a datum plane.

3. From the **Method** list, select **Rotate from plane**.
4. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
5. From the part or assembly in the current viewport, select a face or a datum plane to rotate.
6. Select any edge or a datum axis to represent the axis of rotation. The edge or datum axis you select for the axis of rotation does not have to be coplanar with the face or datum plane you selected to rotate.

   Abaqus/CAE displays an arrow along the selected edge indicating the positive direction of rotation using the right-hand rule.

7. In the text box that appears in the prompt area, type the angle of rotation.

   The datum plane appears. You can modify the angle of rotation by selecting **Feature->Edit** from the main menu bar and selecting the datum plane. If you change the sign of the angle, Abaqus/CAE reverses the direction of rotation.

Additional information

- *Methods for creating a datum plane*
- *Datum creation techniques*

- *Controlling datum display*

Creating datum coordinate systems

This section describes the tools that you can use to create a datum coordinate system.

In this section:

- *Creating a datum coordinate system defined by three points*
- *Creating a datum coordinate system at an offset from another coordinate system*
- *Creating a datum coordinate system defined by two lines*

Creating a datum coordinate system defined by three points

You can position a rectangular, cylindrical, or spherical datum coordinate system by selecting a point defining the origin, a point on the *X*- or *R*-axis, or a point in the *X*–*Y* or *R*–θ plane.

The figure below shows an example of creating a rectangular datum coordinate system defined by three points.



When Abaqus/CAE prompts you to select a point, you can select the origin of a datum coordinate system. When Abaqus/CAE prompts you to select an edge, you can select one of the axes of a datum coordinate system.

1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum coordinate system using the  tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **CSYS**.

   The **Method** list indicates the methods you can use to create a datum coordinate system.

3. From the **Method** list, select **3 points**.

   Abaqus/CAE displays the **Create Datum CSYS** dialog box.

4. From the dialog box, enter the name of the datum coordinate system.

   To help keep track of your datum coordinate systems, Abaqus/CAE displays its name in the Model Tree. In addition, you can use the Model Tree to rename the datum coordinate system.

5. From the dialog box, select one of the following datum coordinate systems:

   - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Cylindrical**: The *R*-, θ-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Spherical**: The *R*-, θ-, and ϕ-axes are aligned with the 1-, 2-, and 3-global axes, respectively.

6. From the **Create Datum CSYS** dialog box, click **Continue**.

   Abaqus/CAE displays a default datum coordinate system. The origin of the datum coordinate system is located at the origin of the part or assembly. In addition, the axes of the default datum coordinate system are aligned with the coordinate system of the part or assembly.

7. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.

8. To create a datum coordinate system at a different location or with a different orientation, do the following:

   a. From the part or assembly in the current viewport, select a point that will define the origin of the datum coordinate system or enter the coordinates of the origin.

      Abaqus/CAE displays a temporary datum coordinate system at the selected point. The axes of the datum coordinate system are aligned with the axes of the global coordinate system.

b. From the prompt area, press **[Enter]** in the text field to accept the default orientation of the temporary datum coordinate system or select (or enter the coordinates of) a point that will lie on the *X*- or *R*-axis.

Abaqus/CAE recalculates the orientation of the temporary datum coordinate system.

c. From the prompt area, press **[Enter]** in the text field to accept the new orientation of the temporary datum coordinate system or select (or enter the coordinates of) a point that will lie on the *X*–*Y* or *R*–θ plane.

Abaqus/CAE creates the datum coordinate system. You cannot move or rotate a datum coordinate system.

Additional information

- *Methods for creating a datum coordinate system*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum coordinate system at an offset from another coordinate system

You can position a rectangular, cylindrical, or spherical datum coordinate system by selecting a coordinate system and specifying an offset; you can specify the offset by entering a value or selecting a point.

The figure below shows an example of creating a rectangular datum coordinate system at an offset from the default coordinate system.



When Abaqus/CAE prompts you to select a point, you can select the origin of a datum coordinate system. When Abaqus/CAE prompts you to select an edge, you can select one of the axes of a datum coordinate system.

1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum coordinate system using the  tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **CSYS**.

   The **Method** list indicates the methods you can use to create a datum coordinate system.

3. From the **Method** list, select **Offset from CSYS**.

   Abaqus/CAE displays the **Create Datum CSYS** dialog box.

4. From the dialog box, enter the name of the datum coordinate system.

   To help keep track of your datum coordinate systems, Abaqus/CAE displays its name in the Model Tree. In addition, you can use the Model Tree to rename the datum coordinate system.

5. From the dialog box, select one of the following datum coordinate systems:

   - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Cylindrical**: The *R*-, θ-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Spherical**: The *R*-, θ-, and ϕ-axes are aligned with the 1-, 2-, and 3-global axes, respectively.

6. From the **Create Datum CSYS** dialog box, click **Continue**.
7. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
8. From the current viewport, select a datum coordinate system.
9. From the buttons in the prompt area, select one of the following:

Enter Value

In the text field in the prompt area, enter the *X*-, *Y*-, and *Z*-coordinates of the offset from the selected default coordinate system.

Select Point

From the part or assembly in the current viewport, select a point to define the offset from the selected default coordinate system.

The datum coordinate system appears. You cannot move or rotate a datum coordinate system.

Additional information

- *Methods for creating a datum coordinate system*
- *Datum creation techniques*
- *Controlling datum display*

Creating a datum coordinate system defined by two lines

You can position a rectangular, cylindrical, or spherical datum coordinate system by selecting two edges that define it.

The first edge defines the *X*- or *R*-axis, and the *X*–*Y* or *R*-θ-plane passes through the second.

The figure below shows an example of creating a rectangular datum coordinate system defined by two lines.



When Abaqus/CAE prompts you to select an edge, you can select one of the axes of a datum coordinate system.

1. From the main menu bar, select **Tools->Datum**.

   The **Create Datum** dialog box appears. The dialog box outlines the types of datum geometry you can create.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also create a datum coordinate system using the  tool, located in the module toolbox. For a diagram of the datum tools in the toolbox, see *Using the Datum toolset*.

2. From the list of types at the top of the dialog box, choose **CSYS**.

   The **Method** list indicates the methods you can use to create a datum coordinate system.

3. From the **Method** list, select **2 lines**.

   Abaqus/CAE displays the **Create Datum CSYS** dialog box.

4. From the dialog box, enter the name of the datum coordinate system.

   To help keep track of your datum coordinate systems, Abaqus/CAE displays its name in the Model Tree. In addition, you can use the Model Tree to rename the datum coordinate system.

5. From the dialog box, select one of the following datum coordinate systems:

   - **Rectangular**: The *X*-, *Y*-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Cylindrical**: The *R*-, θ-, and *Z*-axes are aligned with the 1-, 2-, and 3-global axes, respectively.
   - **Spherical**: The *R*-, θ-, and ϕ-axes are aligned with the 1-, 2-, and 3-global axes, respectively.

6. From the **Create Datum CSYS** dialog box, click **Continue**.
7. If desired, select **Make Independent** from the prompt area to create the datum as an independent feature.
8. From the part or assembly in the current viewport, select a straight edge that will define the *X*- or *R*-axis of the datum coordinate system.
9. Select a straight edge that will lie in the *X*–*Y* or *R*–θ-plane.

   The datum coordinate system appears. You cannot move or rotate a datum coordinate system.

Additional information

- *Methods for creating a datum coordinate system*
- *Datum creation techniques*
- *Controlling datum display*

The Discrete Field toolset

A discrete field is a spatially varying field where values are associated with a node or an element. The Discrete Field toolset allows you to create and manage discrete fields.

In this section:

- *Using the Discrete Field toolset*
- *Creating discrete fields*
- *Evaluating discrete fields*
- *Creating discrete fields for material volume fractions*

Using the Discrete Field toolset

The Discrete Field toolset allows you to define spatially varying parameters. The parameters in a discrete field can be associated with specified elements or nodes. For example, you can define a spatially varying thickness or orientation (Cartesian, cylindrical, or spherical). The Discrete Field toolset is available in the Property module, Interaction module, and Load module.

Select **Tools**->**Discrete Field**->**Create** from the main menu bar to create a new discrete field; select **Edit** from the same menu to change an existing discrete field. Either command opens the discrete field editor, which allows you to associate data with elements or nodes.

Creating discrete fields

The Discrete Field toolset is available in the Property module, Interaction module, and Load module. Select **Tools->Discrete Field->Create** from the main menu bar to create a discrete field.

1. From the main menu bar, select **Tools->Discrete Field->Create**.

   Tip: You can also click **Create** in the **Discrete Field Manager**.

2. In the **Name** text field, enter a name for the discrete field. For information on naming objects, see *Using basic dialog box components*.
3. In the **Description** text field, enter a description for the discrete field.
4. Choose **Elements** or **Nodes** to specify the type of location to which the field will be applied.
5. Choose the data type.
   - Choose **Scalar** to enter scalar field data.
   - The **Orientation** option is available for fields associated with elements. Choose **Orientation**, and do the following:
     1. Choose the coordinate system type—**Cartesian**, **Cylindrical**, or **Spherical**.
     2. Toggle on **Supplied orientation directions are defined in part space** to specify that the orientation directions are defined at the part or part instance level. When you define the orientations in the part space, Abaqus transforms the orientation values from the part's coordinate system into the assembly coordinate system using the part instance transformation. Toggle off this option to specify that the orientation directions are defined at the assembly level.
   - Choose **Prescribed condition** to include degree of freedom values in the field data.
6. Enter the default component value or values. For information on how Abaqus uses the default values, see *Evaluating discrete fields*. Scalar discrete fields require one default component value. Orientation discrete fields require six default component values. Prescribed condition discrete fields require six default component values and up to six degree of freedom values.
7. In the **Field Data** table, enter the following values:
   - Element or node number and its associated component value or values.

     When creating discrete fields that will be used with assembly-level or history objects, such as loads or interactions, you must specify the complete name of the node or element numbers, as described in *Naming Conventions*. For example, you would specify the part instance name and element id using the notation *Part_instance_name.EID*, such as `PART-2-1.20`.
   - For prescribed condition discrete fields, enter the degree of freedom values.

   Click mouse button 3 on the table to see a menu that allows you to do the following:
   - Cut, copy, and paste values
   - Insert and delete rows
   - Clear a table cell or the entire table
   - Read from or write to a comma-separated file

   For more information, see *Entering tabular data*.

8. Click **OK** to create the discrete field and to exit the editor.

Additional information

- *Defining sections*
- *Composite layups*

Evaluating discrete fields

If you specify a discrete field for a prescribed condition, Abaqus/CAE evaluates the discrete field when the input file is written.

- If an element or node is found in both the region selected for the prescribed condition and the discrete field, Abaqus/CAE multiplies the magnitude of the prescribed condition by the value at each element or node to determine the final values that are submitted to the analysis. A material assignment predefined field is the only exception; this prescribed condition has no magnitude.
- If an element or node is found in the region selected for the prescribed condition but not in the discrete field, the magnitude of the prescribed condition is multiplied by the default value and submitted to the analysis.

Table 1 summarizes how the value written to the input file is determined for scalar discrete fields.

Table 1: Evaluating scalar discrete fields.

Element or node in region selected for prescribed condition?	Element or node in discrete field?	Value written to input file
Yes	Yes	Magnitude of prescribed condition × magnitude specified in discrete field
Yes	No	Magnitude of prescribed condition × default value specified in discrete field
No	Yes	Not written to input file

For prescribed conditions that have multiple degrees of freedom, such as displacements, prescribed condition discrete fields are used; and evaluating the discrete field becomes more complex. For prescribed conditions where you can activate the individual degrees of freedom, only the active degrees of freedom in the prescribed condition are considered; any degree of freedom that is not active in the prescribed condition will be ignored. The magnitudes specified for each degree of freedom in the prescribed condition are multiplied by either the magnitude specified in the discrete field or the default value for that degree of freedom in the discrete field.

For example, if you define a displacement boundary condition, as shown in *Figure 1*, for `Node-set-1` (containing nodes `Part-2-1.10`, `Part-2-1.20`, and `Part-2-1.30`) and a prescribed condition discrete field, as shown in *Figure 2*, the values that are submitted to the input file are shown in *Table 2*.

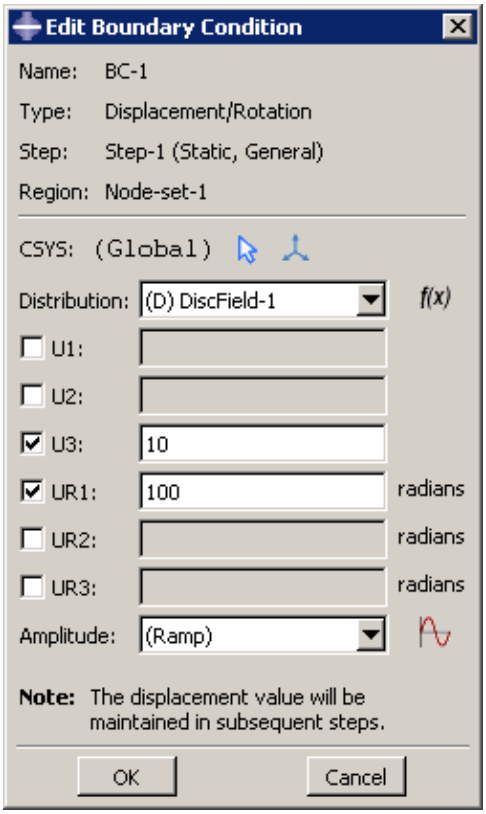


Figure 1: Displacement boundary condition with two degrees of freedom activated.

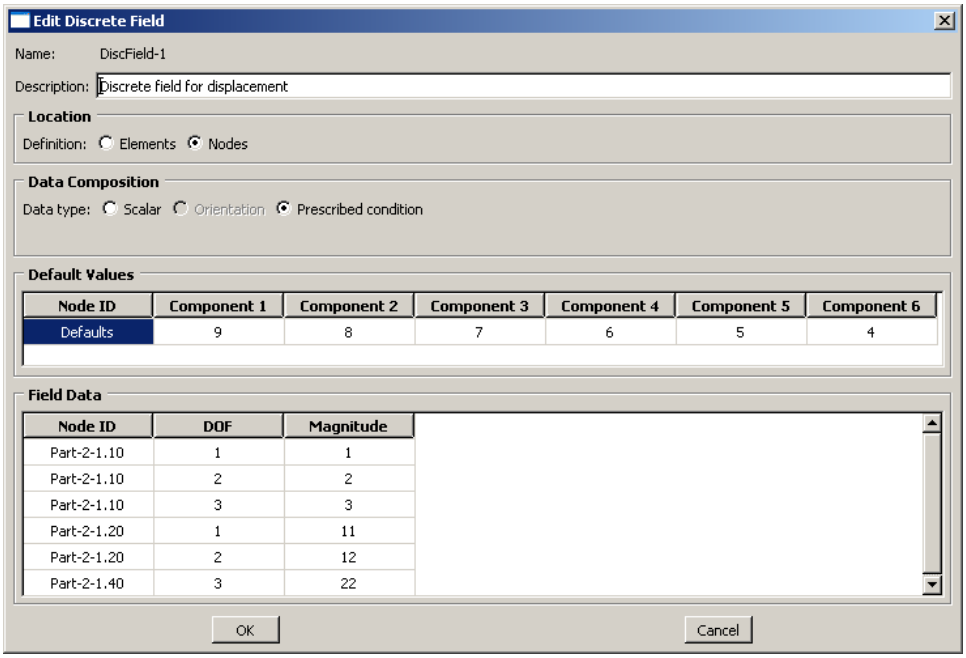


Figure 2: Prescribed condition discrete field used to define a spatially varying displacement boundary condition.

Table 2: Results for evaluating a prescribed condition discrete field.

Node ID	Degree of freedom	Value written to input file
`Part-2-1.10`	3 (**U3**)	30
`Part-2-1.10`	4 (**UR1**)	600
`Part-2-1.20`	3 (**U3**)	70
`Part-2-1.20`	4 (**UR1**)	600
`Part-2-1.30`	3 (**U3**)	70
`Part-2-1.30`	4 (**UR1**)	600
`Part-2-1.40`	3 (**U3**)	None

Degrees of freedom for boundary conditions and predefined fields that result in a zero magnitude are written to the input file as such; however, degrees of freedom for loads that result in zero magnitudes are not written to the input file.

When creating discrete fields that will be used with assembly-level and history objects, such as loads or interactions, you must specify the complete name of the node or element numbers, as described in *Naming Conventions*.

Creating discrete fields for material volume fractions

The volume fraction tool is used to create discrete fields representing the overlap between two part instances on an element-by-element basis. One of the part instances must be a meshed Eulerian part. The other part instance acts as reference geometry. The discrete field that is created associates each element in the Eulerian part with a volume fraction based on the amount of space that the reference part instance occupies within that element. This discrete field can be used to assign materials in an Eulerian model (see *Assigning materials to Eulerian part instances*). For an overview on using the volume fraction tool in Eulerian models, refer to *Using the volume fraction tool*.

The volume fraction tool is available in the Interaction module and the Load module. Select **Tools->Discrete Field->Volume Fraction Tool** from the main menu bar to create a discrete field using the volume fraction tool.

1. From the main menu bar, select **Tools->Discrete Field->Volume Fraction Tool**.
2. In the viewport, select a meshed Eulerian part instance. The discrete field that is created will be associated with the elements in this Eulerian part.

   Warning:

   You should not regenerate or edit the Eulerian part mesh after using the volume fraction tool. Any changes to the mesh—including regenerating it from replay or journal files—may invalidate the created discrete field.

3. In the viewport, select a reference part instance that intersects the previously selected Eulerian part instance. This part instance must be a three-dimensional solid or a three-dimensional shell. The reference part instance cannot contain free edges or interior dividing faces (free or non-manifold edges).

   Note:

   If the reference instance consists of meshed geometry, Abaqus/CAE always uses the meshed representation of the part instance to calculate volume fractions. If the reference instance is partially meshed, only the meshed portion of the part is considered in the volume fraction calculations.

   The **Volume Fraction Tool** dialog box appears.

4. In the **Name** text field, enter a name for the discrete field. For information on naming objects, see *Using basic dialog box components*.
5. In the **Description** text field, enter a description for the discrete field.
6. To change the Eulerian instance or reference instance, click **Edit**, and select a new part instance in the viewport.
7. Specify the **Accuracy** (**Low**, **Medium**, or **High**) for the volume fraction calculations. A lower accuracy results in faster performance for the tool.
8. Specify the **Material location**:
   - Selecting **Inside reference instance** implies that the material will be assigned to the region that is occupied or enclosed by the reference instance. A nonzero value is assigned to all Eulerian elements within this region.
   - Selecting **Outside reference instance** implies that the material will be assigned to the region that is not occupied or enclosed by the reference instance. A nonzero value is assigned to all elements that are not fully occupied or enclosed by the reference instance.
9. Specify a **Scale factor** between zero and one. All values in the discrete field are multiplied by this scale factor. By default, the scale factor is one.

10. To create a set of all nodes connected to elements with nonzero volume fractions in the discrete field, toggle on **Node set**, and enter a name for the set.
11. To create a set of all elements connected to elements with nonzero volume fractions in the discrete field, toggle on **Element set**, and enter a name for the set.
12. Click **OK** to create the discrete field and to close the **Volume Fraction Tool** dialog box.

Additional information

- *Assigning materials to Eulerian part instances*
- *Using the volume fraction tool*
- *Using the Discrete Field toolset*

The Edit Mesh toolset

You can use the Edit Mesh toolset in the Mesh module to improve the mesh quality. You can modify an orphan mesh, or you can modify an Abaqus native mesh.

In this section:

- *What can I do with the Edit Mesh toolset?*
- *What is the difference between editing an orphan mesh, a meshed part, and a meshed part instance in the assembly?*
- *Meshing strategies and mesh editing techniques*
- *Using the mesh editing tools*
- *Editing nodes*
- *Editing elements*
- *Editing the entire mesh*
- *Editing and refining an orphan mesh*
- *Undoing or redoing a change in the Edit Mesh toolset*

What can I do with the Edit Mesh toolset?

This section describes how you can use the Edit Mesh toolset to modify a mesh in the Mesh module. All of the tools are available when you are modifying a mesh that contains orphan elements and nodes; however, only a few of the tools are available for modifying mesh nodes or elements that are tied to geometry, including meshed part instances in the assembly.

In this section:

- *Manipulating nodes*
- *Manipulating elements*
- *Manipulating the mesh*
- *Refining the mesh*

Manipulating nodes

The Edit Mesh toolset provides the following tools that allow you to manipulate the nodes in your mesh:

- Create a node. You can specify the coordinates of the new node either in the global coordinate system or in a datum coordinate system that you specify.
- Edit nodes. You can specify the new coordinates of the nodes either in the global coordinate system or in a datum coordinate system that you specify. Alternatively, you can specify an offset from the current position. You can edit a single node, or you can edit multiple nodes simultaneously.
- Drag nodes. You can click and drag the nodes in a model. You can drag only one node at a time, and you can use the **Project to geometry** option to project an exterior node to the geometry with which it is associated. Element quality checks are automatically activated to show element errors and warnings while you drag a node. For more information on element quality checks, see *Verifying element quality*.
- Project nodes. You can specify a vertex, edge, or face and Abaqus/CAE projects the nodes from their current positions onto the selected entity. You can project a single node, or you can project multiple nodes simultaneously. You can also project nodes to other nodes, element edges, element faces, datum points, datum axes, or datum planes.
- Delete nodes. Any elements associated with the deleted nodes are also deleted. In addition, you have the option of deleting any remaining nodes that would be left unassociated with any elements once the nodes selected for deletion and their associated elements are deleted.
- Merge selected nodes. If you select only two nodes to merge, Abaqus/CAE creates a new node at the midpoint of the selected nodes. If you select more than two nodes, you can specify the **Node merging tolerance**, which is the maximum distance between nodes that will be merged. Abaqus/CAE deletes nodes that are closer than the specified distance and replaces them with a single new node. The location of the new node is the average position of the group of nodes that were merged into the new node.

  While Abaqus/CAE is removing duplicate nodes, you can choose to remove duplicate elements that have the same connectivity. You can also merge instances of parts in the Assembly module and you have the option to merge nodes within this process; for more information, see *Merging and cutting part instances*.
- Smooth selected nodes. You can select external nodes, and Abaqus/CAE smooths the node positions with respect to the surrounding nodes. You can smooth any external nodes of the native mesh on a part or on an assembly as long as they are not part of a region boundary; you can also smooth external orphan mesh nodes that lie within a planar mesh face.
- Adjust the position of the midside node of second-order elements to allow for the singularity at the crack tip in a fracture mechanics analysis. You can select nodes and enter a bias parameter between 0 and 1. Abaqus/CAE moves the midside nodes along connected element edges to a position based on the parameter that you entered. For example, if you enter a parameter of 0.25, Abaqus/CAE biases the position of the midside nodes one quarter of the length of the element edge away from the selected nodes.

  For more information, see *Controlling the singularity at the crack tip for a small-strain analysis*, and *Constructing a Fracture Mechanics Mesh for Small-Strain Analysis with the Conventional Finite Element Method*.
- Renumber selected nodes within an orphan mesh. You can renumber the selected nodes by specifying a starting label and an increment or by offsetting the existing label by a specified value.



Tip: The Mesh Edit Undo feature can roll back any change you make to the nodes in the mesh. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

For detailed instructions about each of these node manipulation techniques, see *Editing nodes*.

Manipulating elements

The Edit Mesh toolset provides the following tools that allow you to manipulate the elements in your mesh:

- Create an element. You must specify the shape of the element that you want to create, and you must select the nodes in the order appropriate for that element shape.
- Delete elements. You have the option of deleting any nodes that would be left unassociated with any elements once the selected elements are deleted.
- Flip the surface normal direction of shell elements.
- Collapse a selected edge of quadrilateral or triangular elements. Collapsing an edge is useful for removing slivers from quadrilateral and triangular elements, as shown in *Figure 1*.

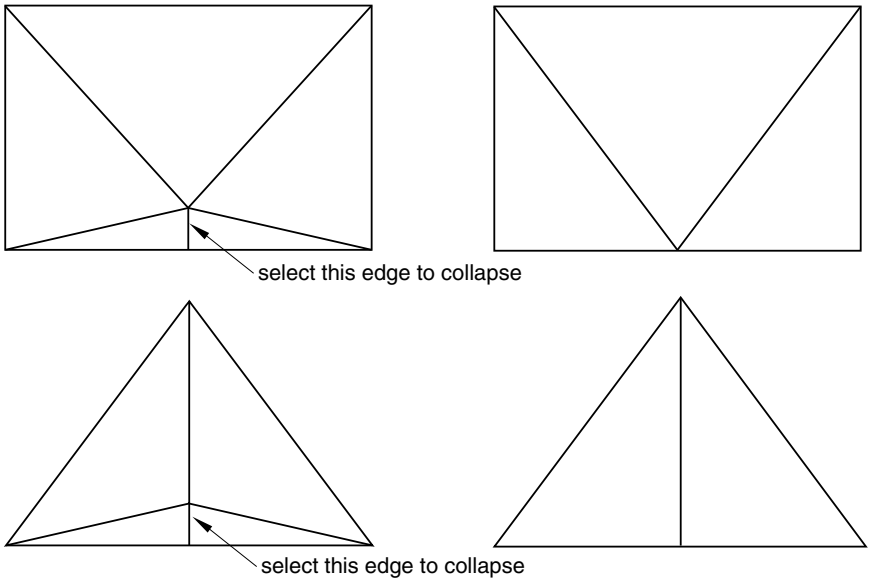


Figure 1: Collapsing an edge to remove slivers from quadrilateral and triangular elements.

When you preview a tetrahedral boundary mesh, you can collapse selected edges of the triangular elements to remove slivers. In some cases, when you create an all-quadrilateral mesh using the advancing front algorithm, Abaqus/CAE may generate quadrilateral elements with a short edge. You can collapse the short edge and create a well-formed triangular element, as shown in *Figure 2*.

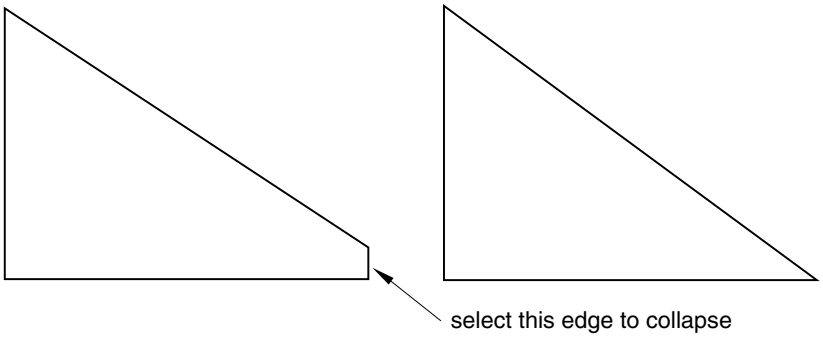


Figure 2: Collapsing an edge of a quadrilateral element.

- Split a selected edge of a quadrilateral or triangular element into two parts. You can split the edge at its midpoint, or you can click on the location of the split. You can use a combination of tools to clean up your mesh. For example,

Figure 3 illustrates how you can split an edge and then collapse the resulting edge to remove a long narrow triangular element.

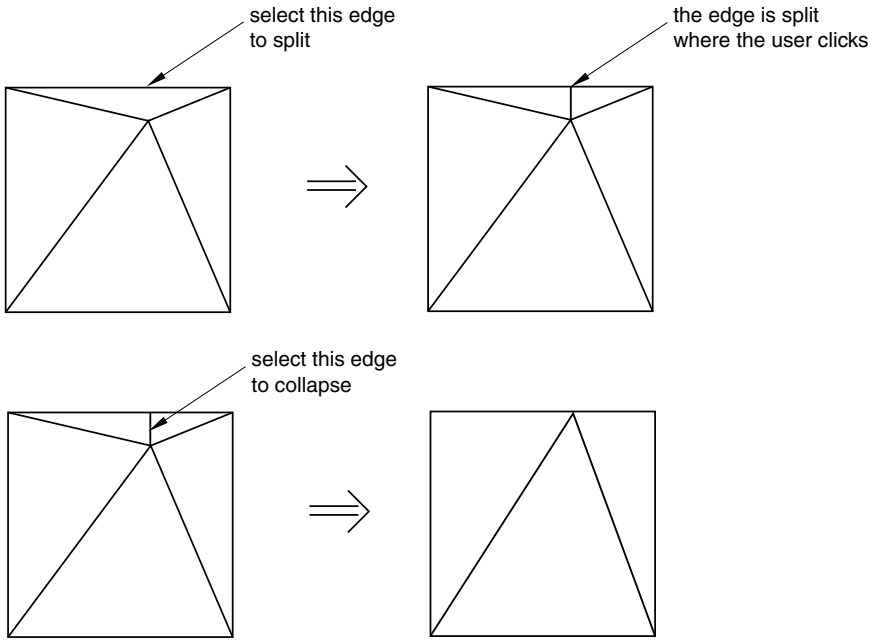


Figure 3: Splitting and collapsing edges of a triangular element.

If you are editing a native mesh in the Mesh module, Abaqus/CAE projects the new node on the geometry. As a result, you can use this tool for improving a coarse mesh around curved edges, as shown in *Figure 4*.

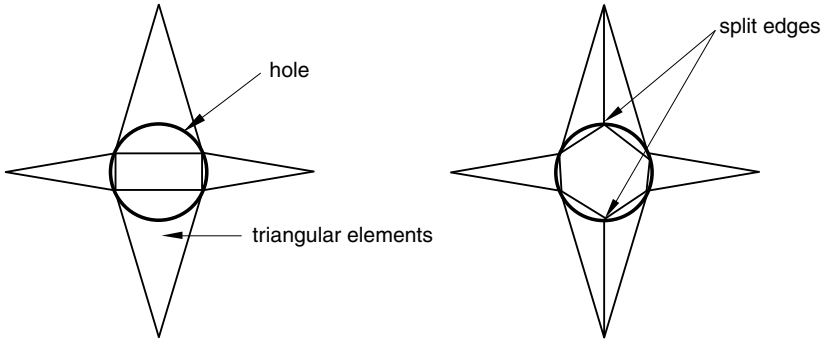


Figure 4: Improving the mesh around a hole by splitting edges.

- Swap the diagonal of a pair of adjacent triangular elements, as shown in *Figure 5*. You can swap the diagonal of either first- or second-order adjacent triangular elements if they are the same order. In addition, the adjacent triangular elements must have consistent normals. If necessary, you can flip the normals to make them consistent.

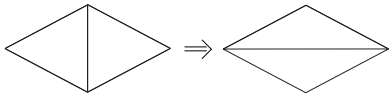


Figure 5: Swap the diagonal of a pair of adjacent triangular elements.

- Split a quadrilateral element into two triangular elements, as shown in *Figure 6*. You cannot split a 5-node quadrilateral element, and you cannot split a gasket element.

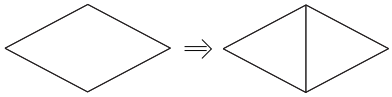


Figure 6: Split a quadrilateral element into two triangular elements.

- Combine two adjacent triangular elements into one quadrilateral element, as shown in *Figure 7*. You can combine either first- or second-order elements, but you cannot combine a combination of both. In addition, you can combine elements only if the adjacent triangular elements have consistent normals. If necessary, you can flip the normals to make them consistent.

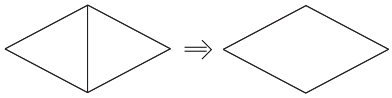


Figure 7: Combine two adjacent triangular elements into one quadrilateral element.

- Orient the stack direction of a continuum shell, cohesive, cylindrical, or gasket mesh. You can stack only hexahedral, wedge, and quadrilateral elements to form a continuum shell, cohesive, cylindrical, or gasket mesh. As a result, you can use this tool to orient the stack direction of only hexahedral, wedge, and quadrilateral elements. If you have assigned cylindrical elements or second-order gasket elements to the region, you must first change the assignment to conventional elements before you reorient the stack direction. You can then convert the region back to a cylindrical or gasket mesh by reassigning cylindrical elements or second-order gasket elements.
- Renumber selected elements within an orphan mesh. You can renumber the selected elements by specifying a starting label and an increment or by offsetting the existing label by a specified value.

Tip: The Mesh Edit Undo feature can roll back any change you make to the element in the mesh. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

For detailed instructions about manipulating elements, see *Editing elements*.

Manipulating the mesh

The Edit Mesh toolset provides the following tools that allow you to manipulate the mesh:

- Create layers of solid elements that are offset normal to selected element faces of an existing mesh, as shown in *Figure 1*. This tool is useful for meshing shell-like parts with solid elements. You can specify the total thickness of all the layers and the number of layers to create. The first layer of elements can be merged with the existing elements, or it can be offset by a specified distance. The selected element faces can be from either shell or solid elements, but not from both. If you selected shell element faces, you can choose whether to delete the shell elements after Abaqus/CAE generates the layers of solid elements.

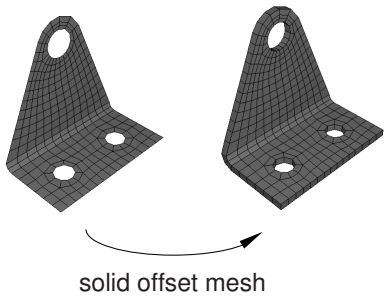


Figure 1: Creating layers of solid elements.

  You should use this tool to generate continuum shell and cohesive elements, because the solid elements that Abaqus/CAE generates are stacked normal to the surface from which they were offset.

- Create layers of shell elements that are offset normal to selected element faces from an existing mesh, as shown in *Figure 2*.

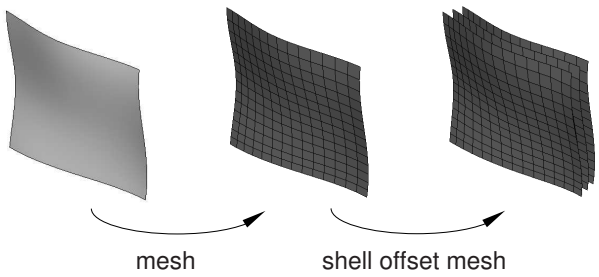


Figure 2: Creating layers of shell elements.

  This tool is similar to the previous tool; however, instead of specifying the total thickness of the layers, you specify the distance between the layers. You should use this tool to create an offset copy of an existing shell mesh and to create multi-layered composite models or reinforcements.

- Wrap a mesh. You can wrap a planar orphan mesh about the global *Z*-axis at a specified radius. The wrapping procedure will relocate a node at point (x, y) on the planar mesh to (r, θ, z), where r is the specified radius, $\theta = \frac{x}{r}$, and $z=y$.

- Collapse edges. You can specify the minimum element edge size desired in the mesh. Abaqus/CAE merges the end nodes of element edges that are shorter than the specified length. By default, Abaqus/CAE checks all element edges in a mesh part, but you can specify a different element domain.

  For structured meshes you can specify a thickness direction or reference edge to limit the edge directions considered for collapse. If you select a thickness direction, Abaqus/CAE uses the direction vector to measure the elements instead of measuring only the distance between nodes. If elements are oriented such that the direction is topologically

ambiguous, Abaqus/CAE highlights the elements so you can either select a domain that excludes them or add a reference edge to further define the desired edges. When you choose a reference edge, with or without a thickness direction, Abaqus/CAE considers edges for collapse only if they are topologically parallel to the reference edge. A reference edge can be applied only within a single section of structured mesh; multiple separate sections must be handled in separate collapse operations if reference edges are required.

You cannot collapse small edges in an element domain that contains quadratic elements.

- Grow edges. You can specify the minimum element edge size that you want to appear in the model, and Abaqus/CAE grows smaller edges to the specified minimum length by adjusting the nodes outward, effectively taking pieces from adjacent edges. Abaqus/CAE cannot increase the length of an edge if adjusting its nodes would cause the adjacent element edges to be too small. However, if you add a thickness direction, Abaqus/CAE will grow all the elements in that direction such that they will meet the minimum size.

  The options for this tool are the same as those for the collapse edges tool; you can specify an element domain other than the entire mesh part, and you can use a thickness direction or a reference edge within a structured mesh to grow only those edges that meet your direction criteria. You cannot grow edges in an element domain that contains quadratic elements.

  Note:

  Use care when specifying an element domain other than the entire mesh. Abaqus/CAE considers for growth only those element edges within the selected domain. If short element edges exist near the border of the domain, Abaqus/CAE does not check the length of adjacent edges outside the domain before growing the ones inside the domain. Growth of edges at the domain border could result in short edges or inverted elements outside the domain.

- Convert triangular elements to tetrahedral elements. You can convert a closed three-dimensional shell of triangular elements into a solid mesh of tetrahedral elements. The triangular shell elements must completely enclose a volume that can be filled with solid elements.
- Convert solid orphan mesh elements to shell orphan mesh elements. This tool creates triangular or quadrilateral shell elements on the free faces of the part. For mesh parts with unmerged nodes that create crack features or mesh parts that have joined faces with incompatible meshes, Abaqus/CAE creates shell elements on both sides of the interface.
- Merge layers. You can select a group of adjacent elements and join them along a specified direction. Each merged element will have the combined shape of the parent elements and the same element type. After selecting the elements to merge, you choose an edge within the elements to be merged to indicate the merge direction, as shown on the left in *Figure 3*. The top two layers of elements were selected using the **by topology** selection method (for more information, see *Using the topology method to select multiple elements*); for clarity the selection is not shown in the figure.

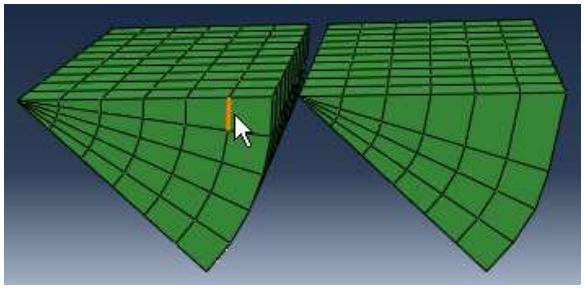


Figure 3: Merging element layers.

You need not select entire layers of elements to merge. However, if you do not select entire layers, Abaqus/CAE will warn you that the new elements will be incompatible with the surrounding mesh. You can then choose to continue or cancel the merge procedure. You cannot merge quadratic elements.

- Subdivide layers. You can select a group of elements and split them by equally dividing the element edges. Abaqus/CAE prompts you to select an edge, a face, or **All Directions** to indicate the direction along which the new layers will be created and enter the number of divisions to make from each element edge. You cannot subdivide quadratic elements.
- Copy a mesh pattern. You can apply a two-dimensional mesh pattern to a similar geometric target on the same part or assembly. After selecting the source mesh and target geometry, you select several nodes and map them to points on the geometry, then Abaqus/CAE finishes the mapping process to apply the mesh pattern to the target.
- Associate mesh with geometry. You can associate orphan or bottom-up mesh entities with a selected geometric entity. Abaqus/CAE prompts you to select a vertex, edge, or face, and then select the node, element edges, or element faces to associate with the geometry. Association can be used to apply loads to a mesh or to create a compatible mesh at the interface between orphan mesh elements and geometry.
- Delete mesh association. You can delete the mesh-geometry association for vertices, edges, faces, or entire geometric regions. You can also delete the association of a native mesh to create an orphan mesh for a portion of a model.
- Insert cohesive seams. You can open pairs of connected elements at crack regions and insert a layer of pore pressure cohesive elements that are created as orphan elements in the gap region. Abaqus/CAE creates several sets and surfaces to help you make selections in subsequent procedures, such as section assignment.



Tip: The Mesh Edit Undo feature can roll back any change you make to the mesh. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

In addition, *Changing the labels of all nodes and elements*, describes how you can renumber all of the nodes and/or elements of a part or selected part instances in the assembly.

For detailed instructions about manipulating the mesh, see *Editing the entire mesh*.

Refining the mesh

The Edit Mesh toolset provides the following tools that allow you to refine a planar mesh of triangular elements:

- Set size. You can specify a global element size for the remesh procedure. Abaqus/CAE changes the density of the new mesh to reflect the new target element size. You can change the element size if the mesh contains either linear or quadratic elements.
- Remove size. You can maintain the global element size during the remesh procedure. Abaqus/CAE maintains the edges of the elements along the boundary of the part while improving the mesh quality in the interior. You can refine the mesh if the mesh contains either linear or quadratic elements.
- Remesh. You can remesh a planar mesh containing linear or quadratic triangular elements.



Tip: The Mesh Edit Undo feature can roll back any mesh refinement changes. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

For detailed instructions about mesh refinement, see *Editing and refining an orphan mesh*.

What is the difference between editing an orphan mesh, a meshed part, and a meshed part instance in the assembly?

You can display either parts containing a combination of orphan and native mesh components or the assembly in the Mesh module. You can use all of the tools in the Edit Mesh toolset to edit an orphan mesh; however, you can use only the following tools to edit a meshed part instance in the assembly:

- Edit, drag, project, merge, and smooth nodes
- Delete elements
- Collapse element edges
- Split an edge of a quadrilateral or triangular element
- Swap the diagonal of a pair of adjacent triangular elements
- Split a quadrilateral element into two triangular elements
- Combine two triangular elements into one quadrilateral element

If you select a part instead of a part instance in the assembly, you can use the mesh editing tools listed above plus the following additional tools:

- Create nodes and elements
- Offset mesh layers to create solid or shell layers
- Associate the mesh with geometry
- Delete mesh associativity
- Copy mesh pattern

In general, you cannot use the Edit Mesh toolset to modify the mesh of a dependent part instance. However, you can edit, drag, and project the nodes of a dependent part instance; Abaqus/CAE modifies the original meshed part, and your modifications appear on all dependent instances of the part.

You can create an orphan mesh in the Mesh module, or you can import an orphan mesh from an output database or an Abaqus input file. You can add geometric features to an orphan mesh in the Part module. An instance of an orphan mesh is always a dependent instance and; therefore, you cannot use the Edit Mesh toolset to edit an instance of an orphan mesh in the assembly. However, you can display the original orphan mesh that was used to create the instance and edit the part. For more information, see *What is the difference between a dependent and an independent part instance?*.

The Mesh Edit Undo feature can roll back any change you make to the mesh. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

Meshing strategies and mesh editing techniques

This section describes strategies for improving your mesh using the Edit Mesh toolset and techniques that you can use to create the desired mesh.

In this section:

- *A strategy for improving the mesh*
- *Using offset meshes in Abaqus/CAE*
- *Reducing element distortion and collapse during mesh offsetting*
- *Allowing for branching in offset solid meshes*
- *Creating a mid-plane shell mesh from a thin solid model*
- *Using a combination of tools to mesh an imported solid part with tetrahedral elements*

A strategy for improving the mesh

If you have a complex assembly, you should select **Mesh->Verify** to verify the quality of the mesh before you submit the job for analysis. The mesh verify tool can do the following:

- Highlight elements of a selected shape that do not meet specified criteria, such as aspect ratio.
- Print mesh statistics, such as the total number of elements of the chosen shape, the number of highlighted elements, and the average and worst values of the selection criterion.
- Highlight elements that do not pass the mesh quality tests that are included with the input file processor in Abaqus/Standard and Abaqus/Explicit.

If the mesh verify tool indicates that you should try to improve the quality of the mesh, you should first try the following before turning to the Edit Mesh toolset:

- Change the seed distribution
- Add or modify partitions
- Change the mesh technique

In addition, you might try modifying the parts in the Part module, or you might try using the Virtual Topology toolset and regenerating the mesh. You should treat the Edit Mesh toolset as the final step in the meshing process and use it only to make minor adjustments to nodes and elements. Abaqus/CAE tries to preserve attributes, such as loads and boundary conditions, if you make changes to the mesh. If you modify a part, Abaqus/CAE deletes the mesh when you return to the Mesh module; as a result, you will lose any edits that you made to the mesh.

Using offset meshes in Abaqus/CAE

The offset mesh tools in the Edit Mesh toolset have many uses in Abaqus/CAE, as follows:

Continuum shell meshes

You can use the solid offset mesh tool to mesh thin solids that are essentially thickened shells, such as sheet metal, composites, or molded plastic components. The solid offset mesh tool is useful for creating continuum shell elements because Abaqus/CAE orients the elements consistently in the thickness direction. For more information, see *Meshing parts with continuum shell elements*.

Cohesive elements

You can assign cohesive elements to quadrilateral elements of a two-dimensional model. For three-dimensional models you can use the solid offset mesh tool to generate a layer of cohesive elements on a mesh. The offset mesh tool generates elements that are oriented consistently with a stack direction that is normal to the thickness direction, and it is a convenient tool for quickly creating a layer of cohesive elements. For more information, see *Embedding cohesive elements in an existing three-dimensional mesh*.

Skin reinforcements

If your skin can be defined as a single layer, creating the skin in the Property module is the preferred approach. However, you can use the shell offset mesh tool to create a skin that uses multiple reinforcement layers. For more information, see *Using offset meshes to create skin reinforcements*.

You can also use the solid offset mesh tool in conjunction with the mesh generation tools in the Mesh module by doing the following:

1. Create a portion of the mesh in the Mesh module.
2. Use the solid offset mesh tool to complete the mesh.

You can create sets containing the elements from an offset mesh. You can create a single set that contains the offset elements, or you can create separate sets for each layer of offset elements. In addition, if you are creating a solid offset mesh, you can create surfaces from the top and the bottom of the offset mesh, provided the first layer of elements is not embedded in the original mesh. You can use these sets and surfaces in subsequent procedures to help you select elements from the offset mesh. For example, you can select a set when assigning a section to a layer of offset elements. Similarly, you can select a surface when you tie one side of a layer of cohesive elements to the surrounding bulk material.

Reducing element distortion and collapse during mesh offsetting

Abaqus/CAE creates an offset mesh by

- offsetting the nodes normal to the boundary of an existing mesh surface, and
- building elements that propagate out in the normal direction.

When you create a mesh that is offset from a concave mesh surface, the elements tend to converge, as shown in *Figure 1*.

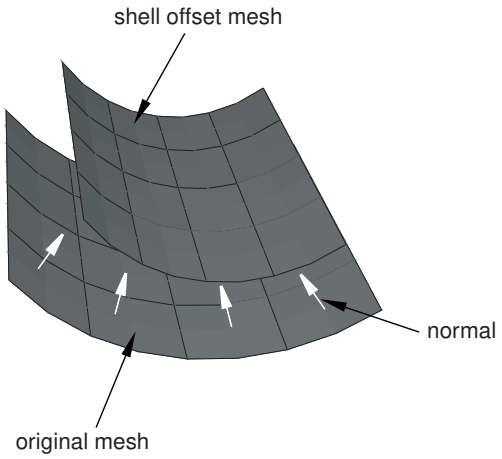


Figure 1: Elements converge when offset from a concave surface.

As a result, elements may collapse or become inverted, and the magnitude of the offset is limited by a combination of the degree of curvature and the element size, as shown in *Figure 2*.

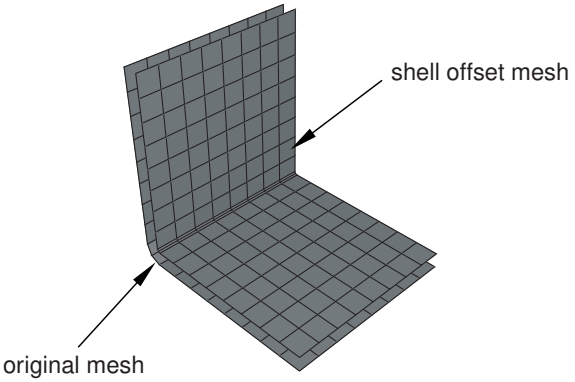


Figure 2: When you offset small elements in a concave region, the elements may collapse or become inverted.

You can try to avoid element collapse and element inversion by selecting **Constant thickness around corners** from the **Offset Mesh** dialog box. When this option is selected, Abaqus/CAE tapers the offset elements that propagate from sharp corners. The resulting elements have a constant nodal offset distance as opposed to a constant distance between the layers of the element faces. *Figure 3* illustrates the effect on the offset mesh pattern of requesting a constant thickness around corners.

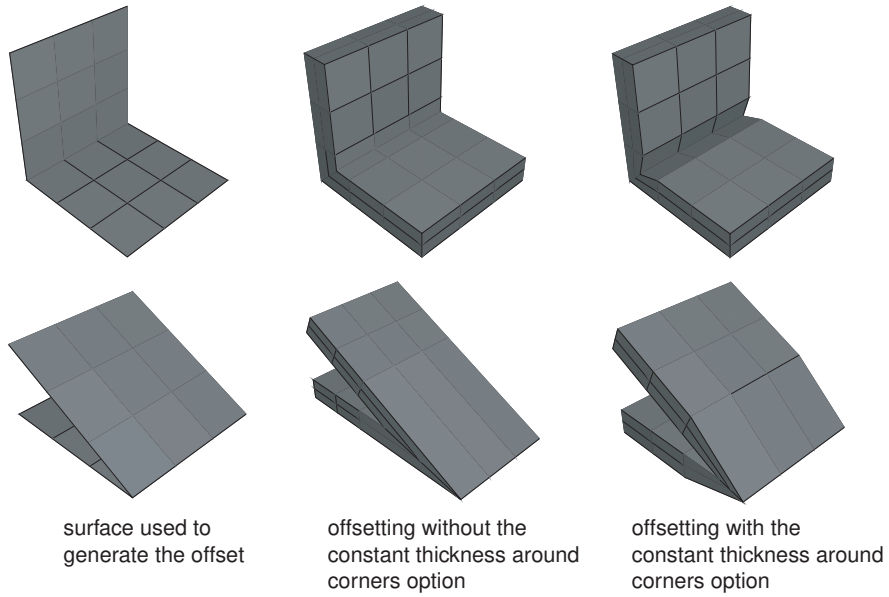


Figure 3: The effect on the offset mesh pattern of requesting a constant thickness around corners.

The effect on the offset mesh pattern is similar for both solid and shell offset meshes. You can also reduce element distortion by using a larger element size in the location of the concave region. However, you can control the element size only if you can regenerate the mesh from which you are offsetting.

In contrast, when you create a mesh that is offset from a convex mesh surface, the elements tend to fan out, as shown in *Figure 4*.

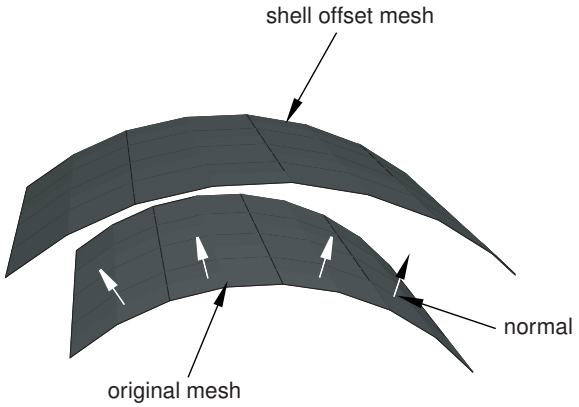


Figure 4: Elements diverge when offset from a convex surface.

Allowing for branching in offset solid meshes

If a part has “branches,” as shown in *Figure 1*, creating an offset solid mesh is not straightforward because the direction in which to offset is ambiguous in the region of the branch.

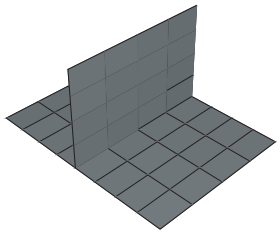


Figure 1: The direction in which to offset is ambiguous in the region of the branch.

To help you mesh this configuration, Abaqus/CAE provides the option to copy the elements in the region of the branch to a set.

You can then do the following:

1. Use display groups to remove the set containing elements in the region of the branch from the orphan mesh.
2. Create offset solid meshes from the non-branching regions that remain, as shown in *Figure 2*.

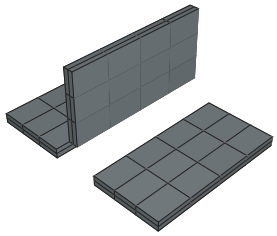


Figure 2: Create offset solid meshes from the non-branching regions.

3. Use display groups to show only the set containing the branch, as shown in *Figure 3*.

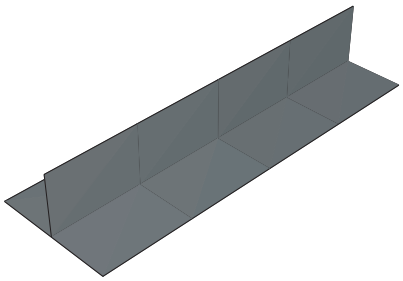


Figure 3: Use display groups to show only the set containing the branch.

4. Create an offset solid mesh from one side of the branching region. Take care that the offset direction is correct (Abaqus/CAE indicates the offset direction by coloring the element faces). In addition, do not delete the base shell elements until you have finished offsetting the branch.
5. Create an offset solid mesh from the other side of the branching region, as shown in *Figure 4*.

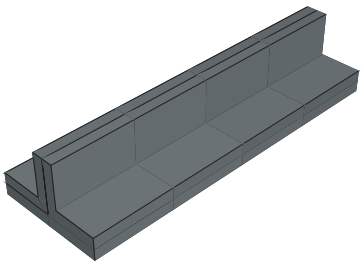


Figure 4: The offset mesh in the branching region.

6. Use display groups to show both regions.
7. Use the merge nodes tool in the Edit Mesh toolset to merge the offset meshes created in Steps 2, 4, and 5.

Creating a mid-plane shell mesh from a thin solid model

You can use the following procedure to create a mid-plane shell mesh from a thin solid model using the offset meshing tool:

1. Convert the solid part to a shell using the **From solid** shell tool in the Part module.
2. Isolate a collection of faces on the upper or lower surface of the part using the **Remove faces** tool in the Geometry Edit toolset.
3. Mesh the simplified model with shell elements, and create a mesh part.
4. Create an offset shell mesh, and specify an **Initial Offset** corresponding to half the model thickness.

Using a combination of tools to mesh an imported solid part with tetrahedral elements

In some cases you may not be able to mesh an imported solid part with tetrahedral elements because of very thin triangular elements in the surface mesh or because some sliver faces cannot be meshed with triangles. The following procedure explains how you can use a combination of tools in the Mesh module to mesh the part successfully:

1. Do one of the following:
   - Start with a tetrahedral boundary mesh. Go to the Mesh module, and create a tetrahedral boundary mesh on the solid.
   - Start with a mesh of linear triangles.
     1. Convert the solid part into a shell part by selecting **Shape->Shell->From Solid** from the main menu bar.
     2. Go to the Mesh module, and mesh the shell part with linear triangular elements.
2. Select **Mesh->Create Mesh Part** from the main menu bar to create a new part that is an orphan mesh. For more information, see *Creating a mesh part*.
3. Change the displayed object to the orphan mesh.
4. Select **Tools->Edit Mesh** from the main menu bar, and do the following to clean the mesh:
   1. From the **Category** field, choose **Mesh**.
   2. From the **Method** list, select **Collapse edges**.

   Cleaning the mesh will automatically merge nodes on short element edges and remove the collapsed elements. For more information, see *Collapsing short edges of a linear orphan mesh*.
5. Use the Edit Mesh toolset to repair any remaining bad elements or gaps manually. For more information, see *Editing elements*.
6. Select **Tools->Edit Mesh** from the main menu bar, and select **Conversion** to replace the shell mesh of linear triangles with a solid mesh of linear tetrahedrons. For more information, see *Converting a triangular shell mesh to a solid tetrahedral mesh*.
7. Go to the Assembly module, and select **Instance->Replace** to replace the original shell part instance with an instance of the orphan mesh.
8. If you want the orphan mesh to use second-order tetrahedral elements, you can assign a quadratic element type to the orphan mesh in the Mesh module.

Using the mesh editing tools

This section is an overview of the tools that are available in the Edit Mesh toolset.

You can access the tools from the main menu bar by selecting **Mesh->Edit**. Abaqus/CAE displays the **Edit Mesh** dialog box. You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

Node editing tools

The following tools are available for editing nodes:

- Create node. For more information, see *Creating a node*.
- Edit node. For more information, see *Editing the position of selected nodes*.
- Drag node. For more information, see *Dragging nodes*.
- Project nodes. For more information, see *Projecting nodes*.
- Delete node. For more information, see *Deleting nodes*.
- Merge nodes. For more information, see *Merging nodes*.
- Adjust midside nodes. For more information, see *Adjusting the position of midside nodes*.
- Smooth external nodes. For more information, see *Smoothing external nodes*.
- Renumber nodes. For more information, see *Renumbering nodes*.

Element editing tools

The following tools are available for editing elements:

- Create element. For more information, see *Creating an element*.
- Delete element. For more information, see *Deleting elements*.
- Flip normal. For more information, see *Reversing the surface normal direction of shell elements*.
- Orient stack direction. For more information, see *Orienting the stack direction*.
- Collapse edge. For more information, see *Collapsing an edge of a quadrilateral or triangular element*.
- Split edge. For more information, see *Splitting an edge of a quadrilateral or triangular element*.
- Swap diagonal. For more information, see *Swapping the diagonal of a pair of adjacent triangular elements*.
- Split. For more information, see *Splitting quadrilateral elements*.
- Combine. For more information, see *Combining triangular elements*.
- Renumber elements. For more information, see *Renumbering elements*.

Mesh editing tools

The following tools are available for editing the entire mesh:

- Offset solid. For more information, see *Generating layers of solid elements offset from an existing mesh*.
- Offset shell. For more information, see *Generating layers of shell elements offset from an existing mesh*.
- Wrap. For more information, see *Wrapping a mesh about the Z-axis*.
- Collapse. For more information, see *Collapsing short edges of a linear orphan mesh*.

- Grow. For more information, see *Growing short edges of a linear orphan mesh*.
- Convert. For more information, see *Converting a triangular shell mesh to a solid tetrahedral mesh*, and *Converting a solid orphan mesh to a shell orphan mesh*.
- Merge. For more information, see *Merging elements*.
- Subdivide. For more information, see *Subdividing elements*.
- Copy mesh pattern. For more information, see *Copying a mesh pattern*.
- View and edit mesh-geometry associativity. For more information, see *Viewing and editing mesh-geometry associativity*.
- Delete mesh-geometry associativity. For more information, see *Deleting mesh-geometry associativity*.
- Insert cohesive seams. For more information, see *Inserting cohesive seams*.

Mesh refinement tools

The following tools are available for refining the mesh:

- Set size.
- Remove size.
- Remesh.

For more information, see *Editing and refining an orphan mesh*.

Undo and redo for mesh edits

You can undo and redo multiple changes that you make using the tools in the Edit Mesh toolset, and control the number of available undo levels. For more information, see *Undoing or redoing a change in the Edit Mesh toolset*.

Editing nodes

This section describes how you can use the Edit Mesh toolset to edit the nodes that comprise orphan meshes and Abaqus/CAE native meshes.

In this section:

- *Creating a node*
- *Editing the position of selected nodes*
- *Dragging nodes*
- *Projecting nodes*
- *Deleting nodes*
- *Merging nodes*
- *Adjusting the position of midside nodes*
- *Smoothing external nodes*
- *Renumbering nodes*
- *Selecting the nodes to edit*

Creating a node

Select **Mesh->Edit** from the main menu bar to create a node by entering its coordinates in either the global coordinates system or a local coordinate system.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part**, and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Create**.

   Abaqus/CAE displays the existing nodes. In hidden and shaded mode, only the visible nodes are displayed.
5. If desired, you can enter the coordinates for the new node in a datum coordinate system that you have created rather than in the global coordinate system:
   a. In the prompt area, click **Select**.
   b. In the viewport, select the datum coordinate system to be associated with the coordinates of the new node. (For more information, see *Creating datum coordinate systems*.)
6. In the **Coordinates** field in the prompt area, enter the coordinates of the new node.

   Abaqus/CAE creates the new node.

   Tip: If you make a mistake while creating a node, click **Undo** in the **Edit Mesh** dialog box to remove the most recently created node.

7. Repeat Steps 3 and 4 as often as necessary to create additional nodes.
8. When you have finished creating nodes, click the cancel button in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*

Editing the position of selected nodes

You can edit the position of one or more selected nodes from a meshed part, a meshed part instance in the assembly, or an orphan mesh.
In addition, if you select multiple nodes to edit, you can assign a single value to a particular coordinate for each of the nodes. For example, you can select a group of nodes and assign an *X*-axis coordinate of `10.0` to each of the nodes. This is particularly useful when you want to position the nodes in the same plane; for example, when defining the mating surface of a cylinder head or engine block.

In general, you cannot use the Edit Mesh toolset to edit a dependent part instance; however, you can edit the nodes of a dependent part instance. Abaqus/CAE moves the nodes of the original mesh, and your modifications appear on all dependent instances of the part.



Note:

You can undo or redo the changes you make to nodal position directly from the **Undo** portion of the **Edit Nodes** dialog box. The **Undo** and **Redo** buttons in this dialog box provide rollback control for nodal position changes only; if you want to undo or redo other mesh editing operations, use the **Edit Mesh** dialog box. For more information about edit mesh undo, see *Undoing or redoing a change in the Edit Mesh toolset*.

1. Enter the Mesh module, and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part**, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Edit**.

      Abaqus/CAE displays the existing nodes. In hidden and shaded mode, only the visible nodes are displayed.
4. From the prompt area, choose the selection method and select the nodes to edit. For more information, see *Selecting the nodes to edit*.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

   Abaqus/CAE displays the **Edit Nodes** dialog box and the minimum and maximum values of the coordinates of the selected nodes along each axis.
5. If desired, you can enter the new coordinates for the nodes in a datum coordinate system that you have created rather than in the global coordinate system:
   a. At the top of the **Edit Nodes** dialog box, click **Select**.
   b. In the viewport, select the datum coordinate system to be associated with the new nodal coordinates. (For more information, see *Creating datum coordinate systems*.)

6. To specify the new position of the selected nodes by offset, choose **Offsets** from the **Specification method** options, then do one of the following:

Enter values

Type the 1-, 2-, and 3-components of the offset from the current location directly into the dialog box.

Measure offset vector

Click  and pick two points in the viewport. Abaqus/CAE populates the 1-, 2-, and 3-components of the offset based on the components of the vector from your first selected point to your second selected point.

7. To specify the new position of the selected nodes by coordinates, choose **Coordinates** from the **Specification method** options, then do one of the following:

Enter values

Type the 1-, 2-, and 3-components directly into the dialog box. A value of **As is** indicates that the coordinate along a particular axis is different for at least two of the selected nodes; click the arrow to the right of the component field, and select **Specify** to specify a value.

Measure coordinates

Click  and pick the new point from the viewport.

8. If you are editing nodes on the boundary of an Abaqus/CAE native mesh in the Mesh module, Abaqus/CAE first moves the nodes to the new position and then projects them back to the geometry on which they were originally located. As a result, nodes that were located on a surface will remain on that surface but in a different position. Toggle off **Project to geometry** to move the nodes to their new position independently from the underlying geometry.
9. Click **OK** or **Apply** to move the nodes to the new position. If you click **OK**, Abaqus/CAE moves the nodes and prompts you to select new nodes to edit. If you click **Apply**, Abaqus/CAE moves the nodes; however, the nodes remain selected, and you can continue specifying a new position.



Tip: If you make a mistake while editing nodes, click **Undo** in the **Edit Mesh** dialog box to undo the edit. Abaqus/CAE restores the node or nodes to their original position.

10. Repeat the previous steps as often as necessary to edit additional nodes.
11. When you have finished editing nodes, click **Cancel** to close the **Edit Nodes** dialog box.

Additional information

- *What can I do with the Edit Mesh toolset?*

Dragging nodes

You can drag nodes on a meshed part, a meshed part instance in the assembly, or an orphan mesh. Dragging provides a quick way to reposition nodes and to improve element quality. When you activate the node dragging function, Abaqus/CAE automatically activates the solver element quality checks, highlighting elements with warnings in yellow and highlighting elements with errors in magenta. You can also include tests for shape metrics or size metrics among the tests that Abaqus/CAE automatically performs on elements as you drag nodes. The highlighting is updated as you drag a node into a new position; therefore, you can immediately see whether you are improving the element quality. In general, you cannot use the Edit Mesh toolset to modify the mesh of a dependent part instance; however, you can drag the nodes of a dependent part instance. Abaqus/CAE modifies the nodes of the original mesh, and your modifications appear on all dependent instances of the part.

Dragging nodes does not change their association. Nodes that are associated with a geometric feature prior to dragging will remain associated with that feature; unassociated nodes will remain unassociated after you drag them onto geometry.

1. Enter the Mesh module, and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part**, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Drag**.

      Abaqus/CAE displays the existing nodes.
4. If desired, click **Element failure criteria** from the prompt area to include shape metrics or size metrics among the tests that Abaqus/CAE automatically performs as you drag nodes.
5. From the prompt area, toggle **Project to geometry** to control the drag behavior of exterior nodes.
   - If you toggle on **Project to geometry**, Abaqus/CAE constrains dragged exterior nodes to their associated geometry. You can drag a node on a mesh face as long as it remains within the underlying geometric face. Likewise, you can drag an edge node only along the polyline that created the edge, and you cannot drag a node associated with a vertex. In orphan mesh regions, you can drag any node; but you cannot drag a node beyond the outer edges of the mesh.
   - If you toggle off **Project to geometry**, Abaqus/CAE constrains dragged nodes to a plane parallel to the computer screen and passing through the original position of the node. Therefore, you can change the available drag locations by using the view manipulation tools before dragging a node. This is also the behavior if you drag interior nodes since they are not constrained to a geometric feature.
6. Position the cursor over the node you want to move, and click and hold mouse button 1 while dragging the node to a new location.

   When you release mouse button 1, Abaqus/CAE displays the node number and new coordinates in the message area.

7. Repeat the previous steps as often as necessary to drag additional nodes.

Additional information

- *What can I do with the Edit Mesh toolset?*
- *Verifying element quality*

Projecting nodes

You can project one or more selected nodes from a meshed part, a meshed part instance in the assembly, or an orphan mesh to a selected vertex, edge, face, node, element edge, element face, datum point, datum axis, or datum plane. This is particularly useful if you want to move nodes in a bottom-up mesh onto nearby geometry. In general, you cannot use the Edit Mesh toolset to modify the mesh of a dependent part instance; however, you can project the nodes of a dependent part instance. Abaqus/CAE projects the nodes of the original mesh, and your modifications appear on all dependent instances of the part.

Projecting nodes does not change their association. Nodes that are associated with a geometric feature prior to projection will remain associated with that feature; unassociated nodes will remain unassociated after you project them onto geometry.

1. Enter the Mesh module, and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part**, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Project**.

      Abaqus/CAE displays the existing nodes. In hidden and shaded mode, only the visible nodes are displayed.
4. From the prompt area, choose the selection method and select the nodes to project. For more information, see *Selecting the nodes to edit*.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.
5. Select a vertex, edge, face, node, element edge, element face, datum point, datum axis, or datum plane to which Abaqus/CAE will move the selected nodes.

   Abaqus/CAE colors the selected entity magenta.
6. Select **Yes** in the prompt area to confirm your selections.

   Abaqus/CAE projects the selected nodes to their new positions using a shortest distance algorithm. If you selected an edge, face, element edge, or element face, Abaqus/CAE projects the nodes onto the axis or plane that would be created by extending the selected edge or face.

   Tip: If you make a mistake while editing nodes, click **Undo** in the **Edit Mesh** dialog box to undo the edit. Abaqus/CAE restores the nodes to their original positions.

   If you select **No**, your node and entity selections are cancelled and the procedure continues from Step 4.

7. Repeat the previous steps as often as necessary to project additional nodes.

Additional information

• *What can I do with the Edit Mesh toolset?*

Deleting nodes

Select **Mesh->Edit** from the main menu bar to delete orphan mesh nodes. Abaqus/CAE also deletes any elements associated with the deleted nodes. If the nodes and associated elements that you are deleting belong to existing node or element sets, those sets are updated accordingly.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part**, and select a part from the list. This tool is available only for deleting orphan mesh nodes.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Delete**.

      If the **Delete** method is not shown, there are no orphan nodes in the selected part.

      Abaqus/CAE displays the existing nodes. In hidden and shaded mode only the visible nodes are displayed.

5. In the viewport, select the nodes to delete. You can **[Shift] + Click** on individual nodes to add them to your selection, or **[Ctrl] + Click** on selected nodes to remove them from your selection.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

   If you would rather select from a list of existing node sets, do the following:
   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of node sets that you have created.
   b. Select the set of nodes that you want to delete, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   Abaqus/CAE displays the selected nodes and the elements associated with the selected nodes. (Both the nodes that you have selected and any elements associated with those nodes will be deleted.)
6. When you have finished selecting nodes to delete, click mouse button 2.
7. The elements associated with the selected nodes may also be associated with unselected nodes. If those unselected nodes would be left unassociated with any element after the nodes and associated elements are deleted, click **Yes** in the prompt area if you want Abaqus/CAE to delete those unselected nodes as well.

   Tip: If you make a mistake while deleting nodes, click **Undo** in the **Edit Mesh** dialog box to undo the deletion. Abaqus/CAE restores the deleted node or nodes.

8. Repeat the previous steps as often as necessary to delete additional nodes.

9. When you have finished deleting nodes, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

Additional information

• *What can I do with the Edit Mesh toolset?*

Merging nodes

Select **Mesh->Edit** from the main menu bar to merge nodes. If you select only two nodes to merge, Abaqus/CAE creates a new node at the midpoint of the selected nodes. If you select more than two nodes, you can specify the maximum distance between nodes, and Abaqus/CAE will merge nodes that are closer than the specified distance. You can also choose to remove duplicate elements that have the same connectivity.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part**, and select a part from the list. This tool is available only for working on parts that contain either orphan mesh nodes or nodes created using the bottom-up meshing procedure.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Merge**.
5. From the prompt area, choose the selection method and select the nodes to merge. For more information, see *Selecting the nodes to edit*.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

6. If you selected only two nodes to merge, Abaqus/CAE creates a new node at the midpoint of the selected nodes.
7. If you selected more than two nodes to merge, do the following:
   a. In the prompt area, enter the **Node merging tolerance** to specify the maximum distance between nodes that you want to merge.
   b. If the value that you entered for the **Node merging tolerance** is too large, Abaqus/CAE may detect duplicate nodes from the same element. Abaqus/CAE will not merge nodes from the same element, but the large tolerance can result in a distorted mesh. If the **Node merging tolerance** is too large, Abaqus/CAE asks if you want to continue merging the selected nodes.
      - Click **Yes** to continue.
      - Click **No** to cancel the merging procedure.

   c. Abaqus/CAE highlights in magenta any nodes that will be merged and asks if you wish to proceed.
      - Click **Yes** to continue.
      - Click **No** to cancel the merging procedure.

      Abaqus/CAE merges the nodes that are closer than the specified distance and replaces them with a single new node. The location of the new node is the average position of the group of nodes that were merged into the new node.

      You can also merge part instances in the Assembly module. For more information, see *Performing Boolean operations on part instances*.

If you previously created node or element sets that include nodes or elements that are deleted by the merge operation, Abaqus/CAE updates your sets accordingly.



Tip: If you make a mistake while merging nodes, click **Undo** in the **Edit Mesh** dialog box to undo the merge. Abaqus/CAE restores the nodes as they were before the merge.

8. Repeat the above steps as often as necessary to merge additional nodes.
9. When you have finished merging nodes, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*

Adjusting the position of midside nodes

Select **Mesh->Edit** from the main menu bar to adjust the position of midside nodes of second-order elements.

You can use this tool to allow for the singularity at the crack tip in a fracture mechanics analysis. However, this tool will allow you to edit any second-order elements; it does not check that you are editing the degenerate quadrilateral or wedge elements required by a fracture mechanics analysis. For more information, see *Controlling the singularity at the crack tip for a small-strain analysis* and *Constructing a Fracture Mechanics Mesh for Small-Strain Analysis with the Conventional Finite Element Method*.

You can select nodes and enter a bias parameter between 0 and 1. Abaqus/CAE moves the midside nodes along connected element edges to a position based on the parameter that you entered. For example, if you enter a parameter of 0.25, Abaqus/CAE biases the position of the midside nodes one quarter of the length of the element edge away from the selected nodes.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part**, and select a part from the list. This tool is available only for working on an parts that contain orphan mesh nodes.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Adjust midside**.

   Abaqus/CAE displays the existing nodes. In hidden and shaded mode only the visible nodes are displayed.
5. Select the nodes toward which the position of midside nodes along connected element edges will be adjusted. You can select nodes, or you can select element edges. If you select element edges, Abaqus/CAE selects the nodes along the element edges.

   You can **[Shift] + Click** on individual nodes to add them to your selection, or **[Ctrl] + Click** on selected nodes to remove them from your selection.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

   If you would rather select from a list of existing node sets, do the following:
   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of node sets that you have created.
   b. Select the set of nodes, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
6. In the prompt area, enter the value of the bias parameter. You must enter a value between 0 and 1.

Abaqus/CAE moves the midside nodes along connected element edges to the specified position.



Tip: If you make a mistake while adjusting nodes, click **Undo** in the **Edit Mesh** dialog box to undo the adjustment. Abaqus/CAE restores the nodes to their previous positions.

7. Repeat Steps 4 and 5 as often as necessary to adjust additional midside nodes.

8. When you have finished adjusting nodes, click the cancel button  in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*
- *Controlling the singularity at the crack tip for a small-strain analysis*
- *Constructing a Fracture Mechanics Mesh for Small-Strain Analysis with the Conventional Finite Element Method*

Smoothing external nodes

Select **Mesh->Edit** from the main menu bar to smooth the positions of nodes on the surfaces of parts or assemblies. The node smoothing operation can improve transitions in mesh density or otherwise improve local mesh quality without remeshing the region or manually repositioning nodes. You can use node smoothing to improve the mesh on a surface before using it as the source of a bottom-up mesh or a swept mesh.

Abaqus/CAE applies a Laplacian smoothing algorithm at each node that you select. The selected nodes are repositioned to equalize their distance from adjoining nodes, creating a smoother surface mesh.

You can select only external nodes of parts or assemblies for smoothing. Boundary nodes are not repositioned by the smoothing operation.

1. Enter the Mesh module, and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part**, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Node**.
   b. From the **Method** list, select **Smooth**.

   Abaqus/CAE displays the existing nodes. In hidden and shaded mode only the visible nodes are displayed.
4. Select the nodes to be smoothed.

   You can **[Shift] + Click** on individual nodes to add them to your selection, or **[Ctrl] + Click** on selected nodes to remove them from your selection.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

   If you would rather select from a list of existing node sets, do the following:
   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of node sets that you have created.
   b. Select the set of nodes, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.
5. In the prompt area, click **Done**.

   Abaqus/CAE smooths the position of the selected nodes.



Tip: If you want to restore the nodes to their previous positions, click **Undo** in the **Edit Mesh** dialog box to undo the smoothing.

6. Repeat Steps 4 and 5 as often as necessary to smooth additional nodes.

   You can apply smoothing to the same node multiple times. After a few iterations, a node will no longer be moved by the smoothing algorithm unless other surrounding nodes are moved.

7. When you have finished adjusting nodes, click the cancel button  in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*

Renumbering nodes

The Edit Mesh toolset allows you to renumber nodes.

Select **Mesh->Edit** from the main menu to renumber selected nodes of an orphan mesh. You can renumber the selected nodes by specifying a starting node label and an increment or by offsetting the existing node label by a specified value. (For information on renumbering the nodes of an Abaqus/CAE native mesh, see *Changing the labels of all nodes and elements*.)

1. Enter the Mesh module.

   This tool is available only for working on parts that do not contain any geometry. You can work in the part or assembly context. However, any renumbering applied to a part instance is applied at the part level and, therefore, affects all instances of that part.

2. From the main menu, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. In the dialog box, do the following:

   a. In the **Category** field, select **Node**.

   b. From the **Method** list, select **Renumber**.

   Abaqus/CAE displays the **Node Renumbering** dialog box. If a single mesh part is displayed in the viewport, the **Part Label Range** displays the range of node labels for the entire part.

4. Do either of the following to specify the renumbering method:

   - Choose **By Start Label**, and enter the value of the first node label and the increment that will be used to number subsequent nodes.
   - Choose **By Offset**, and enter the value by which the existing node labels will be offset.

5. Use one of the following methods to select the nodes that you want to renumber:

Selecting all nodes of a part

- Select **All** (in the part context) or **Part** (in the assembly context) to renumber all the orphan mesh nodes of a part. If there is more than one mesh part instance in the viewport, click the  button to select a part instance.

Selecting nodes using the unordered method

1. Select **Specify** and **Unordered** to select the nodes in any sequence.
2. Click **Select** to select the nodes to renumber.
3. From the prompt area, choose the selection method and select the nodes to renumber. For more information, see *Selecting the nodes to edit*.

Selecting nodes using the directed path method

1. Select **Specify** and **Directed Path** to renumber the nodes along a feature edge from a selected start node to a selected end node.
2. Click **Select** to select the nodes to renumber.
3. Select the start node. Abaqus/CAE displays all the nodes along the feature edge. If desired, modify the angle to redefine how Abaqus/CAE defines a feature edge.
4. Select **Done** from the prompt area. The start node is displayed in purple.
5. Select the end node. The start and end nodes must lie on the same feature edge.

Selecting nodes using the sequence method

1. Select **Specify** and **Sequence** to renumber the nodes based on the sequence in which you select the nodes.
2. Click **Select** to select the nodes to renumber.
3. Select the start node and each additional node in sequence.
4. When you have finished selecting nodes, click mouse button 2.

The **Part Label Range** displays the range of node labels for the renumbered nodes.

6. Click **OK** or **Apply** to renumber the nodes. If you click **Apply**, Abaqus/CAE renumbers the nodes; however, the nodes remain selected, and you can continue renumbering them.
7. Repeat the previous steps as often as required to renumber additional nodes.
8. When you have finished renumbering nodes, click **Cancel** to close the **Node Renumbering** dialog box.

Additional information

- *Changing the labels of all nodes and elements*
- *What can I do with the Edit Mesh toolset?*

Selecting the nodes to edit

When you are selecting the nodes to edit, Abaqus/CAE displays a menu in the prompt area that allows you to choose the technique that you will use to select the nodes. The techniques allow you to select individual nodes or groups of adjacent nodes.

1. Use one of the following methods to select the nodes that you want to edit:

To select individual nodes:

1. From the menu in the prompt area, select **Individually**.
2. Select a node that you want to renumber, or drag-select a group of nodes.
3. **[Shift] + Click** on additional nodes to add them to your selection.
4. If necessary, **[Ctrl] + Click** on selected nodes to remove them from your selection.

To select nodes using the angle method:

1. From the menu in the prompt area, select **by angle**.
2. Enter an angle (from 0° to 90°), and select a node.

   Abaqus/CAE selects every node on element faces adjacent to the selected node until the angle between the element faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects*, for more information.)
3. After you use the **by angle** method, you can **[Shift] + Click** on additional nodes to add them to your selection or **[Ctrl] + Click** on selected nodes to remove them from your selection. (See *Combining selection techniques*, for more information.)

To select nodes using the feature edge method:

1. From the menu in the prompt area, select **by feature edge**.
2. Enter an angle (from 0° to 90°).

   Abaqus/CAE identifies all the feature edges in your model by finding all the element edges where the angle between two adjacent element faces is greater than the angle specified.
3. Select an element edge or node.

   Abaqus/CAE follows the feature edge that passes through the selected element edge or node. The feature edge is truncated if another feature edge intersects it at an angle greater than the angle specified in the previous step.
4. Abaqus/CAE selects all the nodes along the feature edge. (See *Using the angle and feature edge method to select multiple objects*, for more information.) Alternatively, you can select the node at the intersection of multiple feature edges, and Abaqus/CAE selects the nodes along all these feature edges.
5. After you use the **by feature edge** method, you can **[Shift] + Click** on additional nodes to add them to your selection or **[Ctrl] + Click** on selected nodes to remove them from your selection. (See *Combining selection techniques*, for more information.)

To select nodes by specifying an existing node set:

1. Click **Sets** on the right side of the prompt area. Abaqus/CAE displays the **Region Selection** dialog box containing a list of node sets that you have created.
2. Select the set of nodes that you want to renumber, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

2. When you have finished selecting nodes, select **Done** from the prompt area.

Additional information

- *Selecting objects within the current viewport*
- *What can I do with the Edit Mesh toolset?*

Editing elements

This section describes how you can use the Edit Mesh toolset to edit the elements that comprise orphan meshes and Abaqus/CAE native meshes.

In this section:

- *Creating an element*
- *Deleting elements*
- *Reversing the surface normal direction of shell elements*
- *Orienting the stack direction*
- *Collapsing an edge of a quadrilateral or triangular element*
- *Splitting an edge of a quadrilateral or triangular element*
- *Swapping the diagonal of a pair of adjacent triangular elements*
- *Splitting quadrilateral elements*
- *Combining triangular elements*
- *Renumbering elements*
- *Selecting the elements to edit*

Creating an element

Select **Mesh->Edit** from the main menu bar to create elements from selected nodes.

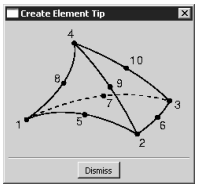
1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Part module and Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Create**.

   Abaqus/CAE displays the existing nodes. In hidden and shaded mode only the visible nodes are displayed.
5. In the prompt area, click the arrow next to the **Element shape** field, and select the element shape of your choice from the list that appears.
6. In the viewport, select the nodes that will define the element. You must select the required number of nodes for the element shape specified in the previous step. In addition, you must select the nodes in a specific order. From the right of the prompt area, click **Tip** to see a figure of the element shape and connectivity. For example, Abaqus/CAE displays the following figure when you are creating a 10-node tetrahedral element:

   Once you have selected the required number of nodes, Abaqus/CAE creates the new element.

   Tip: If you make a mistake while creating an element, click **Undo** in the **Edit Mesh** dialog box to remove the most recently created element.

7. Repeat the previous steps as often as necessary to create additional elements. If you change the element shape, click the **Tip** button again to update the element shape and connectivity figure.
8. When you have finished creating elements, click mouse button 2 or the cancel button in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*

Deleting elements

Select **Mesh->Edit** from the main menu bar to delete elements. You have the option of instructing Abaqus/CAE to delete any nodes that are left unassociated with any elements once the elements are deleted. If the elements and associated nodes that you are deleting belong to existing element or node sets, those sets are updated accordingly.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list. This tool is available only for working on orphan mesh elements or for working on elements created using the bottom-up meshing procedure
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Delete**.
5. If you want to delete all nodes that are left unassociated with any elements once the selected elements are deleted, toggle on **Delete associated unreferenced nodes** in the prompt area.
6. Select the elements to delete. You can **[Shift] + Click** on elements to add them to your selection or **[Ctrl] + Click** on selected elements to remove them from your selection.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

   If you would rather select from a list of existing element sets, do the following:
   a. Click **Sets** on the right side of the prompt area.

      Abaqus/CAE displays the **Region Selection** dialog box containing a list of element sets that you have created.
   b. Select the set of elements that you want to delete, and click **Continue**.

   Note: The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   Abaqus/CAE displays the selected elements.
7. When you have finished selecting elements to delete, click mouse button 2.

   Abaqus/CAE deletes the selected elements. In addition, if you toggled on **Delete associated unreferenced nodes**, Abaqus/CAE also deletes nodes that would be left unassociated with any elements once the selected elements are deleted. The elements and nodes are also removed from any existing sets.

   Tip: If you make a mistake while deleting elements, click **Undo** in the **Edit Mesh** dialog box to undo the deletion. Abaqus/CAE restores the deleted element or elements.
8. Repeat the above steps as often as necessary to delete additional elements.

9. When you have finished deleting elements, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

Additional information

• *What can I do with the Edit Mesh toolset?*

Reversing the surface normal direction of shell elements

Select **Mesh->Edit** from the main menu bar to reverse the surface normals of selected shell elements. The elements can be either quadrilaterals or triangles. During the procedure to reverse the surface normals, Abaqus/CAE displays the part using the shaded render style, and the front and back faces of each shell element appear in different colors. If you change the direction of the surface normal in a region, Abaqus/CAE does not change the orientation of prescribed conditions or named surfaces applied to that region.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Flip Normal**.

      Abaqus/CAE displays the part using the shaded render style. The front of each element is colored brown, and the back of each element is colored purple.

5. From the prompt area, choose the type of region, **Geometry** or **Mesh**, and toggle **Highlight conflicting edges** to display or hide a yellow border between connecting elements that have conflicting normals.
6. From the prompt area, choose the selection method and select the shell region or orphan shell elements whose normals you want to reverse. For more information, see *Selecting the elements to edit*.
7. When you have finished selecting regions or elements, select **Done** from the prompt area.

   If you selected geometric regions, Abaqus/CAE changes the normals of the elements and returns to Step 5.
8. If you selected orphan elements, choose a method for reversing the surface normal:
   - Click **Flip all** to reverse the normal of all selected elements.
   - Click **Select normal** to change the normals of the selected elements so that they point in the same direction as the normal of a reference element that you specify.
9. If you chose **Select normal** in the previous step, do the following:
   a. In the viewport, select the reference element.
   b. In the prompt area, click **OK**.

      Abaqus/CAE changes the normals of the selected elements so that they point in the same direction as the reference element normal and returns to Step 5 so you can make more selections.
10. Click mouse button 2 to exit the procedure.

    Abaqus/CAE reverts to the render style that was selected prior to starting the procedure to reverse the surface normals.

Tip: If you make a mistake while reversing the surface normals of orphan elements, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original surface normals for the selected elements.

If you make a mistake while reversing the surface normals of native elements in a geometric region, select the region and reverse the normals again.

Additional information

• *What can I do with the Edit Mesh toolset?*

Orienting the stack direction

Select **Mesh->Edit** from the main menu bar to orient the stack direction of elements in a continuum shell, cohesive, cylindrical, or gasket mesh. You can orient only hexahedral, wedge, and quadrilateral elements because these are the only elements that can be stacked to form a continuum shell, cohesive, cylindrical, or gasket mesh. If you have assigned cylindrical elements or second-order gasket elements to the region, you must first change the assignment to conventional elements before you reorient the stack direction. You can then convert the region back to a cylindrical or gasket mesh by reassigning cylindrical elements or second-order gasket elements. If you change the orientation of the stack direction of elements in a region, Abaqus/CAE does not change the orientation of prescribed conditions or named surfaces applied to that region.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list. This tool is available only for working on orphan mesh elements. To change the stack direction of native elements, see *Applying a mesh stack orientation*.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Orient stack direction**.

      For hexahedral and wedge elements, Abaqus/CAE colors the bottom face of the mesh stack purple and the top face brown. Similarly, an arrow indicates the orientation of quadrilateral elements. In addition, Abaqus/CAE highlights in red any element faces and edges that have inconsistent orientation. For more information, see *Continuum shells*, *Creating a model with cohesive elements using geometry and mesh tools*, *Gaskets*, and *Swept meshing of cylindrical solids*.

5. In the viewport, select the elements to orient. You can **[Shift] + Click** on individual elements to add them to your selection or **[Ctrl] + Click** on selected elements to remove them from your selection.

   Tip: You can use the **Selection** toolbar to change the picking behavior. For more information, see *Selecting objects within the viewport*.

6. When you have finished selecting elements to orient, click mouse button 2.
7. If your part is meshed with hexahedral or wedge elements, pick a face from one of the selected elements. Abaqus/CAE uses the picked face to represent the top face of the reference orientation. If your part is meshed with quadrilateral elements, pick an edge from one of the selected elements. Abaqus/CAE orients the elements toward the picked edge.

   Tip: If you make a mistake while orienting elements, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original orientation for those elements.

8. Repeat the previous steps as often as necessary to orient additional elements.

9. When you have finished orienting elements, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*
- *About Shell Elements*
- *Creating a model with cohesive elements using geometry and mesh tools*
- *Continuum shells*
- *Gaskets*

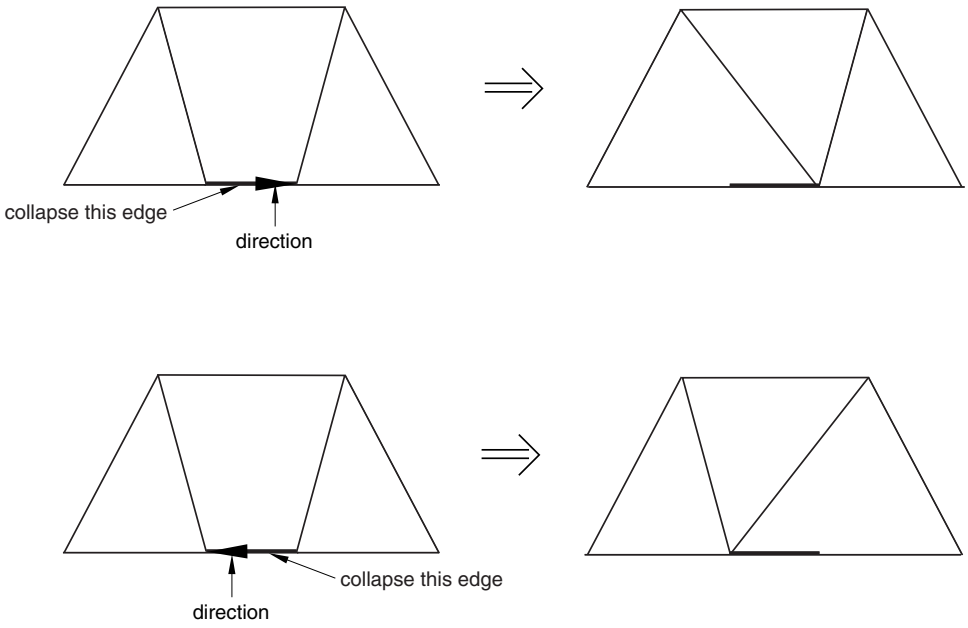
Collapsing an edge of a quadrilateral or triangular element

The Edit Mesh toolset allows you to replace the two nodes at either end of particular element edges with a single node.

Select **Mesh->Edit** from the main menu bar to collapse a selected edge of a triangular or quadrilateral element. In effect, you are replacing the two nodes at either end of the element edge with a single node. You can choose between two methods for positioning the single node:

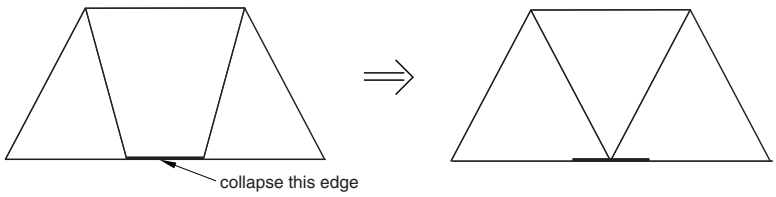
direction

Choose **direction** to collapse the element edge in the specified direction. From the buttons in the prompt area, select **Flip** to change the direction in which Abaqus/CAE will collapse the element. The following figure shows how the **direction** method collapses an element edge along with the effect of changing the direction:



average

Choose **average** to collapse the element edge and readjust the remaining element edges to meet at the midpoint of the collapsed edge. The following figure shows how the **average** method collapses an element edge:



1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.



Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Collapse edge (tri/quad)**.
5. From the menu in the prompt area, choose the method that Abaqus/CAE will use to collapse the edge.
   - Choose **direction** to collapse the element edge in the specified direction.
   - Choose **average** to collapse the element edge and readjust the remaining elements to meet at the midpoint of the collapsed edge.
6. Select the element edge to collapse. You can select only a single edge.

   Abaqus/CAE highlights the selected edge and prompts you to confirm that you want to collapse the selected edge.
7. From the buttons in the prompt area, click **Yes** to collapse the selected edge.

   Abaqus/CAE prompts you to select the next edge to collapse.
8. Repeat the previous steps as often as necessary to collapse additional edges.
9. When you have finished collapsing edges, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.



Tip: If you make a mistake while collapsing edges, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the mesh to its state just before the operation.

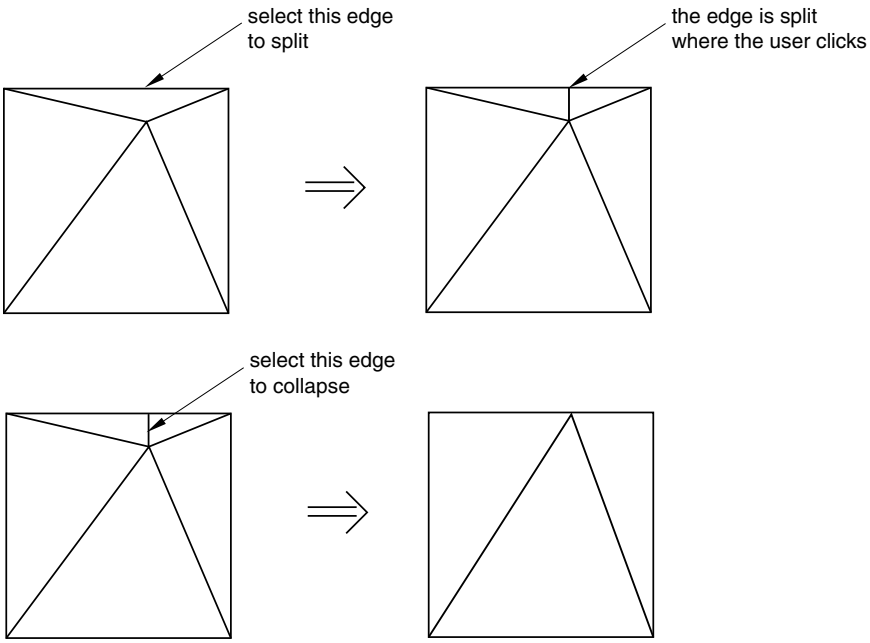
Additional information

• *What can I do with the Edit Mesh toolset?*

Splitting an edge of a quadrilateral or triangular element

The Edit Mesh toolset allows you to split an edge of a quadrilateral or triangular element into two parts.

Select **Mesh->Edit** from the main menu bar to split an edge of a quadrilateral or triangular element into two parts. The following figure shows how you can split an edge and then collapse the resulting edge to remove a long narrow triangular element:



When you split a quadrilateral element, Abaqus/CAE creates the edges that will result in the best-shaped elements based on angle measurements. You cannot select which edges Abaqus/CAE will create. However, after Abaqus/CAE splits the edge, you can use the **Swap diagonal (tri)** tool to change the diagonal; for more information, see *Swapping the diagonal of a pair of adjacent triangular elements*.

1. Enter the Mesh module and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part** and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Split edge (tri/quad)**.
4. From the menu in the prompt area, select the method that you will use to determine the location of the split along the edge. You can choose either **midpoint** or **picked location**.

5. If you chose **midpoint** to determine the location, select the edge to split. You can select only a single edge.

   Abaqus/CAE highlights the edges it will insert to split the selected edge at the midpoint.

6. If you chose **picked location** to determine the location, click on the edge at the location of the desired split.

   Abaqus/CAE highlights the edges it will insert to split the selected edge at the picked location.

7. From the prompt area, click **Split Edge**.

   Abaqus/CAE splits the edge and prompts you to select the next edge to split.

   Tip: If you make a mistake while splitting element edges, click **Undo** in the **Edit Mesh** dialog box to undo the split.

8. Repeat the above steps as often as necessary to split additional edges.

9. When you have finished splitting edges, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

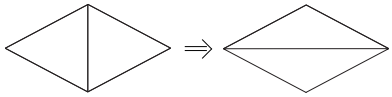
Additional information

• *What can I do with the Edit Mesh toolset?*

Swapping the diagonal of a pair of adjacent triangular elements

The Edit Mesh toolset allows you to swap the diagonal of a pair of adjacent triangular elements.

Select **Mesh->Edit** from the main menu bar to swap the diagonal of a pair of adjacent triangular elements, as shown in the following figure:



The adjacent triangular elements must have consistent normals. If necessary, you can use the Edit Mesh toolset to flip the normals; for more information, see *Reversing the surface normal direction of shell elements*.

1. Enter the Mesh module and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part** and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Swap diagonal (tri)**.
4. Select the edge of the adjacent triangular elements to swap. You can select only a single edge.

   Abaqus/CAE highlights the selected edge and prompts you to confirm that you want to swap the selected edge.
5. Click **Yes** to swap the element edge.

   Abaqus/CAE prompts you to select the next element edge to swap.

   Tip: If you make a mistake while swapping a diagonal, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original diagonal in the two adjacent elements.
6. When you have finished swapping diagonals, click mouse button 2 or the cancel button in the prompt area to exit the procedure.

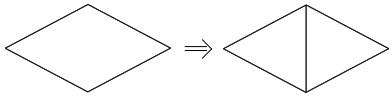
Additional information

- *What can I do with the Edit Mesh toolset?*

Splitting quadrilateral elements

The Edit Mesh toolset allows you to split quadrilateral elements into two triangular elements.

Select **Mesh->Edit** from the main menu bar to split quadrilateral elements into two triangular elements, as shown in the following figure:



You cannot split a 5-node quadrilateral element, and you cannot split a quadrilateral-shaped gasket element. You can select multiple quadrilateral elements to split as long as they are from the same part.

1. Enter the Mesh module and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part** and select a part from the list.
2. From the main menu, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Split (quad to tri)**.
4. Select the quadrilateral elements to split. You can use any of the selection techniques described in *Selecting objects within the current viewport*.

   If your selection includes elements that are not quadrilaterals, Abaqus/CAE automatically removes them from the selection.

   Abaqus/CAE highlights the diagonal it will use to split each quadrilateral element into two triangular elements and prompts you to continue. Abaqus/CAE selects the diagonals that will create the best-shaped elements based on angle measurements. You cannot select which diagonal of the quadrilateral Abaqus/CAE uses to split the element. However, after Abaqus/CAE splits the elements, you can use the **Swap diagonal (tri)** tool to change the diagonals; for more information, see *Swapping the diagonal of a pair of adjacent triangular elements*.
5. Click **Yes** to split the selected elements.

   Abaqus/CAE prompts you to select more elements to split.

   Tip: If you make a mistake while splitting quadrilateral elements, click **Undo** in the **Edit Mesh** dialog box to undo the change.

6. When you have finished splitting elements, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.

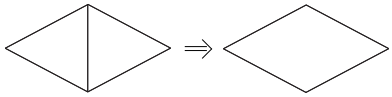
Additional information

- *What can I do with the Edit Mesh toolset?*

Combining triangular elements

The Edit Mesh toolset allows you to combine two adjacent triangular elements into a single quadrilateral element.

Select **Mesh->Edit** from the main menu bar to combine two adjacent triangular elements into a single quadrilateral element, as shown in the following figure:



You can combine either first- or second-order elements, but you cannot combine a combination of both. In addition, you can combine elements only if the adjacent triangular elements have consistent normals. If necessary, you can use the Edit Mesh toolset to flip the normals; for more information, see *Reversing the surface normal direction of shell elements*.

1. Enter the Mesh module and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part** and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.
3. In the dialog box, do the following:
   a. In the **Category** field, select **Element**.
   b. From the **Method** list, select **Combine (tri to quad)**.
4. Select the first triangular element to combine.
5. Select the second triangular element to combine. The second element must be adjacent to the first, and the second element must be the same order as the first.

   Abaqus/CAE indicates how it will combine two adjacent triangular elements into a single quadrilateral element and prompts you to continue.
6. Click **Yes** to combine the elements.

   Abaqus/CAE prompts you to select the next elements to combine.

   Tip: If you make a mistake while combining triangular elements, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the two triangular elements.
7. When you have finished combining elements, click mouse button 2 or the cancel button in the prompt area to exit the procedure.

Additional information

- *What can I do with the Edit Mesh toolset?*

Renumbering elements

The Edit Mesh toolset allows you to renumber elements.

Select **Mesh->Edit** from the main menu to renumber selected elements. You can renumber the selected elements by specifying a starting label and an increment or by offsetting the existing label by a specified value. (For information on renumbering the elements of an Abaqus/CAE native mesh, see *Changing the labels of all nodes and elements*.)

1. Enter the Mesh module.

   This tool is available only for working on parts that do not contain any geometry. You can work in the part or assembly context. However, any renumbering applied to a part instance is applied at the part level and, therefore, affects all instances of that part.

2. From the main menu, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. In the dialog box, do the following:

   a. In the **Category** field, select **Element**.

   b. From the **Method** list, select **Renumber**.

   Abaqus/CAE displays the **Element Renumbering** dialog box. If a single mesh part is displayed in the viewport, the **Part Label Range** displays the range of element labels for the entire part.

4. Do either of the following to specify the renumbering method:

   - Choose **By Start Label**, and enter the value of the first element label and the increment that will be used to number subsequent elements.
   - Choose **By Offset**, and enter the value by which the existing element labels will be offset.

5. Use one of the following methods to select the elements that you want to renumber:

Selecting all elements of a part

- Select **All** (in the part context) or **Part** (in the assembly context) to renumber all the orphan mesh elements of a part. If there is more than one mesh part instance in the viewport, click the  button to select a part instance.

Selecting elements using the unordered method

1. Select **Specify** and **Unordered** to select the elements in any sequence.
2. Click **Select** to select the elements to renumber.
3. From the prompt area, choose the selection method and select the elements to renumber. For more information, see *Selecting the elements to edit*.

Selecting elements using the directed path method

1. Select **Specify** and **Directed Path** to renumber the elements along an edge from a selected start element to a selected end element.
2. Click **Select** to select the elements to renumber.
3. Select the start element. Abaqus/CAE displays all the elements along the feature edge. If desired, modify the angle to redefine how Abaqus/CAE defines a feature edge.
4. Select **Done** from the prompt area.
5. Select the end element. The start and end elements must lie on the same feature edge.

Selecting elements using the sequence method

1. Select **Specify** and **Sequence** to renumber the elements based on the sequence in which you select the elements.
2. Click **Select** to select the elements to renumber.
3. Select the start element. Abaqus/CAE displays all the elements along the feature edge. If desired, modify the angle to redefine how Abaqus/CAE defines a feature edge.
4. Select **Done** from the prompt area. The start element is displayed in purple.
5. Select the end element. The start and end elements must lie on the same feature edge.

The **Part Label Range** displays the range of element labels for the renumbered elements.

6. Click **OK** or **Apply** to renumber the elements. If you click **Apply**, Abaqus/CAE renumbers the elements; however, the elements remain selected, and you can continue renumbering them.
7. Repeat the previous steps as often as required to renumber additional elements.
8. When you have finished renumbering elements, click **Cancel** to close the **Element Renumbering** dialog box.

Additional information

- *Changing the labels of all nodes and elements*
- *What can I do with the Edit Mesh toolset?*

Selecting the elements to edit

When you are selecting the elements to edit, Abaqus/CAE displays a menu in the prompt area that allows you to choose the technique that you will use to select the elements. The techniques allow you to select individual nodes or groups of adjacent elements.

1. Use one of the following methods to select the elements that you want to edit:

Selecting individual elements

1. From the menu in the prompt area, select **Individually**.
2. Select an element that you want to renumber, or drag-select a group of elements.
3. **[Shift] + Click** on additional elements to add them to your selection.
4. If necessary, **[Ctrl] + Click** on selected elements to remove them from your selection.

Selecting elements using the angle method

1. From the menu in the prompt area, select **by angle**.
2. Enter an angle (from 0° to 90°), and select an element. Abaqus/CAE selects every element on element faces adjacent to the selected element until the angle between the element faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects*, for more information.)
3. After you use the **by angle** method, you can **[Shift] + Click** on additional elements to add them to your selection or **[Ctrl] + Click** on selected elements to remove them from your selection. (See *Combining selection techniques*, for more information.)

Selecting elements using the feature edge method

1. From the menu in the prompt area, select **by feature edge**.
2. Enter an angle (from 0° to 90°). Abaqus/CAE identifies all the feature edges in your model by finding all the element edges where the angle between two adjacent element faces is greater than the angle specified.
3. Select an element edge. Abaqus/CAE follows the feature edge that passes through the selected element edge or node. The feature edge is truncated if another feature edge intersects it at an angle greater than the angle specified in the previous step.
4. Abaqus/CAE selects all the elements along the feature edge. (See *Using the angle and feature edge method to select multiple objects*, for more information.) Alternatively, you can select the node at the intersection of multiple feature edges, and Abaqus/CAE selects the elements along all these feature edges.
5. After you use the **by feature edge** method, you can **[Shift] + Click** on additional elements to add them to your selection or **[Ctrl] + Click** on selected elements to remove them from your selection. (See *Combining selection techniques*, for more information.)

Specifying an existing element set

1. Click **Sets** on the right side of the prompt area. Abaqus/CAE displays the **Region Selection** dialog box containing a list of element sets that you have created.

2. Select the set of elements that you want to renumber, and click **Continue**.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

2. When you have finished selecting elements, select **Done** from the prompt area.

Additional information

- *Selecting objects within the current viewport*
- *What can I do with the Edit Mesh toolset?*

Editing the entire mesh

This section describes how you can use the Edit Mesh toolset to perform a global edit of a mesh containing orphan mesh components or Abaqus/CAE native mesh components or a combination of both mesh types.

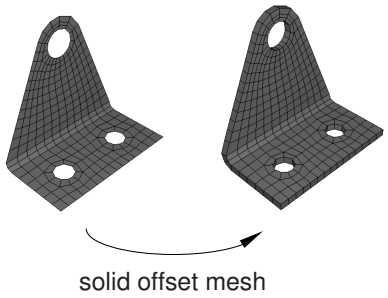
In this section:

- *Generating layers of solid elements offset from an existing mesh*
- *Generating layers of shell elements offset from an existing mesh*
- *Wrapping a mesh about the Z-axis*
- *Collapsing short edges of a linear orphan mesh*
- *Growing short edges of a linear orphan mesh*
- *Converting a triangular shell mesh to a solid tetrahedral mesh*
- *Converting a solid orphan mesh to a shell orphan mesh*
- *Merging elements*
- *Subdividing elements*
- *Copying a mesh pattern*
- *Viewing and editing mesh-geometry associativity*
- *Deleting mesh-geometry associativity*
- *Inserting cohesive seams*

Generating layers of solid elements offset from an existing mesh

The Edit Mesh toolset allows you to generate layers of solid elements that are offset normal to element faces selected from an existing mesh.

Select **Mesh->Edit** from the main menu bar to generate layers of solid elements that are offset normal to element faces selected from an existing mesh, as shown in the following figure:



1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Mesh**.
   b. From the **Method** list, select **Offset (create solid layers)**.
5. Use one of the following methods to select the element faces from which Abaqus/CAE will generate the layers of solid elements. You can select shell element faces or solid element faces, but you cannot select both. If you select shell element faces, the direction of the surface normals of adjacent shell elements must be consistent. You can reverse the surface normal direction of shell elements using the **Flip normal** tool. For more information, see *Reversing the surface normal direction of shell elements*.

Selecting individual elements

1. From the menu in the prompt area, select **Individually**.
2. Select the element faces.
3. **[Shift] + Click** additional element faces to add them to your selection.
4. If necessary, **[Ctrl] + Click** selected element faces to unselect them.
5. When you have finished selecting element faces, click mouse button 2.

Specifying an existing surface

1. Click **Surfaces** on the right side of the prompt area.

Abaqus/CAE displays the **Region Selection** dialog box containing a list of surfaces that you have created.

2. Select the surfaces, and click **Continue**. The surface must contain only quadrilateral or triangular faces.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

Selecting elements using the angle method

1. From the menu in the prompt area, select **by angle**.
2. Enter an angle (from 0° to 90°), and select an element face. Abaqus/CAE selects every adjacent element face from the selected face until the angle between the element faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects*, for more information.)
3. After you use the **by angle** method, you can **[Shift] + Click** additional element faces to add them to your selection or **[Ctrl] + Click** selected element faces to remove them from your selection. (See *Combining selection techniques*, for more information.)
4. When you have finished selecting element faces, click mouse button 2.

6. From the **Offset Mesh - Solid Layers** dialog box that appears, do the following:
   a. If you selected shell element faces from which to offset, you can choose the color that indicates the desired offset direction. The two colors from which you can choose correspond to the two sides of the selected element faces. You can also choose **Both** to indicate that Abaqus/CAE should create layers of solid elements on both sides of the selected shell elements.
   b. Enter the **Total thickness** of all the layers combined. The value that you enter must be positive. However, you can enter a value of 0 if you are creating zero-thickness elements that will be used in a cohesive analysis.
   c. Enter the **Number of layers**.
   d. If desired, toggle on **Specify initial offset**, and enter a value for the initial offset. A negative initial offset indicates overclosure between elements.
   e. If the surface from which you are offsetting contains sharp corners, toggle on **Constant thickness around corners**. You should also toggle this option on if you anticipate that the generated offset elements may collapse or become inverted. For more information, see *Reducing element distortion and collapse during mesh offsetting*.
   f. To share the nodes of the first layer of solid elements with the nodes of the selected element faces, toggle on **Share nodes with base shell/surface**. This option is available only if the first layer of solid elements is not offset from the selected element faces.
   g. To delete the selected shell elements after creating the layers of solid elements, toggle on **Delete base shell elements**. This option is available only if you selected orphan shell element faces from which to offset.
   h. If you want to create a single set containing all of the elements from the solid offset mesh, toggle on **Create a set for new elements**.
   i. If you want to create separate sets containing the elements from each layer of the solid offset mesh, toggle on **Separate set for each layer**. Abaqus/CAE appends **Layer-*layer number*** to the set name prefix when it creates separate sets for each layer.

j. If all or part of the solid offset mesh will not be embedded in the existing mesh, you can create surfaces from the top and bottom surfaces of the offset mesh. by toggling on **Create top and bottom surfaces**. Abaqus/CAE appends **BottomSurf** and **TopSurf** to the name prefix when it creates the two surfaces.

k. If desired, enter a new name for the single set or a new prefix for the separate sets and/or surfaces.

7. Abaqus/CAE creates the layers of solid elements offset normal to the selected element faces. When you have finished generating layers of solid elements, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.



Tip: If you make a mistake while generating layers of elements, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the mesh to its state just before the operation.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

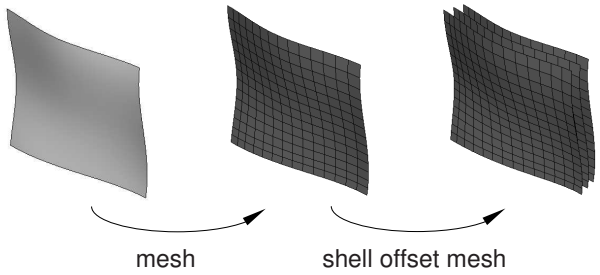
Additional information

- *Using offset meshes to create skin reinforcements*
- *Adhesive joints and bonded interfaces*
- *Using offset meshes in Abaqus/CAE*
- *Reducing element distortion and collapse during mesh offsetting*
- *Allowing for branching in offset solid meshes*
- *Creating a mid-plane shell mesh from a thin solid model*

Generating layers of shell elements offset from an existing mesh

The Edit Mesh toolset allows you to generate layers of shell elements that are offset normal to element faces selected from an existing mesh.

Select **Mesh->Edit** from the main menu bar to generate layers of shell elements that are offset normal to element faces selected from an existing mesh, as shown in the following figure:



1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, do the following:
   a. In the **Category** field, select **Mesh**.
   b. From the **Method** list, select **Offset (create shell layers)**.
5. Use one of the following methods to select the element faces from which Abaqus/CAE will generate the layers of shell elements. You can select shell element faces or solid element faces, but you cannot select both. If you select shell element faces, the direction of the surface normals of adjacent shell elements must be consistent. You can reverse the surface normal direction of shell elements using the **Flip normal** tool. For more information, see *Reversing the surface normal direction of shell elements*.

Selecting individual elements

1. From the menu in the prompt area, select **Individually**.
2. Select the element faces.
3. **[Shift] + Click** additional element faces to add them to your selection.
4. If necessary, **[Ctrl] + Click** selected element faces to unselect them.
5. When you have finished selecting element faces, click mouse button 2.

Specifying an existing surface

1. Click **Surfaces** on the right side of the prompt area.

Abaqus/CAE displays the **Region Selection** dialog box containing a list of surfaces that you have created.

2. Select the surfaces, and click **Continue**. The surface must contain only quadrilateral or triangular faces.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

Selecting elements using the angle method

1. From the menu in the prompt area, select **by angle**.
2. Enter an angle (from 0° to 90°), and select an element face. Abaqus/CAE selects every adjacent element face from the selected face until the angle between the element faces is equal to or exceeds the angle that you entered. (See *Using the angle and feature edge method to select multiple objects*, for more information.)
3. After you use the **by angle** method, you can **[Shift] + Click** additional element faces to add them to your selection or **[Ctrl] + Click** selected element faces to remove them from your selection. (See *Combining selection techniques*, for more information.)
4. When you have finished selecting element faces, click mouse button 2.

6. From the **Offset Mesh - Shell Layers** dialog box that appears, do the following:
   a. If you selected shell element faces from which to offset, you can choose the color that indicates the desired offset direction. The two colors from which you can choose correspond to the two sides of the selected element faces. You can also choose **Both** to indicate that Abaqus/CAE should create layers of shell elements on both sides of the selected shell elements.
   b. Enter the **Distance between layers**. You can enter a value of 0; for example, if you want to create multiple layers of skin reinforcements that share nodes.
   c. Enter the **Number of layers**.
   d. If desired, toggle on **Specify initial offset**, and enter a value for the initial offset. A negative initial offset indicates overclosure between elements.
   e. If the surface from which you are offsetting contains sharp corners, toggle on **Constant thickness around corners**. You should also toggle this option on if you anticipate that the generated offset elements may collapse or become inverted. For more information, see *Reducing element distortion and collapse during mesh offsetting*.
   f. To share the nodes of the first layer of shell elements with the nodes of the selected element faces, toggle on **Share nodes with base shell/surface**. This option is available only if the first layer of shell elements is not offset from the selected element faces.
   g. To delete the selected shell elements after creating the layers of shell elements, toggle on **Delete base shell elements**. This option is available only if you selected orphan shell element faces from which to offset.
   h. If you want to create a single set containing all of the elements from the shell offset mesh, toggle on **Create a set for new elements**.
   i. If you want to create separate sets containing the elements from each layer of the shell offset mesh, toggle on **Separate set for each layer**. Abaqus/CAE appends **Layer-*layer number*** to the set name prefix when it creates separate sets for each layer.
   j. If desired, enter a new name for the single set or a new prefix for the separate sets.

7. Abaqus/CAE creates the layers of shell elements offset normal to the selected element faces. When you have finished generating layers of shell elements, click mouse button 2 or the cancel button  in the prompt area to exit the procedure.



Tip: If you make a mistake while generating layers of elements, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the mesh to its state just before the operation.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *Using offset meshes to create skin reinforcements*
- *Adhesive joints and bonded interfaces*
- *Using offset meshes in Abaqus/CAE*
- *Reducing element distortion and collapse during mesh offsetting*

Wrapping a mesh about the Z-axis

Use the Abaqus Scripting Interface method `wrapMesh` to wrap a planar orphan mesh around a cylindrical surface defined by the global *Z*-axis and a specified radius.
The planar mesh must lie in the *X*–*Y* plane, and the modeling space of the part should be three-dimensional. If necessary, you can change the modeling space of a part by clicking mouse button 3 on the part in the Model Tree and selecting **Edit** from the menu that appears.

The wrapping procedure will relocate a node at point (x, y) on the planar mesh to (r, θ, z), where r is the specified radius, $\theta = \frac{x}{r}$, and $z=y$, as shown in *Figure 1*. A point on the planar orphan mesh located at (0, 0) is mapped to location $(r, 0, 0)$ on the cylindrical surface.

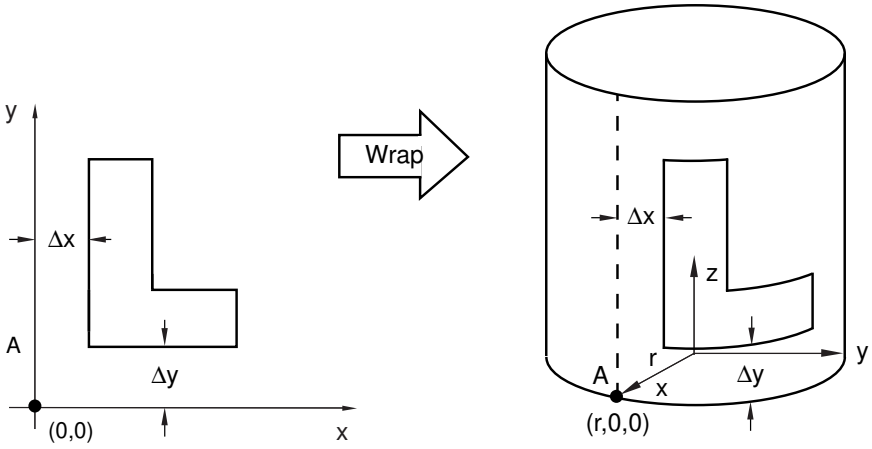


Figure 1: Wrapping a planar orphan mesh.

Because only part instances can be positioned and only parts can be wrapped, if you want to create an instance of an orphan mesh that is wrapped and oriented properly in the assembly, some planning is required. The following steps will help you achieve the desired goal:

1. Use the tools in the Assembly module to translate and/or rotate an instance of the planar orphan mesh in the assembly.
2. Use the **Create Mesh Part** tool in the Mesh module to create a planar mesh part from the reoriented instance.
3. Wrap the reoriented mesh part.
4. Create an instance of the wrapped part, and use the tools in the Assembly module to translate and/or rotate the instance in the assembly.

If you are trying to create a full 360° wrap, discretization approximations may result in a small gap between nodes along the seam edge. You must use the merge nodes tool to stitch the seam by merging pairs of nodes into a single node; for more information, see *Merging nodes*.

The following shows an example of using the Abaqus Scripting Interface to wrap a mesh:

```
model = mdb.models['Model-1']
part = model.parts['Part-1']
part.wrapMesh(radius=5.0)
```

You can use the command line interface to enter Abaqus Scripting Interface commands, as described in *Using the Python interpreter*.

Additional information

- *What can I do with the Edit Mesh toolset?*

• *Edit mesh commands*

Collapsing short edges of a linear orphan mesh

Select **Mesh->Edit** from the main menu bar to collapse selected edges of a linear orphan mesh. You can specify the minimum length of the edges of elements that should be collapsed, and Abaqus/CAE will merge the end nodes of element edges that are shorter than the specified length. You can specify an element domain other than the entire mesh, and you can specify a direction and/or reference edge to collapse only those edges that match these criteria. You can collapse the mesh only if the element domain contains only linear elements.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Collapse short edges**.

   Abaqus/CAE displays the **Collapse short edges** dialog box.
5. Enter the **Edge tolerance** to specify the minimum length of the edges of elements indicating that you want to collapse element edges that are shorter than the tolerance. The default value is approximately 10% of the average element size in the part.
6. If desired, toggle on **Specify** and click **Select** to choose an element domain other than the entire mesh.
7. If desired, toggle on **Thickness direction** and enter a direction vector or click the arrow icon to create a vector by picking points in the viewport.

   Abaqus/CAE uses the direction vector to measure the elements instead of measuring only the distance between nodes. When you use a direction vector, the collapse operation must be limited to specific edges of the selected elements to avoid collapsing edges that are perpendicular to the intended direction. If elements are oriented such that the direction is topologically ambiguous, Abaqus/CAE highlights the elements so you can either select a domain that excludes them or add a reference edge to further define the desired edges.
8. If desired, toggle on **Reference edge** and click **Select** to choose a reference edge.

   When you use a reference edge, Abaqus/CAE checks for short edges in only those element edges that are topologically parallel to the selected edge.

   Note:

   If your element domain includes unconnected element sections, you can select a reference edge only within a single contiguous part of the domain. You may need to use the **Collapse short edges** tool multiple times on smaller element domains to achieve the desired results.
9. Click **OK**.

   Abaqus/CAE highlights the elements that will have edges collapsed. Click **Yes** in the prompt area to collapse the edges or **No** to return to the **Collapse short edges** dialog box.

   If none of the element edges are shorter than the minimum length specified, Abaqus/CAE displays an error message that indicates the shortest element length.



Tip: If you make a mistake while collapsing short edges in a mesh, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original mesh.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Growing short edges of a linear orphan mesh

Select **Mesh->Edit** from the main menu bar to grow selected edges of a linear orphan mesh. You can specify the minimum length of the edges of elements that should be grown, and Abaqus/CAE will move the end nodes of element edges that are shorter than the specified length. You can specify an element domain other than the entire mesh, and you can specify a direction and/or reference edge to grow only those edges that match these criteria. You can grow edges in the mesh only if the element domain contains only linear elements.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Grow short edges**.

   Abaqus/CAE displays the **Grow short edges** dialog box.
5. Enter the **Edge tolerance** to specify the minimum length of the edges of elements indicating that you want to grow element edges that are shorter than the tolerance. The default value is approximately 10% of the average element size in the part.
6. If desired, toggle on **Specify** and click **Select** to choose an element domain other than the entire mesh.

   Note:

   Abaqus/CAE does not check edges outside of the specified domain at all. If there are short element edges adjacent to the selected domain and an edge within the domain grows, the repositioned nodes could make an element outside the domain too small or possibly invert an element.

7. If desired, toggle on **Thickness direction** and enter a direction vector or click the arrow icon to create a vector by picking points in the viewport.

   Abaqus/CAE uses the direction vector to measure the elements instead of measuring only the distance between nodes. When you use a direction vector, the grow operation must be limited to specific edges of the selected elements to avoid growing edges that are perpendicular to the intended direction. If elements are oriented such that the direction is topologically ambiguous, Abaqus/CAE highlights the elements so you can either select a domain that excludes them or add a reference edge to further define the desired edges.
8. If desired, toggle on **Reference edge** and click **Select** to choose a reference edge.

   When you use a reference edge, Abaqus/CAE checks for short edges in only those element edges that are topologically parallel to the selected edge.

   Note:

   If your element domain includes unconnected element sections, you can select a reference edge only within a single contiguous part of the domain. You may need to use the **Grow short edges** tool multiple times on smaller element domains to achieve the desired results.

9. Click **OK**.

   Abaqus/CAE highlights the elements that will have edges grown. Click **Yes** in the prompt area to grow the edges or **No** to return to the **Grow short edges** dialog box.

   If none of the element edges are shorter than the minimum length specified, Abaqus/CAE displays an error message that indicates the shortest element length.

   Tip: If you make a mistake while collapsing short edges in a mesh, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original mesh.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Converting a triangular shell mesh to a solid tetrahedral mesh

Select **Mesh->Edit** from the main menu bar to convert a closed three-dimensional shell of triangular elements into a solid mesh of tetrahedral elements. The original shell part must be meshed with triangular elements that fully enclose a volume with no gaps.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list. This tool is available only for working on orphan mesh elements.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

4. From the **Category** field, choose **Mesh**.
5. From the **Method** list, select **Convert tri to tet**.
6. From the buttons that appear in the prompt area, click **Yes**.

   Abaqus/CAE converts the triangular mesh to a tetrahedral mesh.

   Tip: You can recover the triangular mesh in its state just before the conversion by clicking **Undo** in the **Edit Mesh** dialog box.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Converting a solid orphan mesh to a shell orphan mesh

Select **Mesh->Edit** from the main menu bar to convert a solid orphan mesh to a shell orphan mesh. The **Convert solid to shell** tool creates triangular or quadrilateral shell elements on the free element faces of the part; that is, the element faces that do not share all of their nodes with another face. If your part contains joined faces with elements of different element shapes or with elements of a different geometric order, Abaqus/CAE creates shell elements on both sides of the interface. Likewise, if your solid mesh part contains neighboring element faces that reference different coincident nodes (like the nodes that represent cracks), Abaqus/CAE creates shell elements on both sides of the interface.

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part from the list. This tool is available only for working on orphan mesh elements.
3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. From the **Category** field, choose **Mesh**.
5. From the **Method** list, select **Convert solid to shell**.
6. From the buttons that appear in the prompt area, click **Yes**.

   Abaqus/CAE converts the solid mesh to a shell mesh.

   Tip: You can recover the solid mesh in its state just before the conversion by clicking **Undo** in the **Edit Mesh** dialog box.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Merging elements

Select **Mesh->Edit** from the main menu bar to merge adjacent orphan mesh elements. The merged elements have the same element type and basic shape as the originals. You can merge elements in the mesh only if the element domain contains only linear elements.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part that contains orphan mesh elements from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Merge layers**.
5. Select the elements that you want to merge. You can use any of the following methods:

   - Select elements from the viewport. You can select elements individually, by angle, by feature edge, or by topology (for more information, see *Selecting objects within the current viewport*). Click mouse button 2 when you have finished making selections.
   - Select one or more existing element sets. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets. Select one or more sets from the list, and click **Continue**.

   Elements being merged must be adjacent to each other.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

6. Select an element edge to indicate the merge direction.

   Abaqus/CAE attempts to merge the elements. If merging the elements will create incompatibilities in the mesh, Abaqus/CAE highlights the nodes at which the incompatibilities will occur and asks if you want to continue. Click **Yes** to merge the elements or **No** to clear your selections and return to Step 4.

   Tip: If you make a mistake while merging elements in a mesh, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original mesh.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Subdividing elements

Select **Mesh->Edit** from the main menu bar to subdivide one or more selected orphan mesh elements. The elements to be subdivided must be connected. You can select an edge, a face (solid elements), or **All Directions** to divide elements in one, two, or all directions—two for shell elements, three for solid elements. You must also specify the number of divisions to make. Abaqus/CAE divides the elements evenly into the specified number of divisions in each direction. The new elements have the same element type as the originals. You can subdivide elements in the mesh only if the element domain contains only linear elements.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part that contains orphan mesh elements from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Subdivide layers**.
5. Select the elements that you want to subdivide. You can use any of the following methods:

   - Select elements from the viewport. You can select elements individually, by angle, by feature edge, or by topology (for more information, see *Selecting objects within the current viewport*). Click mouse button 2 when you have finished making selections.
   - Select one or more existing element sets. Click **Sets** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets. Select one or more sets from the list, and click **Continue**.

   The selected elements must form a single connected set.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

6. Select an element edge to indicate a single direction of subdivision; select the face of a solid element to indicate two directions; select **All Directions** in the prompt area to indicate all directions—two for shell elements or three for solid elements.

   Abaqus/CAE attempts to subdivide the elements. If dividing the elements will create incompatibilities in the mesh, Abaqus/CAE displays a dialog box indicating that the resulting mesh will be nonconforming. Click **Yes** to divide the elements or **No** to clear your selections and return to Step 4.

   Tip: If you make a mistake while dividing elements in a mesh, click **Undo** in the **Edit Mesh** dialog box to undo the change. Abaqus/CAE restores the original mesh.

After editing an orphan mesh, you should always verify node-, element-, and surface-based features such as section assignments, loads, and interactions to ensure that they are correctly applied to the modified mesh.

Additional information

- *What can I do with the Edit Mesh toolset?*

Copying a mesh pattern

Select **Mesh->Edit** from the main menu bar to copy a two-dimensional mesh pattern onto a target face. The elements of the pattern must form a connected set that can cover a geometric face. The target face must be a portion of the same assembly or part as the pattern, and the target face must contain at least as many loops as the pattern.

1. Enter the Mesh module, and do one of the following:
   - From the **Object** field in the context bar, select **Assembly**.
   - From the **Object** field in the context bar, select **Part**, and select a part from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.
3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Copy mesh pattern**.
5. Select the source mesh. You can use any of the following methods:
   - Select elements from the viewport. You can select elements individually, by face angle, or by face curvature (for more information, see *Selecting objects within the current viewport*). Click mouse button 2 when you have finished making selections.
   - Select an existing surface. Click **Surfaces** on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available surfaces. Select a surface from the list, and click **Continue**.

   The selected elements must form a single connected two-dimensional face.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.
6. Select the target face.

   The target face must be part of the same assembly or part as the source, and the face must contain at least as many loops as the source.
7. In the **Copy mesh pattern** dialog box, associate node positions on the source mesh with points on the target face.

   You should associate at least three nodes from the outer edge (primary loop) of the source mesh with points on the target geometry.

   a. Click  in the **Copy mesh pattern** dialog box to add nodes and points to the table.
   b. Select the desired nodes from the source mesh, and click **Done** in the prompt area.
   c. Select the corresponding points on the target face.

      Abaqus/CAE highlights the current node in pink as you make selections. When you have selected a point for each node, the **Copy mesh pattern** dialog box appears with the selections displayed in the table.

d. Highlight a position in the table, and click  to edit the position.



Tip: You cannot edit nodes. To change a node selection use  to delete the highlighted row, then add a new node and position to replace it.

e. Click **OK** in the dialog box

Abaqus/CAE maps the remaining nodes onto the target face and displays the copied mesh pattern. If there are problems mapping the remaining nodes to the target, Abaqus/CAE displays error messages to help you resolve them.



Tip: If the copied mesh pattern does not meet your expectations, click **Undo** in the **Edit Mesh** dialog box to remove it.

Additional information

• *What can I do with the Edit Mesh toolset?*

Viewing and editing mesh-geometry associativity

You can view or edit the association between orphan or bottom-up mesh elements and the underlying geometry by selecting **Mesh->Associate Mesh with Geometry** from the main menu bar and choosing a vertex, edge, or face that bounds a bottom-up region or a group of orphan elements. Elements generated by a top-down technique are associated automatically with the region geometry; you cannot view or edit the associativity between these native mesh elements and the corresponding region geometry.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part that contains orphan mesh or bottom-up elements from the list.
2. From the main menu bar, select **Mesh->Associate Mesh with Geometry**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the  tool in the Mesh module toolbox, or you can click the  tool to display the **Edit Mesh** dialog box—the mesh association method is located in the **Mesh** category within the dialog box. (For more information on the Mesh module toolbox, see *Using the Mesh module toolbox*.)

3. Select a geometric entity—a vertex, edge, or face—whose associativity you want to view or edit.

   Abaqus/CAE highlights in yellow any nodes, element edges, or element faces that are associated with the selected geometry. If nothing is highlighted in yellow, no part of the mesh is associated with the selected entity.
4. Use the following selection techniques to change the mesh-geometry associativity (for more information, see *Selecting objects within the viewport*"):
   a. Click a node, element edge, or element face to cancel any existing selections and begin selecting new items.
   b. **[Shift] + Click** to add nodes, element edges, or element faces to the current selection, or **[Ctrl] + Click** to unselect items.
   c. Use the **by angle** selection method available in the prompt area to select element faces based on the angle between them.
   d. You can use the tools in the **Selection** toolbar to limit the types of objects that you can select in the viewport.

   Note:

   In some cases you may need to pick interior mesh faces or edges to create the correct mesh-geometry association. To allow you to select interior objects, Abaqus/CAE does not limit your associativity selections to the "Objects closest to the screen" even when this selection option is active (see *Filtering your selection based on the position of the object*, for more information). As a result, you need to ensure that you do not inadvertently associate unwanted elements, especially if you use drag-select to select multiple objects.
5. When you have finished selecting mesh objects to associate, click **Done**.

   Abaqus/CAE saves the new associativity.

6. Repeat Steps 3 through 5 to verify or change the association of other areas, or click the cancel button  in the prompt area to end the associativity procedure.

Additional information

- *Selecting objects within the viewport*
- *Bottom-up meshing*
- *Mesh-geometry association*
- *Using the Mesh module toolbox*
- *What can I do with the Edit Mesh toolset?*

Deleting mesh-geometry associativity

You can delete the association between nodes, element edges, or elements and the underlying geometry by selecting **Mesh->Delete Mesh Associativity** from the main menu bar and choosing a vertex, edge, or face. If you delete the associativity between Abaqus/CAE geometry and the mesh, the affected mesh components become orphan mesh nodes and elements. This may be useful if you are working with a complex part and want to keep some portions of the mesh unchanged while changing the geometry in other areas.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part from the list.
2. From the main menu bar, select **Mesh->Delete Mesh Associativity**.

   Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also click the  tool in the Mesh module toolbox, or you can click the  tool to display the **Edit Mesh** dialog box—the delete associativity method is located in the **Mesh** category within the dialog box. (For more information on the Mesh module toolbox, see *Using the Mesh module toolbox*.)

3. Select one or more geometric entities—vertices, edges, faces, or entire regions—whose associativity you want to delete.

   You can use a combination of drag select, **[Shift] + Click**, and **[Ctrl] + Click** to select geometric entities. For more information, see *Selecting objects within the current viewport*.

   Note:

   To delete the association of a native mesh region and create an orphan mesh, you must select the entire region.

4. If desired, you can toggle off **Delete association from bounding entities** in the prompt area.

   By default, when you delete the associativity of a region, Abaqus/CAE also deletes the association between nodes, element edges, and faces and the corresponding vertices, edges, and faces.

5. When you have finished selecting geometric entities, click **Done**.

   Abaqus/CAE highlights the geometric entities in red and the associated mesh entities in yellow so you can confirm your selections.

6. Click **Yes** to delete the mesh association or **No** to cancel your selections.

7. Repeat Steps 3 through 6 to delete the associativity of other areas, or click the cancel button  in the prompt area to end the procedure.

Additional information

- *Selecting objects within the viewport*
- *Bottom-up meshing*
- *Mesh-geometry association*
- *Using the Mesh module toolbox*
- *What can I do with the Edit Mesh toolset?*

Inserting cohesive seams

Select **Mesh->Edit** from the main menu bar to insert pore pressure cohesive elements in crack regions, which allows you to model fluid flow into the adjacent material. You can select faces (solid cells), element faces (three-dimensional orphan meshes), edges (shells), or element edges (two-dimensional orphan meshes) to identify the crack regions. The region around the crack must be meshed prior to inserting cohesive seams. By default, Abaqus/CAE creates several sets and surfaces to help you make selections in subsequent procedures, such as section assignment.

1. Enter the Mesh module. Select **Part** from the **Object** field in the context bar, and select a part that contains the crack region from the list.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. From the **Category** field, choose **Mesh**.
4. From the **Method** list, select **Insert cohesive seams**.
5. Select the faces (for three-dimensional objects) or edges (for two-dimensional objects) that represent crack regions. You can use any of the following methods:

   - Select faces or edges from the viewport. You can select faces individually, by angle, or, for native meshes, by face curvature. You can select edges individually or by edge angle (for more information, see *Selecting objects within the current viewport*). Click mouse button 2 when you have finished making selections.
   - Click **Sets/Surfaces** (for native meshes) or **Surfaces** (for orphan meshes) on the right side of the prompt area to display the **Region Selection** dialog box containing a list of available sets or surfaces. Select one or more sets or surfaces from the list, and click **Continue**.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets/Surfaces** (or **Surfaces**) on the right side of the prompt area.

6. By default, Abaqus/CAE creates several sets and surfaces. Click **Options** on the right side of the prompt area to select the sets or surfaces to be created and, if desired, to change the set or surface names from the default values.
7. From the buttons along the bottom of the **Options** dialog box, do the following:

   - Click **OK** to save your settings and to close the dialog box.
   - Click **Defaults** to reset the set and surface selections and names to the default values.
   - Click **Clear** to clear all of the set and surface selections and names.
   - Click **Cancel** to close the dialog box.

8. Click **Done** in the prompt area.

Abaqus/CAE inserts cohesive pore pressure elements and creates the specified sets and surfaces.

Additional information

• *What can I do with the Edit Mesh toolset?*

Editing and refining an orphan mesh

You can use the Edit Mesh toolset to refine an orphan mesh.
You can refine a planar, triangular mesh with or without a specified global element size.

Select **Mesh->Edit** from the main menu bar to refine a planar, triangular mesh. You can remesh the part using the following methods:

With a specified global element size

Before you remesh the part, you have the option of assigning a target element size to the entire part. You can then remesh the part, and the density of the new mesh reflects the new target element size. For example, when the part in *Figure 3* is remeshed with a global element size of 15.0, *Figure 1* shows the resulting mesh.

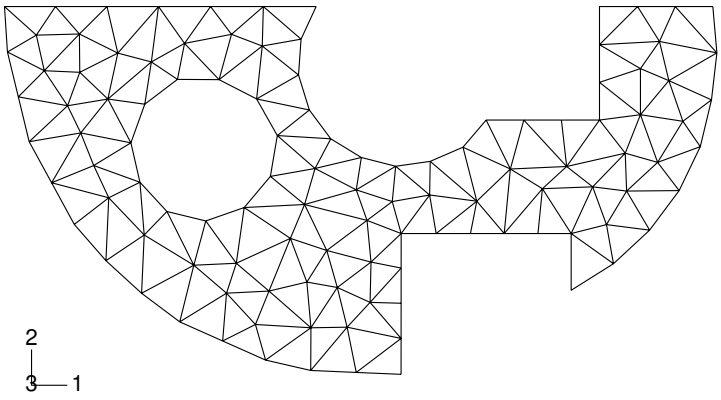


Figure 1: A global element size of 15.0.

Figure 2 shows the part remeshed with a global element size of 8.0.

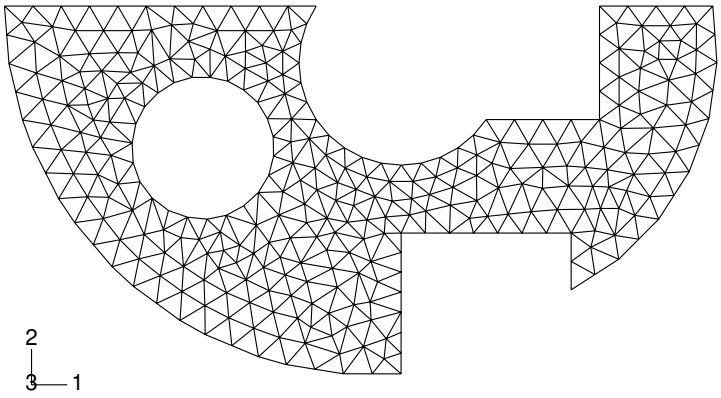


Figure 2: A global element size of 8.0.

Without a specified global element size

If no global element size is specified, Abaqus/CAE maintains the edges of the elements along the boundary of the part while improving the mesh quality in the interior of the part. The resulting mesh topology is different from the original mesh topology. For example, *Figure 3* shows a distorted mesh.

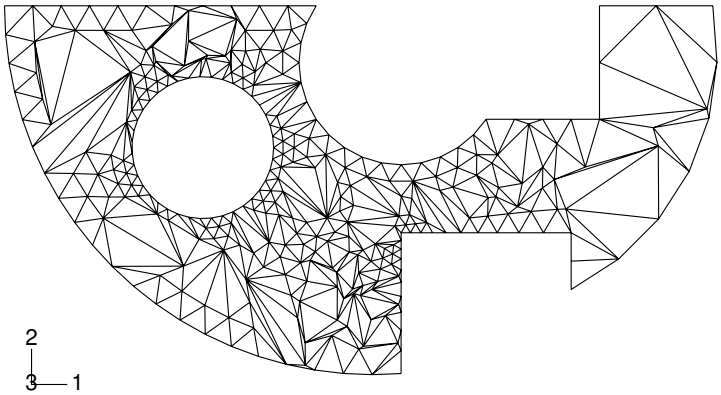


Figure 3: A distorted mesh.

When the part is remeshed, the quality of the mesh improves dramatically, as shown in *Figure 4*.

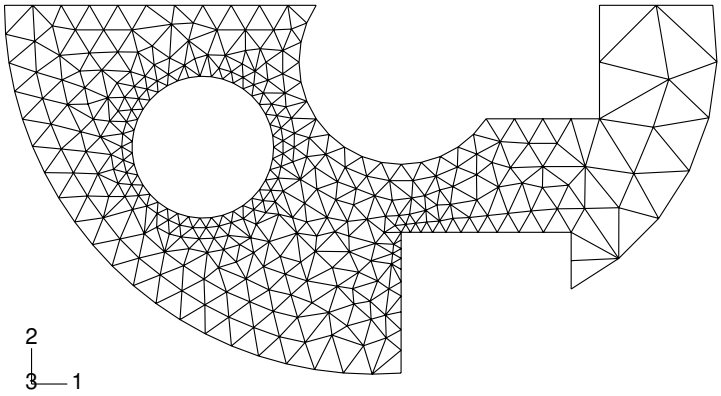


Figure 4: The part is remeshed without specifying a global element size.

Additional information

- *What can I do with the Edit Mesh toolset?*

Refine a planar, triangular mesh with a specified global element size

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part that contains only orphan mesh elements from the list.

   Tip: To refine the orphan elements within a hybrid part, suppress the geometric features and resume them after completing the refinement operation.

3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, choose **Refinement** from the **Category** field.
5. From the **Method** list, select **Set size**.
6. In the prompt area, type the global element size of your choice, and press **[Enter]**.

   Abaqus/CAE displays a circle that indicates what the size of the elements will be after you remesh the part.

7. From the **Method** list, select **Remesh**.
8. From the buttons that appear in the prompt area, click **Yes**.

   Abaqus/CAE attempts to refine the mesh. If you make a mistake while refining the mesh, click **Undo** in the **Edit Mesh** dialog box to undo the refinement.

Refine a planar, triangular mesh without a specified global element size

1. Enter the Mesh module.
2. From the **Object** field in the context bar, select **Part** and select a part that contains orphan mesh elements from the list.

   Tip: To refine the orphan elements within a hybrid part, suppress the geometric features and resume them after completing the refinement operation.

3. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

4. In the dialog box, choose **Refinement** from the **Category** field.
5. From the **Method** list, select **Remove size**.
6. From the **Method** list, select **Remesh**.
7. From the buttons that appear in the prompt area, click **Yes**.

Abaqus/CAE attempts to refine the mesh. If you make a mistake while refining the mesh, click **Undo** in the **Edit Mesh** dialog box to undo the refinement.

Undoing or redoing a change in the Edit Mesh toolset

This section describes how you can undo or redo mesh changes that you make with the tools in the Edit Mesh toolset. This section also explains how you can configure the cache that stores the rollback information for your Edit Mesh operations.

These topics also apply to undoing and redoing bottom-up meshing operations (for more information on bottom-up meshing, see *Bottom-up meshing*).

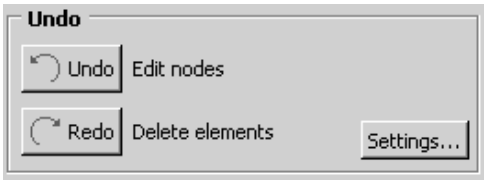
In this section:

- *Support for undo and redo in the Edit Mesh toolset*
- *Undoing or redoing mesh editing actions*
- *Enabling undo and managing the cache*

Support for undo and redo in the Edit Mesh toolset

Abaqus/CAE supports multiple levels of undo and redo for any of the tools in the Edit Mesh toolset and for meshes that you generate using the bottom-up meshing technique.

This capability means that a single undo removes your most recent change to the mesh, a second undo removes the next most recent change, and so on. Similarly, a single redo restores the most recently undone mesh change. The graphic below shows that the **Undo** section of the **Edit Mesh** dialog box includes short descriptions next to the **Undo** and **Redo** buttons, highlighting the mesh change that will be undone or redone if you click the button.



When undo is disabled, a note appears in the description space instead, indicating that undo is disabled under **Settings**. To re-enable undo, see *Enabling undo and managing the cache*.

Abaqus/CAE supports undo operations by recording a history of your mesh editing and bottom-up meshing operations in a memory cache. The allowable size of this cache, the size of your mesh, and the types of mesh editing you performed all determine how many levels of undo are permitted. For example, if your mesh has 100,000 elements and you define a maximum cache size of 0.5 million elements, Abaqus/CAE can provide at least five levels of undo. Operations such as creating or moving nodes do not require a large cache to undo.

Tip: You can determine the number of elements in your mesh by running a **Part mesh** or **Instance mesh** query. For more information, see *Querying the model*.

Large caches can consume a lot of memory, so consider the number of levels of undo you will need for your part or part instance. If you want to ensure at least one level of undo for any mesh editing operation, specify a cache size at least as large as the number of elements in your part or part instance.

Undoing or redoing mesh editing actions

The undo and redo options that appear in the **Undo** section of the **Edit Mesh** and **Create Bottom-Up Mesh** dialog boxes are specific to the selected part, part instance, or assembly. When you switch to a different item and edit its mesh, the undo options change to reflect the history of the selected item. Abaqus/CAE updates the state of the buttons and their descriptions to reflect the history of the newly selected part, part instance, or assembly.

1. Enter the Mesh module.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the tool, located at the bottom of the Mesh module toolbox.

   The **Undo** section of this dialog box includes the **Undo** and **Redo** buttons and next to each button descriptions of the mesh editing action that will be undone or redone when you click it.
3. Click **Undo** to undo the indicated mesh editing operation or click **Redo** to redo the indicated mesh editing operation.

   Abaqus/CAE performs the indicated change.
4. Continue editing the mesh or undoing or redoing changes; close the **Edit Mesh** dialog box when you are done.

Note:

When you perform a new mesh editing operation, or any other operation that alters the mesh , any redo information about the affected part or part instance will be lost. Undo and redo information is also deleted if you delete a region mesh, apply top-down meshing, or add partitions if the added partitions require part of the mesh to be deleted.

Additional information

- *Bottom-up meshing*
- *What can I do with the Edit Mesh toolset?*
- *Enabling undo and managing the cache*

Enabling undo and managing the cache

Enabling undo is global and persistent; when this setting is enabled, you will be able to undo mesh editing changes for any part or assembly meshes during your session. The setting remains enabled during your next Abaqus/CAE session as well. Disabling undo frees any memory associated with this setting and causes any existing undo or redo cache on any parts or assemblies to be lost.

1. Enter the Mesh module.
2. From the main menu bar, select **Mesh->Edit**.

   Abaqus/CAE displays the **Edit Mesh** dialog box.

   Tip: You can also display the **Edit Mesh** dialog box using the  tool, located at the bottom of the Mesh module toolbox.

3. Click **Settings**.

   The **Mesh Edit Undo Settings** dialog box appears.
4. Toggle on **Enable Undo** to enable undo for mesh editing operations.
5. Enter a value in the **Max total cache size** field in units of millions of elements, and click **OK**.

Additional information

- *Bottom-up meshing*
- *What can I do with the Edit Mesh toolset?*
- *Undoing or redoing mesh editing actions*

The Feature Manipulation toolset

You build a model in Abaqus/CAE by creating a series of features.
For detailed discussions of features and feature-based modeling, see *The relationship between parts and features* and *Manipulating features in the Assembly module*. This chapter explains how to use the Feature Manipulation toolset to modify and manage the existing features in your model and how to tune the performance of feature regeneration.

In this section:

- *Using the Feature Manipulation toolset*
- *Using the Model Tree to manage features*
- *Tuning feature regeneration*
- *Modifying and manipulating features*

Using the Feature Manipulation toolset

You can access the Feature Manipulation toolset by selecting **Feature** from the main menu bar or by clicking mouse button 3 on a feature in the Model Tree. You can also access the feature tools from the module toolboxes. *Figure 1* shows the icons for all the feature tools in the module toolboxes.

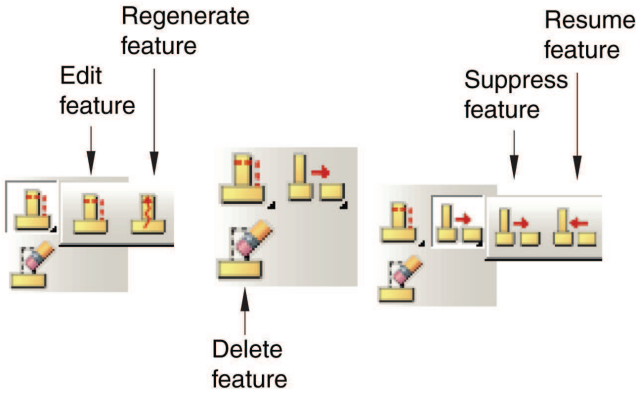


Figure 1: The feature tools.

The following list describes how you can use the **Edit**, **Regenerate**, **Suppress**, **Resume**, and **Delete** tools to modify features or to control the appearance of features in a part or assembly.

Edit a feature

You can edit the features of a part only in the Part module. Any changes that you make to a part are applied automatically to each instance of that part in the assembly. Likewise, you can edit features of the assembly only in the Assembly and Mesh modules. For information on restrictions that apply to editing partitions and datum geometry, see *Understanding a datum as a feature*, and *Understanding partitions*.

Regenerate a feature

Regeneration is the process of recalculating model geometry after a feature of the model has been modified. By default, Abaqus/CAE automatically regenerates a part or assembly after you have edited one of its features. However, you can control whether or not features are regenerated automatically by toggling the **Regenerate on OK** option off or on in the **Feature Editor**. For more information, see *Editing a feature*.

Suppress a feature

When you are manipulating many features in a complex model (for example, when you are exploring design alternatives), Abaqus/CAE allows you to temporarily disable certain features to simplify the display or to speed regeneration. A suppressed feature is not visible and cannot be partitioned or meshed. In addition, Abaqus/CAE ignores all suppressed features when regenerating a part or assembly. You can edit a suppressed feature, although your changes will have no effect on the part or assembly until you resume the feature. Suppressing a feature also suppresses any children of that feature. (For information on parent and child features, see *The relationship between parts and features*.)

Abaqus/CAE does not include a suppressed feature in the input file that it generates when you submit a job for analysis. However, Abaqus/CAE does include in the input file prescribed conditions that refer to the suppressed feature or regions of the suppressed feature. To ensure that the analysis completes successfully, you should do one of the following:

- Resume the suppressed feature.
- Edit the prescribed condition, and apply the prescribed condition to a different region.
- Delete the prescribed condition.

If you want to temporarily make a part instance invisible in the assembly, you can hide the part instance by selecting options in the **Assembly Display Options** dialog box. For more information, see *Controlling instance visibility*.



Resume a feature

When you resume a suppressed feature, it reappears in the display of the part or assembly and becomes reestablished in the model. If you edited the feature when it was suppressed, resuming the feature causes Abaqus/CAE to regenerate the model automatically to take into account your changes. When you resume a feature, you have the option of resuming all of its children as well. (You cannot resume a child feature without also resuming its parent.)

Tuning regeneration performance

Tuning feature regeneration performance is a balance between the convenience of saved states and the effect on performance of memory consumption. In most cases the default settings will result in acceptable regeneration performance; however, you can tune the speed of regeneration by selecting **Feature->Options** from the main menu bar. For more information, see *Tuning feature regeneration*.



Delete a feature

Deleting a feature removes the feature from the model permanently. If you delete a parent feature, all of its children will also be deleted and cannot be recovered.

In addition, you can reduce all the feature and parameter information to a simple definition of the part when you copy a part to a new part. If you reduce the feature list while copying a part, Abaqus/CAE will regenerate the new part faster if you subsequently modify it; however, you will no longer be able to modify any parameters of the new part. To copy a part, select **Part->Copy** from the main menu bar in the Part module.

Additional information

• *Modifying and manipulating features*
• *Copying a part*

Using the Model Tree to manage features

The Model Tree, shown in *Figure 1*, includes a list of all available features in the model. You can click mouse button 3 on a feature in the Model Tree and use the menu that appears to manage the features.

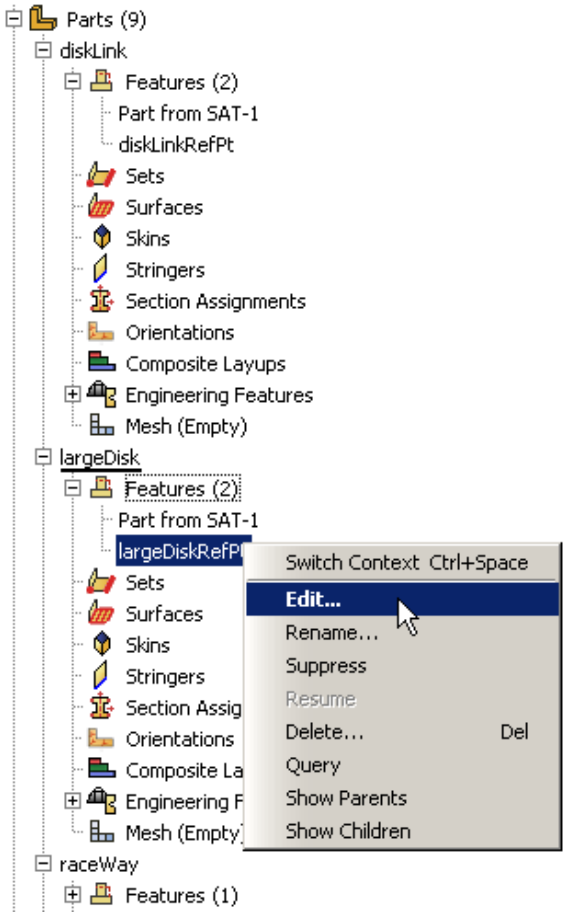


Figure 1: The Model Tree.

You must use the Model Tree to manipulate assembly constraints, suppressed features, or failed features since they are not visible in the viewport. You can also select multiple features to delete, suppress, or resume at the same time. As you select features in the Model Tree, they are highlighted in the viewport (if they are visible). In addition, parents or children of any feature can be highlighted in the viewport.

The Model Tree displays the status of each feature in the model. The icon before each feature name indicates the status:

- A yellow check mark indicates that the feature is modified but has not been regenerated. If a part is locked by a database upgrade, all active features show this status until they are regenerated when the part is unlocked. Abaqus/CAE also displays a yellow check mark next to a feature that you edited but did not subsequently regenerate.
- A padlock indicates that a part or the assembly has been locked by the user or by a database upgrade.
- A red “X” indicates that the feature is suppressed.
- A red “!” indicates that the feature has failed to regenerate. A red “!” next to the part name indicates that the part is invalid.

To get more information about a feature, click mouse button 3 on the feature in the Model Tree and select **Query** from the menu that appears. The information is displayed in the message area.

Additional information

- *Using the Feature Manipulation toolset*
- *Manipulating features in the Assembly module*
- *Modifying and manipulating features*

Tuning feature regeneration

This section describes how you can tune feature regeneration using settings in the **Feature Options** dialog box. You open the **Feature Options** dialog box by selecting **Feature->Options** from the main menu bar. In most cases the default settings will result in acceptable regeneration performance. You should modify the feature options only if Abaqus/CAE takes a long time to regenerate a part or the assembly or if memory consumption is excessive.

The settings in the **Feature Options** dialog box apply only to the current model and are saved in the model database. If you change to a different model, Abaqus/CAE reverts to the default settings or to any modified settings associated with the new model.

In this section:

- *What is regeneration?*
- *What is a geometric state?*
- *What is a cache?*
- *What are self-intersection checks?*
- *How are position constraints regenerated?*

What is regeneration?

Whenever you modify a feature of a part, Abaqus/CAE has to regenerate the part. Similarly, to incorporate the modified feature into any instances of the part, Abaqus/CAE also regenerates features in the assembly when you enter an assembly-related module. However, to save time, Abaqus/CAE determines the minimum number of features that need to be regenerated and regenerates only the features that were affected by the modification. If your parts and your assembly are relatively simple, you will not notice Abaqus/CAE performing this regeneration. However, in a more complex model the regeneration can result in a decrease in performance. Modifications to features that trigger regeneration include using the Feature Manipulation toolset to edit, suppress, resume, and delete a feature.

The Model Tree indicates that a feature needs to be regenerated using a yellow check mark, as described in *Using the Model Tree to manage features*. You can always force Abaqus/CAE to regenerate a part of the assembly by selecting **Feature->Regenerate** from the main menu bar or by clicking mouse button 3 on the **Features** container in the Model Tree.

What is a geometric state?

A geometric state refers to the internal geometric representation of a part or of the assembly; for example, the equations of the curves and surfaces that define the topology and the connectivity of those curves and surfaces. A geometric state is a snapshot of this internal representation. Abaqus/CAE stores this snapshot and creates a geometric state in memory at regular intervals while you work on your model and then regenerate features.

If Abaqus/CAE did not save geometric states, regeneration would start from the first feature that you created and continue through all the features that need to be regenerated, making regeneration very costly. By saving states, Abaqus/CAE can determine which state is closest to the feature that was modified. Regeneration starts from that point and continues through all the features that need to be regenerated. As a result, the speed of regeneration can be increased significantly.

Ideally, to gain the maximum regeneration performance, you would save the geometric state after every feature modification. However, storing every state would consume large amounts of memory and hinder overall performance. Deciding how many states to save is a tradeoff between regeneration speed and memory consumption.

By default, Abaqus/CAE automatically saves up to five geometric states for each part and for the assembly. If Abaqus/CAE has already stored the specified number of states, it deletes one of the older states before saving the most recent state. You can use the **Geometry Caching for Fast Regeneration** options in the **Options** dialog box to change the number of states automatically saved by Abaqus/CAE. Abaqus/CAE uses an internal logic to decide when to create a geometric state; you cannot control when a state is created.

What is a cache?

A cache is a portion of memory. Abaqus/CAE uses the following feature-related caches to store geometric states:

- The **Current part cache** contains snapshots of the geometric state of the current part. Abaqus creates one part cache for each part in the model.
- The **All part caches** contains snapshots of the geometric state of all the parts in the current model.
- The **Assembly cache** contains snapshots of the geometric state of the assembly. Abaqus creates one assembly cache for the model.

If you find that Abaqus/CAE is consuming an excessive amount of memory on your workstation and affecting the performance of your system, you can use the **Clear** buttons in the **Options** dialog box to erase the memory associated with the part and assembly caches. Clearing this memory will increase performance; however, the snapshots of the geometric state will be erased and regeneration will be slower. If you toggle off the option to cache geometric states, Abaqus/CAE does not clear the states that have already been stored. In addition, Abaqus/CAE stores a separate cache for each model. As a result, you can make more memory available by deleting the caches of any models that you are not currently working on.

If you spent a lot of time making a series of changes to a part and are satisfied that your design is correct, you can manually save the current state in memory by clicking **Cache current state** in the **Options** dialog box. As a result, Abaqus/CAE can return to this state and regenerate any subsequent additions or modifications. However, if you continue to automatically cache geometric states, Abaqus/CAE may erase the state that you saved manually after it has saved the maximum number of geometric states. Alternatively, you can toggle off the option to automatically cache geometric states and continue to manually save states at significant intervals.

The **Geometry Caching for Fast Regeneration** options in the **Options** dialog box indicate how many geometric states are stored in the three feature-related caches. A cache can contain a combination of geometric states that you saved manually and states that Abaqus/CAE saved automatically. You can query the **Current part cache** and the **Assembly cache** to determine the positioning of geometric states relative to the sequence of features.

What are self-intersection checks?

Self-intersection checks are tests used to help ensure that complex geometry you create in Abaqus/CAE is valid for analysis.
If faces of a feature intersect other faces of the same feature, Abaqus/CAE displays a warning that there are invalid intersections and does not create the feature. A feature that intersects itself may not be meshable and may cause problems with the analysis. Abaqus/CAE allows you to check for self-intersection during feature creation; however, the checking process will slow down creation of the feature. In addition, checking for self-intersection will slow down regeneration of the feature. The time required to complete the tests varies with the complexity of the feature you are attempting to create or regenerate.

The **Feature Options** dialog box allows you to toggle self-intersection checks on and off. **Perform self-intersection checks** is toggled off by default. In general, self-intersection is possible only when you are creating features with complex geometry, such as loft features (for more information, see *What is lofting?* and *Self-intersection checks*). If you know that your features do not self-intersect or if you plan to remove the self-intersection during a subsequent operation, you can leave **Perform self-intersection checks** toggled off to speed up feature regeneration.

How are position constraints regenerated?

By default, when Abaqus/CAE regenerates the assembly, it regenerates all of the position constraints before regenerating other features that might depend on the position, such as partitions and datums.

However, if the position constraint uses an entity that was created by selecting from a part instance in the Assembly module, such as a face created by a partition or a datum axis running through two vertices, Abaqus/CAE changes its behavior and regenerates features in the order that you created them. As a result, partitions and datums that you created before the position constraint may not move along with the movable instance.

If you are working in the assembly-related modules, the **Feature Options** dialog box includes a **Constraints** field that allows you to control this regeneration behavior. For more information, see *Tuning regeneration performance*.

Modifying and manipulating features

This section describes how to use the Feature Manipulation toolset.

In this section:

- *Editing a feature*
- *Renaming a feature*
- *Suppressing features*
- *Resuming suppressed features*
- *Deleting features*
- *Regenerating a part or assembly*
- *Tuning regeneration performance*
- *Using the Model Tree to obtain feature information*

Editing a feature

Select **Feature->Edit** from the main menu bar to modify geometric information for a selected feature. You can also edit a feature by clicking mouse button 3 on the feature in the Model Tree and selecting **Edit** from the menu that appears.

The changes that you can make to a feature depend on how it was created. You can change any numerical parameters that define the feature, and, where applicable, you can change a section sketch. However, you cannot modify a feature that was created by picking geometric entities, such as points and edges, on the screen. For example, you can modify the following parameters:

- Depth of an extrusion
- Radius of a fillet
- Coordinates of a datum point
- Shape of an extruded solid

If you cannot make the required changes with the Feature Manipulation toolset, you must delete the feature and create a new one that incorporates the changes.

1. From the main menu bar, select **Feature->Edit**.

    **Tip:** You can also edit a feature by clicking the  tool located in the module toolbox.

2. Select a feature to modify. You can select the feature either directly from the current viewport or from the Model Tree. For more information, see *Selecting objects within the viewport* and *Working with the Model Tree and the Results Tree*.

   The feature editor appears.

3. Edit the feature by doing one of the following:

   - In the **Parameters** field, type a new value for the desired parameter.
   - Select **Edit Section Sketch** to start the Sketcher and make the changes to your sketch. For information on using the Sketcher, see *The Sketch module*.

4. + Click **OK** to implement the changes in your model.

   If you are editing a part feature, the part is regenerated immediately to incorporate your changes, and the assembly is regenerated the next time you enter a module that displays the assembly. If you are editing an assembly feature, the assembly is regenerated immediately.

    **Note:** By default, Abaqus/CAE regenerates the part or assembly every time you edit a feature. If you prefer to postpone regeneration of the part or assembly, toggle **Regenerate on OK** off before you click **OK** in the feature editor. When you are ready to regenerate the part or assembly, select **Feature->Regenerate** from the main menu bar.

Additional information

• *Using the Feature Manipulation toolset*

Renaming a feature

You can rename a feature by clicking mouse button 3 on the feature in the Model Tree and selecting **Rename** from the menu that appears.

1. Click mouse button 3 on the feature in the Model Tree, and select **Rename** from the menu that appears.
   The **Rename Features** dialog box appears.
2. Enter a new name for the feature.
3. + Click **OK** to complete renaming the feature.
   The Model Tree displays the new name.

Additional information

• *Using the Feature Manipulation toolset*

Suppressing features

Select **Feature->Suppress** from the main menu bar to suppress selected features. Suppressing features is equivalent to temporarily removing them from the part or assembly. Suppressed features remain suppressed when you regenerate a part or assembly. You can also suppress a feature by clicking mouse button 3 on the feature in the Model Tree and selecting **Suppress** from the menu that appears.

You cannot suppress the base feature. In addition, if you suppress parent features, all of their child features are also suppressed automatically. Suppressed features can be restored by selecting **Feature->Resume** from the main menu bar or by clicking mouse button 3 on the feature in the Model Tree. When you resume a suppressed feature, its children are not resumed unless you select them. To help you select a feature's children, you can click mouse button 3 on the feature in the Model Tree and select **Show children** from the menu that appears.

1. From the main menu bar, select **Feature->Suppress**.

    **Tip:** You can also suppress features using the  tool located in the module toolbox.

2. Select one or more features to suppress. You can select features from the current viewport or from the Model Tree. For more information, see *Selecting objects within the viewport*," and *Working with the Model Tree and the Results Tree*.

   Abaqus/CAE highlights the selected feature. If you selected features that have child features, Abaqus/CAE also highlights the children.

   If you selected features from the viewport, click **Done** in the prompt area when you are finished making selections.

3. If you selected features from the viewport, click **Yes** in the prompt area. If you selected features from the Model Tree, click mouse button 3 on one of the feature names and select **Suppress** from the menu that appears.

   The selected features and any child features disappear from the viewport.

4. To exit the suppress procedure, either

   - click the cancel button  in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or
   - select another operation from the **Feature** menu.

Additional information

- *Using the Feature Manipulation toolset*
- *Resuming suppressed features*

Resuming suppressed features

Select **Feature->Resume** from the main menu bar to resume suppressed features. You can also resume a group of features by selecting the features in the Model Tree, clicking mouse button 3, and selecting **Resume** from the menu that appears.

Resuming a feature fully restores it to the part or assembly. You can resume the last feature you suppressed, all suppressed features, or just selected features. When you resume a child feature, Abaqus/CAE also resumes the parent features automatically. However, when you resume a parent feature, its children are not resumed unless you select them. To help you select a feature's children, you can click mouse button 3 on the feature in the Model Tree and select **Show children** from the menu that appears.

1. From the main menu bar, select **Feature->Resume**.

    **Tip:** You can also resume a feature using the  tool located in the module toolbox.

2. Do one of the following:

   - Click **Last Set** to resume the most recently suppressed feature and all its parents.
   - Click **All** to resume all suppressed features.
   - Select the feature to resume from the Model Tree. You can select multiple features to resume.

   The resumed features reappear in the viewport.

Additional information

- *Using the Feature Manipulation toolset*
- *Suppressing features*

Deleting features

Select **Feature->Delete** from the main menu bar to delete selected features. You can also delete a feature by clicking mouse button 3 on the feature in the Model Tree and selecting **Delete** from the menu that appears.

When you delete parent features, you also delete all of their child features. You cannot resume deleted features.

1. From the main menu bar, select **Feature->Delete**.

   Tip: You can also delete features using the tool located in the module toolbox.

2. Select one or more features to delete. You can select features from the current viewport or from the Model Tree. For more information, see *Selecting objects within the viewport*," and *Working with the Model Tree and the Results Tree*.

   Abaqus/CAE highlights the selected features. If you selected features that have child features, Abaqus/CAE also highlights the children.

   If you selected features from the viewport, click **Done** in the prompt area when you are finished making selections.

3. If you selected features from the viewport, click **Yes** in the prompt area. If you selected features from the Model Tree, click mouse button 3 on the feature and select **Delete** from the menu that appears.

   The selected features and any child features are deleted.

4. To exit the delete procedure, either

   - click the cancel button in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or
   - select another operation from the **Feature** menu.

Additional information

- *Using the Feature Manipulation toolset*

Regenerating a part or assembly

When you modify features in a complex part or assembly, it may be convenient to postpone regeneration until you make all your changes, since regeneration can be time consuming. (For more information, see *Editing a feature*.) When you are ready to regenerate the part or assembly, you can select **Feature->Regenerate** from the main menu bar. You can also regenerate a feature by clicking mouse button 3 on the feature in the Model Tree and selecting **Regenerate** from the menu that appears.

1. From the main menu bar, select **Feature->Regenerate**.

   Tip: You can also regenerate a feature using the  tool located in the module toolbox.

   Abaqus/CAE regenerates the part or assembly.

2. If regeneration fails, a dialog box appears. In this case you can choose one of the following options:

   - + Click**Undo Changes** to return to the original part or assembly and discard all your changes.
   - + Click**Keep Changes** to continue the regeneration, suppressing those features that failed to regenerate.

     Note:

     Features that are suppressed as a result of the failure are resumed automatically if you correct the cause of the failure.

Additional information

- *Using the Feature Manipulation toolset*
- *Suppressing features*

Tuning regeneration performance

The **Feature Options** dialog box provides settings that enable you to control whether Abaqus/CAE performs self-intersection checks and, in the Assembly module, to control the order in which Abaqus/CAE regenerates position constraints relative to other features. Tuning these options can help you speed up feature regeneration for your part or assembly. Select **Feature->Options** from the main menu bar to display the **Feature Options** dialog box. You can also tune regeneration by clicking mouse button 3 on the feature in the Model Tree and selecting **Options** from the menu that appears.

In most cases the default settings will result in acceptable regeneration performance. You should modify the feature options only if Abaqus/CAE takes a long time to regenerate a part or the assembly. For more information, see *Tuning feature regeneration*.



Note:

Abaqus/CAE provides access to additional performance tuning options on the **Memory** tabbed page of the **Options** dialog box. The options on this page enable you to specify a limit for kernel memory, set a threshold for when Abaqus/CAE runs in reduced memory mode, and control geometric states in a memory cache. For more information, see *Customizing memory limits and regeneration options*.

1. From the main menu bar, select **Feature->Options**.

   Abaqus/CAE displays the **Feature Options** dialog box.
2. If you know that your features do not self-intersect or if you plan to remove the self-intersection during a subsequent operation, you can toggle off **Perform self-intersection checks** to speed up feature regeneration. For more information, see *What are self-intersection checks?*.
3. If desired, modify the behavior that determines how Abaqus/CAE regenerates position constraints. For more information, see *How are position constraints regenerated?*.
4. + Click **OK** to change the feature options for the current model.

   The feature options apply only to the current model and are saved in the model database.

   If desired, you can also change the cache settings in the **Memory** tabbed page of the **Options** dialog box to change the number of geometric states that Abaqus/CAE automatically caches in memory; clear the memory associated with the part and assembly caches; or cache the current state of the part. For more information, see *Customizing memory limits and regeneration options*.

Using the Model Tree to obtain feature information

The Model Tree displays the feature names along with icons that indicate their status—active, suppressed, modified, or failed. In addition, you can display information on a particular feature by clicking mouse button 3 on the feature in the Model Tree and selecting from the menu that appears.

1. Click mouse button 3 on the feature in the Model Tree.
2. From the menu that appears, choose one of the following options:

   - Select **Show Parents** or **Show Children** to display the parents or children of that feature in the current viewport. Abaqus/CAE also selects the parents or children.
   - Select **Query** to display detailed information. Abaqus/CAE displays the following information about the selected feature:

     - Name and description; for example, solid extrude
     - Status—active or suppressed
     - The name of its parent, if any
     - The names of its children, if any
     - The value of any parameters that define the feature

   The query information appears in the message area and is also written to the replay file (`abaqus.rpy`) in the form of comments.

Additional information

• *Using the Feature Manipulation toolset*

The Filter toolset

Filters allow you to remove extraneous field output data or history output data—noise—during the analysis of a model without a loss of resolution in the desired data range. You can also use filters to filter field output or history output before the data are saved to the output database (`.odb`) file; as a result, filters can also reduce the size of the output database.
The Filter toolset allows you to create and manage filters in the Step module.

In this section:

- *Filtering field and history data*
- *Applying bounding values to field and history data*
- *Creating a filter*

Filtering field and history data

You can create filters in Abaqus/CAE and apply them to field output requests or history output requests for Abaqus/Explicit analyses. Abaqus filters data while the analysis is running; filtering during the analysis (“real time” filtering) can reduce the size of the output database by excluding high-frequency data before it is saved. Real time filtering also avoids potential aliasing problems in the resulting data. Aliasing is a loss of valid results data; aliasing will occur if the sampling frequency (the frequency at which the data are saved) is less than twice the highest frequency expected in the results. For example, if a sine wave was sampled at only two points, the aliased result would appear to be a straight line, whereas sampling at least four points would reproduce the shape of the curve.

The Filter toolset allows you to create the following filters for use with Abaqus/Explicit field output requests and history output requests:

- **Butterworth**
- **Type I Chebyshev**
- **Type II Chebyshev**

For more specific information about these filters, see *Filtering Output and Operating on Output in Abaqus/Explicit*. The different filter types are distinguished by their capabilities to transition from the acceptance of low-frequency data to the rejection of data above the filter's **Cutoff frequency**. An ideal filter would stop all data above the cutoff frequency and have a flat response (no affect on accepted data); “real” filters include a transition band around the cutoff where some data are accepted and they usually have some effect on the accepted data. The Butterworth filter provides a maximally flat response magnitude with a wider transition band (slower transition) than the Chebyshev filters. The Chebyshev filters introduce an oscillation—a ripple—in the response magnitude, but they have a narrower transition band than the Butterworth filter. The two types of Chebyshev filters differ in where in their response ripples occur; the **Ripple factor** indicates how much oscillation you will allow in exchange for an improved filter response. You can also specify the **Order** of the filter, which determines the size of the filter's transition band: the higher the order, the narrower the transition band, although the computational cost increases as the order increases. The filter order must be a positive, even integer no greater than 20.

In addition to the filters that you can define using the Filter toolset, Abaqus also includes a default **Antialiasing** filter. Abaqus automatically sets the cutoff frequency of the **Antialiasing** filter based on the time interval at which the field output or history output is saved during the analysis. When you define a filter, you must specify the cutoff frequency based on your knowledge of the frequencies expected in the solution. Whether you set the cutoff or Abaqus calculates it, no checks are performed to ensure that the cutoff frequency is appropriate. If the cutoff is set too low, valid data will be filtered from the results; if it is set too high (above half the sampling frequency), no filtering will be performed.

Select **Tools->Filter->Create** from the main menu to create a new filter definition; select **Edit** from the same menu to make changes to an existing definition. Either command opens the filter editor, which allows you to select the options and provide the data needed to define your filter.

To apply a filter to an analysis, include it in a field output request or history output request for an Abaqus/Explicit analysis procedure (for more information, see *Defining output requests*).

Additional information

• *Filtering Output and Operating on Output in Abaqus/Explicit*

Applying bounding values to field and history data

Bounding values in a filter definition enable you to investigate the maximum or minimum values for a particular output request and to establish an upper or lower bound on the variable values that are returned for an output request. You can establish bounding values for both filtered and unfiltered data; to specify these values in a filter definition without performing Butterworth filtering or either type of Chebyshev filtering on your data, define an **Operator** filter, which enables you to implement bounding values on an output request without changing the output data in any other way.

Abaqus/CAE provides three types of bounding values:

- **Maximum** bounding values return the highest value within each output interval for the variables in an output request.
- **Minimum** bounding values return the lowest value within each output interval for the variables in an output request.
- **Absolute maximum** bounding values return the highest absolute value for the variables in an output request within each output interval.

In addition, you can set a limit on the bounding value, which determines the highest or lowest value that Abaqus/CAE records for variables that use this filter. If desired, you can stop the analysis when any variables in an output request reach this limiting value.

By default, each component of a tensor or vector quantity is filtered individually and the maximum, minimum, or absolute maximum value and the limiting values are reported separately for each component. You can, however, apply a filter directly to an invariant.

Creating a filter

Select **Tools->Filter->Create** from the main menu bar to create a filter.
For more information on filters including a default filter type created automatically by Abaqus, see *Filtering field and history data*.

1. From the main menu bar, select **Tools->Filter->Create**.

   Tip: You can also create a filter by clicking **Create** in the **Filter Manager**.

   The **Create Filter** dialog box appears.

2. In the **Name** field, enter a name for the filter. For information on naming objects, see *Using basic dialog box components*.
3. From the **Type** list, select one of the following, then click **Continue**:

   - **Butterworth**
   - **Type I Chebyshev**
   - **Type II Chebyshev**
   - **Operator**

   The **Edit Filter** dialog box appears.

4. If you selected the **Butterworth** filter type or one of the **Chebyshev** filter types, perform the following steps:

   a. Enter the **Cutoff frequency**.

   Note:

   Abaqus does not perform any checks to ensure that the cutoff frequency is appropriate. If the cutoff is set too low, valid data will be filtered from the results; if it is set too high, no filtering will be performed.

   b. Enter the **Order** for the filter, which must be a positive, even integer value no greater than 20.

5. If you selected one of the **Chebyshev** filter types, accept the default ripple factor or enter a value.

   The ripple factor determines the magnitude of data oscillation that you will allow in exchange for a faster transition between accepted and rejected data. For more information, see *Filtering Output and Operating on Output in Abaqus/Explicit*.

6. If desired, specify the type of bounding value for which you want to filter data.

   - Select **Maximum** to output the maximum value within each output interval for each variable in the output request.
   - Select **Minimum** to output the minimum value within each output interval for each variable in the output request.
   - Select **Absolute maximum** to output the maximum absolute value within each output variable for each variable in the output request.

   You can also select **None** to toggle off all bounding value–related output requests for this filter.

7. If desired, specify a limit for bounding value requests by entering a value in the **Bounding value limit** field. This value limits the highest or lowest value that Abaqus/CAE records for variables that use this filter.

You can also stop the analysis if the bounding value limit is reached. Toggle on **Stop analysis upon reaching limit** to halt a job when the maximum or minimum value equals the limit value for variables that use this filter.

8. If you selected **Minimum**, **Maximum**, or **Absolute Maximum** for the type of bounding value, you can apply the filter directly to an invariant. To filter the invariant, select **First** or **Second** as the **Invariant**.
9. Click **OK** to create the filter.

To apply the filter in an analysis, select **Apply filter** when you create an output request for an Abaqus/Explicit analysis procedure. For more information, see *Modifying field output requests* or *Modifying history output requests*.

Additional information

- *Filtering field and history data*
- *Applying bounding values to field and history data*
- *Filtering Output and Operating on Output in Abaqus/Explicit*
- *Modifying field output requests*
- *Modifying history output requests*

The Free Body toolset

Free body cuts display the resultant forces and moments transmitted across a selected surface of your model. Free body cuts simply integrate the internal forces in an element over a section; therefore, they cannot be used accurately across sections containing surface tractions due to cohesive contact or other sources. The Free Body toolset is available only in the Visualization module, and you can create a free body cut only when the current step and frame of the output database includes element force nodal output (NFORC).

This chapter explains how to use the Free Body toolset to create and delete free body cuts, display or hide them in the viewport, and customize several aspects of their appearance.

Abaqus/CAE also enables you to generate reports or create *X–Y* data objects that describe the forces and moments in all active free body cuts in your session. See *Producing a tabular report*, and *Reading X–Y data from all active free body cuts*, respectively.

Abaqus/CAE provides two other methods for displaying free body data:

- You can display resultant forces and moments along an arbitrary plane through your model by toggling on free body display for a planar view cut. For more information, see *Understanding view cuts*.
- You can display the free body nodal forces in a symbol plot to determine which nodes have an imbalance of forces or moments due to applied loads or to visualize the nodal force distribution on internal sections. For more information, see *Producing a symbol plot of free body nodal forces*.

In this section:

- *Resultant forces and moments on free body cuts in Abaqus/CAE*
- *Creating or editing a free body cut*
- *Selection methods for free body cross-sections*
- *Displaying, hiding, and highlighting free body cuts*
- *Customizing free body cut display*

Resultant forces and moments on free body cuts in Abaqus/CAE

A free body cross-section in Abaqus/CAE is an area of your model across which you want to display resultant forces and moments. Once you define a cross-section, Abaqus/CAE displays vectors that show the magnitude and direction of the resultant forces and moments across the area you select. Force vectors are displayed with a single arrowhead and moment vectors with a double arrowhead, as shown in the example in *Figure 1*; by default, the force vectors are red and the moment vectors are blue.

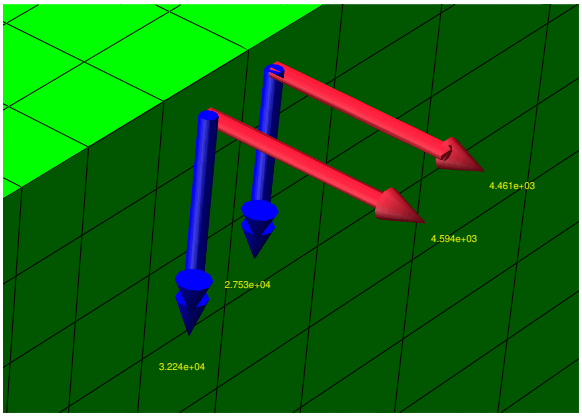


Figure 1: Resultant force and moment display.

You can define the nodes and elements that comprise the cross-section using processes that closely resemble the definition of display groups. Abaqus/CAE enables you to specify the components of the cross-section by including surfaces, display groups, and elements or nodes by number; and you can pick items from the viewport, either by feature angle or individually. See *Creating or editing a free body cut* for detailed instructions. After you define the physical components of your free body cross-section, you can set the location of the summation point (about which resultant moments are taken), and you can indicate the coordinate system transformation that applies when vectors are displayed in component form.

Cross-sections can be created along mesh boundaries only; you cannot specify a cross-section along an arbitrary plane. You can create and display free body cuts only in the Visualization module.

Creating or editing a free body cut

To create a free body cut, select the nodes and elements that comprise your free body cross-section; that is, select the surface of your model across which you want to investigate the resultant force and moment.

Abaqus/CAE provides several methods for selecting the nodes and elements for a cross-section:

- For three-dimensional objects, you can select individual element faces to include in the cross-section. The contribution from all elements connected to the nodes of the selected faces will be considered.
- For two- or three-dimensional objects, you can select individual element edges to include in the cross-section. Each edge provides Abaqus/CAE with an element definition and two or three nodes to include in the cross-section. The contribution from all elements connected to the nodes of the selected edges will be considered.
- For any type of model, you can select the nodes and elements you want to include individually. This method is recommended for selecting nodes and elements from geometrical areas, such as the ends of beams, in which it might be awkward or difficult to select nodes and elements using their edges or faces. The contribution from only the selected elements will be considered.

You can also display a free body cut on a view cut. For more information, see *Understanding view cuts*.

When you define a cross-section using element faces or element edges, Abaqus/CAE provides three additional selection methods that enable you to select the nodes and elements for this definition:

- You can select the faces or edges by picking components in the viewport, either individually or by angle.
- You can select a surface set, which is defined in the output database.
- You can select one or more display groups to use as part of your surface definition.

When you define a cross-section by selecting nodes and elements individually, you can specify the nodes or elements you want to include in the definition by typing their labels directly into the **Free Body Cross-Section** dialog box.

Abaqus/CAE also enables you to customize the summation point for a free body cut, which is the three-dimensional location about which resultant moments are taken. By default, the summation point is placed at the area centroid of the element faces contributing to the free body computation, but you can move the summation point to the centroid location of the nodes for the cross-section or to any three-dimensional location in your model.

You can also adjust the orientation of component vector display for a free body cut. In many cases the most desirable coordinate system for a free body cut is one in which one axis is normal to the surface of the model and another axis is tangential to the surface. This “normal and tangential” coordinate system is created internally and used by default for the display of component vectors. However, you can change the orientation to use any custom coordinate system available in your session, or you can create a new datum coordinate system.

When you edit a free body cut, Abaqus/CAE enables you to change the components included in the free body cross-section, the summation point settings, and the component resolution settings. However, you cannot change the selection method that was used to define a free body cut; for example, if you defined the free body cut using three-dimensional element faces, your subsequent edits to the free body cut will also use this method.

By default, free body cuts persist only for the duration of your session. If you want to retain a free body cut that you have defined to make it available in subsequent sessions, you can save it to an XML file, to the model database, or to an output database. For more information, see *Managing session objects and session options*.

1. Locate the options for creating or editing a free body cut:
   - To create a free body cut, select **Tools->Free Body Cut->Create** from the main menu in the Visualization module.

     Tip: You can also create a free body cut using the  tool in the Visualization module toolbox.

- To edit a free body cut, select **Tools->Free Body Cut->Edit** from the main menu in the Visualization module, and select the free body cut you want to edit.

  Tip: You can also use the free body cut manager to edit a free body cut. From the main menu bar, select **Tools->Free Body Cut->Manager**, and click **Edit** from the dialog box that appears.

  The **Free Body Cross-Section** dialog box opens, from which you can customize the edges, faces, or elements and nodes that constitute the cross-section.

2. If you are creating a free body cut, choose one of the following from the **Selection Method** options, and click **OK**:

   - Select **Based on view cut** to calculate and display a free body cut based on the current view cut. If you choose this selection method, Abaqus/CAE opens the **View Cut Manager**, from which you can display and position a view cut for your model and customize the display of an associated free body cut. For more information, see *Customizing display and calculation of resultant force and moment on the active view cuts*.
   - Select **2D element edges** to specify the nodes and elements implicitly (or indirectly) by selecting element edges. This method is recommended for two-dimensional models.
   - Select **3D element faces** to specify the nodes and elements implicitly (or indirectly) by selecting element faces. This method is recommended for three-dimensional models.
   - Select **Elements and nodes** to select the elements or nodes individually for your free body cross-section.

   The **Free Body Cross-Section** dialog box opens, from which you can select the edges, faces, or elements and nodes that constitute the cross-section.

3. From the **Item** list at the top left of the dialog box, select **Surfaces**, **Display groups**, **Elements**, or **Nodes** as the type of model component to use for the free body cross-section.

   Abaqus/CAE refreshes the **Method** list.

4. Choose a selection method from the **Method** list; and/or select the specific items for the free body cross-section by picking from the viewport, selecting items from the list that appears on the right side of the dialog box, or entering data in the right side of the dialog box.

   For more information about selection options, see *Selection methods for free body cross-sections*.

   Certain items can be highlighted in the viewport to verify your selection. Toggle **Highlight items in viewport** if available.

5. When you have completed your selections, click **OK** to close the **Free Body Cross-Section** dialog box.

   The **Edit Free Body Cut** dialog box opens.

6. If desired, customize the summation point or the coordinate system transformation options for the selected free body cut.

   a. From the **Summation point** options, select the three-dimensional location from which the vectors in the free body cut originate.

      - Select **Centroid of cut** to place the summation point automatically at the area centroid of the element faces contributing to the free body computation.



Note:

When the view cut cuts through a shell, the centroid is based on the length times the shell thickness.

When the view cut cuts through a beam, the centroid is at the nodal connectivity line.

- Select **Nodal average** to place the summation point automatically at the averaged deformed coordinates of all nodes included in the free body cut.
- Select **User-defined**, and specify a custom three-dimensional location in space or click  to select the summation point from the viewport. The summation point is highlighted in the viewport.

b. From the **Component resolution** options, you can specify the coordinate system transformation that takes place when vectors are displayed in component form. (See *Customizing general display options for free body cuts*, for more information about displaying force and moment vectors in component form.)

- Select **Normal and tangential** to orient the component vectors with the normal and the tangent of the surface you select.
- Select **CSYS** and a coordinate system to transform the component vectors to a custom coordinate system. Alternatively, you can click **Create** to create a new datum coordinate system.

The **Component resolution** options affect display of free body cuts only when component vector display is selected in the **Free Body Plot Options** dialog box.

7. Click **OK**.

Abaqus/CAE creates the free body cut and displays it in the viewport.

Additional information

- *Displaying, hiding, and highlighting free body cuts*
- *Customizing free body cut display*

Selection methods for free body cross-sections

You can use the following selection methods to specify the items that will be contained in a free body cross-section:

- To specify edges or faces by picking them directly from the viewport, select **Surfaces** from the **Item** list, then select **Pick from viewport** from the **Method** list. Abaqus/CAE automatically enters the pick mode and prompts you to select items for the display group. See *Selecting objects within the viewport* for more information on picking items in the viewport.

  Click **Done** in the prompt area when you have finished picking items in the viewport.

  Click **Edit selection**, **Add selection**, or **Delete selection** in the dialog box to further modify your viewport selections.
- To specify elements or nodes by number, select **Element labels** or **Node labels** from the **Method** list. Select the name of the part instance to which the nodes or elements belong from the list in the **Part instance** field on the right side of the **Free Body Cross-Section** dialog box. In the **Labels** field, type a list of element or node numbers separated by commas or a range of numbers optionally followed by an increment; for example, 1:10 or 1:10:2.

  You can specify elements or nodes by number when you create a free body cross-section using the **Elements and nodes** selection method.
- To specify element, node, or surface sets, select **Element sets**, **Node sets**, or **Surface sets** from the **Method** list. A list of the available set names in your model appears on the right side of the **Free Body Cross-Section** dialog box. Select one or more set names from this list (see *Selecting multiple items from lists and tables* for more information).
- If you selected **Display groups** from the **Item** list, a list of the available display groups in your model appears on the right side of the **Free Body Cross-Section** dialog box. Select one or more display group names from this list (see *Selecting multiple items from lists and tables* for more information).

Displaying, hiding, and highlighting free body cuts

You can display or hide free body cuts individually in the current viewport by toggling their **Show** check boxes in the **Free Body Cut Manager**, and you can display or hide all of the free body cuts in your session by clicking **Set All On** or **Set All Off**. Active free body cuts are displayed in the viewport and are included in free body cut reports; for more information about reporting free body cut data, see *Generating tabular data reports*. The manager displays all free body cuts defined in your session and indicates whether each one is currently displayed.

You can highlight one or more free body cuts in the viewport by selecting their shortcuts under the **Free Body Cuts** container in the Results Tree.

By default, Abaqus/CAE retains no record of which free body cuts were active when you save free body cut definitions to a file for future use. If you want to retain a record of the activation status of free body cuts, toggle on **List of active view cuts + free bodies** when you save session objects to a file. For more information, see *Managing session objects and session options*.

Customizing free body cut display

This section describes how you can customize the content and appearance of free body cuts using settings in the **Free Body Plot Options** dialog box.
You can open the **Free Body Plot Options** dialog box by selecting **Options->Free Body** from the main menu bar of the Visualization module of Abaqus/CAE.

The changes you make in the **Free Body Plot Options** dialog box apply to all active free body cuts, including those displayed on view cuts, in the current viewport only. Each viewport has separate free body plot options, which provides greater flexibility in displaying free body cuts. For example, you can display only the resultant force vectors for a model's free body cuts in one viewport and display only the resultant moment vectors for the same model and free body cuts in a different viewport.

You can also control the default settings that appear in the **Free Body Plot Options** dialog box by editing the Abaqus environment file (`abaqus_v6.env`). For more information, see *Environment File Settings*.

In this section:

- *Customizing general display options for free body cuts*
- *Customizing colors for vector components*
- *Displaying or hiding individual component vectors*
- *Scaling vector length for free body cuts*
- *Customizing the display of labels in a free body cut*

Customizing general display options for free body cuts

The options on the **General** tabbed page of the **Free Body Plot Options** dialog box enable you to toggle the display of force vectors and moment vectors for free body cuts. You can also customize vector display by showing only the resultant vector or showing component vectors in the *x*-, *y*-, or *z*-directions for both force and moment values.

1. Locate the **General** options.

   Select **Options**->**Free Body** from the main menu bar in the Visualization module.
2. Toggle on **Show forces** to display force vectors for all free body cuts in the current viewport.
3. Toggle on **Show moments** to display moment vectors for all free body cuts in the current viewport.
4. Toggle on **Use constant-length arrows** to display all force and moment values with arrows of the same length. When this option is not selected, the force and moment vectors are displayed with lengths according to their magnitude.
5. From the **Vector display** options, select one of the following options:

   - Select **Resultant** to display only the resultant force and moment vectors.
   - Select **Component** to display the component force and moment vectors in the *x*-, *y*-, or *z*-directions.
6. Click **OK** to implement your changes in the current viewport and to close the **Free Body Plot Options** dialog box.

   By default, your changes are saved for the duration of the session and will affect all subsequent free body cuts plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing colors for vector components

You can adjust the color that Abaqus/CAE uses to display the resultant vector and each component vector in the 1-, 2-, or 3-directions. These settings are available independently for force vectors and moment vectors; the **Free Body Plot Options** dialog box provides identical (but independent) options for **Force** and **Moment** vector colors. By default, the resultant force and component vectors are red and the resultant moment and component vectors are blue.

1. Locate the **Colors** options.

   Select **Options->Free Body** from the main menu bar in the Visualization module, and click the **Color & Style** tab in the dialog box that appears. The **Colors** options appear at the top of the page.

2. Click the **Force** tab or the **Moment** tab to set vector colors for force vectors or moment vectors, respectively.

3. Change the vector color for one of the vectors:

   a. Click the color sample () for the resultant vector or for one of the component vectors.

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.

      The color sample and the selected vector change to the selected color.

   Repeat these steps to change the vector color for one or more of the other vectors.

4. Click **OK** to implement your changes and to close the **Free Body Plot Options** dialog box.

   By default, your changes are saved for the duration of the session and will affect all subsequent free body cuts plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Displaying or hiding individual component vectors

When free body cuts are configured to display individual component vectors in the 1-, 2-, or 3-directions instead of the resultant vector, you can display or hide each of these individual component vectors. These settings are available independently for force vectors and moment vectors; for example, you can hide component vectors for forces in the 3-direction but display component vectors for moments in the same direction.

Toggling the display of individual component vectors does not change the resultant vector display. Resultant vectors are still displayed using all three components.

1. Locate the **Color & Style** options.

   Select **Options**->**Free Body** from the main menu bar in the Visualization module, and click the **Color & Style** tab in the dialog box that appears.

2. Click the **Force** tab or the **Moment** tab to display or hide individual component vectors for force vectors or moment vectors, respectively. The **Components** options appear in the middle of the page.
3. Toggle on the **1**, **2**, or **3** options to include the selected component in vector display.
4. Click **OK** to implement your changes and to close the **Free Body Plot Options** dialog box.

   By default, your changes are saved for the duration of the session and will affect all subsequent free body cuts plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Scaling vector length for free body cuts

The **Scaling** options enable you to adjust the length of the vectors displayed in your free body cuts. Abaqus/CAE calculates vector length as a percentage of the size of your model or the size of your viewport. If multiple free body cuts are displayed, Abaqus/CAE uses the largest force vector or moment vector as the basis for length calculations. You can change the percentage value to increase or decrease the size of vectors, and you can choose either the model or the viewport as the basis for the vector length calculation. You can also set different vector scaling values for display of force vectors and moment vectors; the **Free Body Plot Options** dialog box provides identical (but independent) options for **Force** and **Moment** vector scaling.

1. Locate the **Color & Style** options.

   Select **Options->Free Body** from the main menu bar in the Visualization module, and click the **Color & Style** tab in the dialog box that appears.
2. Click the **Force** tab or the **Moment** tab to change vector scaling settings for force vectors or moment vectors, respectively. The **Scaling** options are in the lower third of the page.
3. From the **Basis** options, select **Screen size** or **Model size**.
4. Drag the **Percent** slider to establish the vector length as a percentage of the size of the screen or model.
5. Click **OK** to implement your changes and to close the **Free Body Plot Options** dialog box.

   By default, your changes are saved for the duration of the session and will affect all subsequent free body cuts plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing the display of labels in a free body cut

Click the **Labels** tab to customize the display of the numeric labels that appear with the vectors in a free body cut. You can customize the appearance of the force and moment vector labels separately; the **Free Body Plot Options** dialog box includes identical (but independent) options for **Force** and **Moment** vector labels.

You can toggle the display of numeric labels on and off, and you can customize the following settings for these labels:

- The text color of the labels.
- The font of the labels.
- The format of the labels (scientific, fixed, or engineering).
- The number of decimal places displayed for the label values.

In addition, you can establish threshold values for label display. If a vector's value is smaller than the threshold value, Abaqus/CAE hides its label.

1. Locate the **Labels** options.

   Select **Options->Free Body** from the main menu bar in the Visualization module, and click the **Labels** tab in the dialog box that appears.
2. Click the **Force** tab or the **Moment** tab to customize label display for force vectors or moment vectors, respectively.
3. Toggle on **Display value next to vector symbol** to show vector labels for the selected type of vector.
4. Select the color of the vector labels for the selected type of vector.

   a. Click the **Text color** color sample .

      Abaqus/CAE displays the **Select Color** dialog box.

   b. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.

   c. Click **OK** to close the **Select Color** dialog box.
5. Select the font of the vector labels for the selected type of vector.

   a. Click **Set Font**.

      Abaqus/CAE displays the **Select Font** dialog box.

   b. Use one of the methods in the **Select Font** dialog box to select a new font, size, and style. For more information, see *Customizing fonts*.

   c. Click **OK** to close the **Select Font** dialog box.
6. To change the format of the vector labels for the selected vector type, click the **Number Format** arrow and select one of the following formats:

   - Select **Scientific** to display legend values in scientific notation, which expresses legend values as a decimal value between 1 and 10 multiplied by a power of 10. When you select this format, the number 501,000 is displayed in the legend as `5.01e5`.
   - Select **Fixed** to display legend values without using exponent values. When you select this format, the number 501,000 is displayed in the legend as `501000.00`.
   - Select **Engineering** to select engineering notation, which expresses values in a similar format to scientific notation but allows exponent values that are multiples of three only. When you select this format, the number 501,000 is displayed in the legend as `501E3`.
7. Click the **Decimal places** arrows to specify the desired number of decimal places for each label.

8. Specify a **Threshold** value for the selected type of vector label. Vectors with a value smaller than this threshold will have their labels hidden in the viewport. The default threshold for vector label display is `1E-006`.
9. Click **OK** to implement your changes and to close the **Free Body Plot Options** dialog box.

   By default, your changes are saved for the duration of the session and will affect all subsequent free body cuts plots in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

The Options toolset

The Options toolset allows you to control memory and regeneration options, configure view manipulation shortcuts, and specify a scale factor for icons.

In this section:

- *Customizing memory limits and regeneration options*
- *Using view manipulation shortcuts*
- *Scaling the size of icons*

Customizing memory limits and regeneration options

A geometric state is a snapshot of the internal representation of a part or an independent part instance. Saving geometric states in a memory cache speeds up regeneration performance; however, the saved states can consume large amounts of memory and decrease overall performance.

The options on the **Memory** tabbed page of the **Options** dialog box allow you to control the amount of memory allocated to the Abaqus kernel, specify the percentage of memory use that will prompt Abaqus/CAE to run in reduced memory mode, and tune the balance between the convenience of saving states and the performance degradation from increasing memory consumption. Select **Tools->Options** from the main menu bar to display the **Options** dialog box, then click the **Memory** tab.

In most cases the default settings will result in acceptable regeneration performance. You should modify the feature options only if Abaqus/CAE takes a long time to regenerate a part or the assembly. For more information, see *Tuning feature regeneration*.

1. From the main menu bar in any module, select **Tools->Options**.

   The **Options** dialog box appears.
2. Display the **Memory** tabbed page.
3. If desired, specify a new **Kernel memory limit** in megabytes. If this setting is zero, no limit is imposed on kernel memory.
4. Toggle on **Run in reduced memory mode when memory reaches**, and specify a percentage of the kernel memory limit above which Abaqus/CAE runs in reduced memory mode. Abaqus/CAE reduces memory by either paging geometric data to disk or by deleting data in inactive parts that can be automatically regenerated when needed.
5. If desired, change the number of geometric states that Abaqus/CAE automatically caches in memory. Ideally, to gain the maximum regeneration performance, you would save the geometric state after every feature modification. However, storing a large number of geometric states consumes large amounts of memory and hinders overall performance. Deciding how many states to save is a tradeoff between regeneration speed and memory consumption. For more information, see *What is a geometric state?*.
6. If you find that Abaqus/CAE is consuming a lot of memory on your workstation and affecting the performance of your system, you can use the **Clear** buttons in the **Options** dialog box to erase the memory associated with the part and assembly caches. Clearing this memory will increase performance; however, the snapshots of the geometric state will be erased and regeneration will be slower. For more information, see *What is a cache?*.
7. If you want to save a current state in memory, click **Cache current state** in the **Options** dialog box. Abaqus/CAE can return to this state and regenerate any subsequent additions or modifications.

   The **Options** dialog box indicates how many geometric states are stored in the caches. Click **Query** to determine the positioning of geometric states relative to the sequence of features in the current part cache or the assembly cache. When Abaqus/CAE is running in reduced memory mode, the geometric states may be deleted if the memory consumption is higher than the specified threshold.
8. + Click **OK** to change the memory limits and regeneration performance options for the current model.

   The options apply only to the current session and are not saved in the model database.

Using view manipulation shortcuts

You can access the pan, rotate, and magnify view manipulation tools using combinations of keyboard and mouse actions. These shortcuts can make it easier and less interruptive to manipulate the view of a model while performing other tasks (for example, selecting elements to define a set). Abaqus/CAE activates the specified view manipulation tool as long as the associated combination of keyboard and mouse buttons is depressed; to exit the view manipulation tool, release the associated keyboard and mouse buttons.

In addition to the default Abaqus/CAE shortcuts, you can configure the view manipulation shortcuts to mimic five other common CAD applications, as outlined in *Table 1*. Change the shortcuts configuration by selecting **Tools->Options** from the main menu bar and changing the **View Manipulation** options. Your configuration is saved for future Abaqus/CAE sessions.

Table 1: View manipulation shortcuts for available configurations ("MB" stands for "mouse button").

Application	Pan	Rotate	Zoom
Abaqus/CAE (default)	**[Ctrl][Alt][MB2]**	**[Ctrl][Alt][MB1]**	**[Ctrl][Alt][MB3]**
CATIA V5	**[MB2]**	**[MB2][MB3]**; or **[MB2][MB1]**	**[MB2][MB3]**, then release **[MB3]**; or **[MB2][MB1]**, then release **[MB1]**
SOLIDWORKS	**[Ctrl][MB2]**	**[MB2]**	**[Shift][MB2]**
HyperView	**[Ctrl][MB3]**	**[Ctrl][MB1]**	**[Ctrl][MB2]**
Pro/ENGINEER Wildfire	**[Shift][MB2]**	**[MB2]**	**[Ctrl][MB2]**
UGS NX	**[MB2][MB3]**	**[MB2]**	**[MB2][MB1]**

Note:

The HyperView key combination for rotation (**[Ctrl][MB1]**) conflicts with the Abaqus/CAE use of this combination to unselect objects in the viewport. To make complex viewport selections where unselecting individual objects may be useful, it is recommended that you choose one of the other shortcut configurations.

Nondefault configurations of the view manipulation shortcuts change the behavior of the magnify tool. In the default Abaqus/CAE configuration, the magnify tool operates by dragging the cursor horizontally; in all other configurations, the magnify tool operates by dragging the cursor vertically.

In the default Abaqus/CAE shortcuts configuration, you can access the alternate modes of the pan, rotate, and magnify tools by holding the **[Shift]** key in addition to the other shortcut keys (see *The view manipulation tools*, for details). This convention conflicts with some of the shortcut combinations in nondefault configurations; therefore, you cannot access alternate modes of the view manipulation tools while using nondefault shortcuts. The alternate modes are always available, regardless of your shortcuts configuration, if you access a view manipulation tool using the standard methods (selecting a tool from the **View** menu in the main menu bar or clicking a tool in the View Manipulation toolbar).

1. From the main menu bar in any module, select **Tools->Options**.

   The **Options** dialog box appears.
2. Display the **View Manipulation** tabbed page.
3. Select the **Application** whose view manipulation shortcuts you want to mimic. Refer to *Table 1* for a summary of shortcuts in different applications.
4. Click **OK** to apply your specified configuration and to close the **Options** dialog box.

Scaling the size of icons

The default size for icons in toolbars and toolboxes is 24 × 24 pixels. The default size for icons in the Model Tree and the Results Tree is 16 × 16 pixels. You can specify a scale factor to increase or decrease the size of these icons so that they appear at a size that is appropriate for your display resolution. For example, increasing the icon size scale factor to 1.5 prompts Abaqus/CAE to display toolbar and toolbox icons at 36 × 36 pixels and Model Tree and Results Tree icons at 24 × 24 pixels.

Changes to icon size are implemented when you restart Abaqus/CAE. Your selected scale factor is saved for future Abaqus/CAE sessions.

1. From the main menu bar in any module, select **Tools->Options**.

   The **Options** dialog box appears.
2. Display the **Icons** tabbed page.
3. Click the arrows to the right of the **Scale factor** field to increase or decrease the size of icons in Abaqus/CAE. The maximum scale factor is 3; the minimum scale factor is 0.5.
4. Click **OK** to apply your specified configuration and to close the **Options** dialog box. You must restart your Abaqus/CAE session to view your changes.

The Geometry Edit toolset

The Geometry Edit toolset provides a set of tools in the Part module that allow you to create or edit the geometry of a part.
You can use the tools to edit the regions of a part that make it invalid or imprecise.

In this section:

- *Editing techniques*
- *Editing techniques*
- *What is stitching?*
- *A strategy for repairing geometry*
- *Creating a part from orphan elements*
- *Editing and repairing edges*
- *Editing and repairing faces*
- *Editing parts*

Editing techniques

You can use the Geometry Edit toolset in the Part module to edit or repair the geometry of a part. The Geometry Edit toolset enables you to edit an imported part to improve its precision and validity; however, Abaqus/CAE does not require parts to be completely precise. In addition, you can use the Geometry Edit toolset to remove small features from a part. You can use the Geometry Edit toolset to edit edges, faces, or the entire part. *Editing techniques*, provides an overview of the methods available for each type of partition. Abaqus/CAE stores most of the edit operations as features. As a result, you can undo an edit by deleting or suppressing the corresponding feature using the Feature Manipulation toolset.

You can access the edit tools by selecting **Tools->Geometry Edit** from the main menu and choosing the desired tool from the **Geometry Edit** dialog box that appears. You can also access the Geometry Edit tools through the Part module toolbox. *Figure 1* shows the hidden icons for all the Geometry Edit tools in the Part module toolbox.

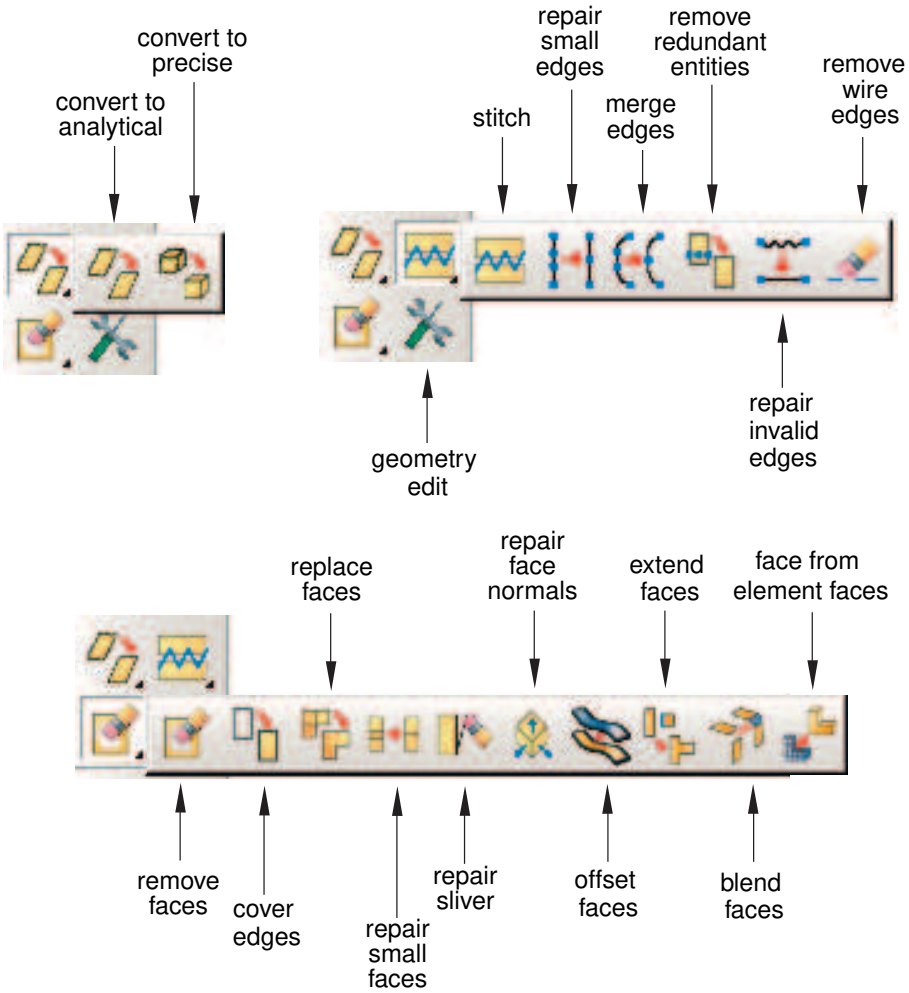


Figure 1: The Geometry Edit toolset toolbox.

To see a tooltip containing a short description of each Geometry Edit tool, hold the mouse over the tool for a moment. For more information, see *Using toolboxes and toolbars that contain hidden icons*.

Editing techniques

This section provides an overview of the different editing techniques.

In this section:

- *Methods for editing edges*
- *Methods for editing faces*
- *Methods for editing entire parts*

Methods for editing edges

Select **Tools->Geometry Edit** from the main menu to display the **Geometry Edit** dialog box.
When you choose **Edge** from the dialog box, the **Tool** list displays the following methods for editing edges:

Stitch

If a part is imported as a group of disconnected faces, you can stitch the resulting small edge gaps. Similarly, you can stitch the resulting gaps after you remove small faces or small slivers from a part. You can perform stitching as a global operation during which Abaqus/CAE stitches all gaps in the part, or you can pick the edges that you want to stitch, stitch edges with gaps smaller than a user-specified tolerance, or use both of these options. You should perform a global stitching operation for your entire part for small gaps only, and this process can be lengthy. For more information, see *What is stitching?*. You can use the Query toolset to highlight any free edges. For more information, see *Using the geometry diagnostic tools*.

Repair small

You can repair selected small edges. Abaqus/CAE removes the small edges and edits the adjoining edges to create a closed geometry.

Merge

You can select a series of connected edges, and Abaqus/CAE merges them into a single edge and deletes redundant vertices along the edges. *Figure 1* illustrates the effect of merging edges.

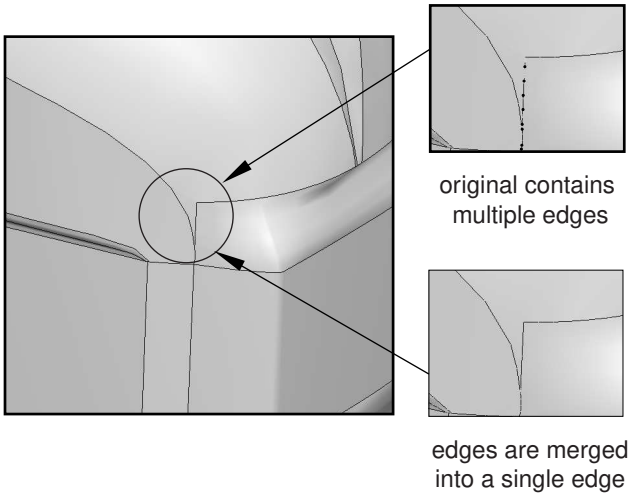


Figure 1: Merging edges.

Remove redundant entities

An imported part can contain redundant vertices that are positioned along a continuous edge. Similarly, an imported part can include redundant edges that are internal edges. Redundant vertices and edges do not change the shape or the area of a part and are not required for a complete definition, as shown in *Figure 2*.

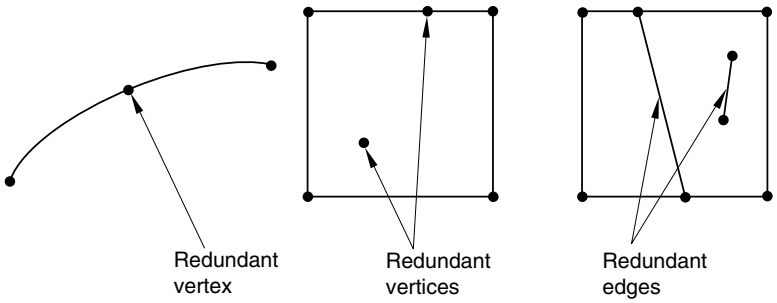


Figure 2: Redundant edges and vertices.

Repair invalid 

In rare cases after you import a part, Abaqus will report that some of its edges are invalid. The **Repair invalid** tool will try to repair the invalid edges by recomputing the data that define them. You should also use this tool if the Query toolset indicates that the part contains only invalid edges.

Remove wire 

You can remove selected wire edges. Abaqus/CAE removes the wire edges.

Methods for editing faces

When you choose **Face** from the **Geometry Edit** dialog box, the **Tool** list displays the following methods for repairing faces:

Remove 

You can remove selected faces (including chamfers, fillets, and holes) from a three-dimensional solid or shell or from a two-dimensional planar part. After you have selected the faces to remove, Abaqus/CAE looks for adjacent faces that define a feature. You can remove the entire feature, or you can remove only the selected faces. When you remove one or more faces from a three-dimensional solid part, Abaqus/CAE converts the part to a shell.

Cover edges 

You can create a face on a three-dimensional part by selecting one or more edges of the new face. Abaqus/CAE loops through the adjacent edges and calculates the location of the new face. If you selected multiple edges that are not connected by a common loop, Abaqus/CAE creates multiple faces, one for each loop of edges. Abaqus/CAE creates the new faces as shells. If the new shells form a closed part, you can use the solid-from-shell tool to convert the part to a solid.

Replace 

In some cases Abaqus/CAE may not be able to accurately recreate some of the faces when you import a part. For example, a planar face may appear wavy or distorted, or Abaqus/CAE might create small faces that have a large impact on the mesh density. You can select connected faces, and Abaqus/CAE replaces them with a single face. The new face has minimal faceting and will be smoother than the original faces. *Figure 1* illustrates the effect of replacing selected faces.

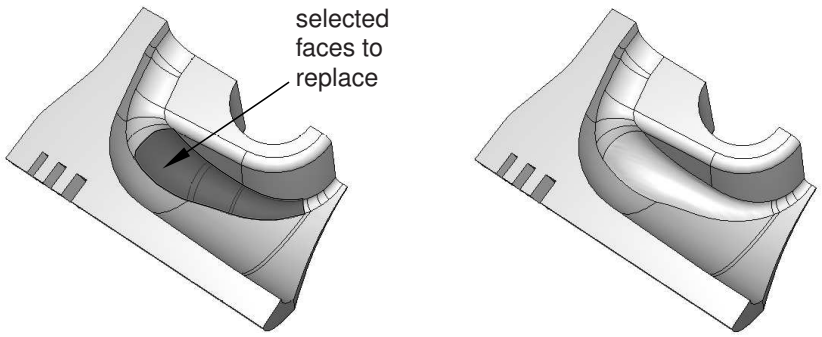


Figure 1: The effect of replacing selected faces.

Alternatively, you can replace the selected faces with a single face that is formed by extending neighboring faces. You can use this tool to remove bosses and small details from imported parts, as shown in *Figure 2*.

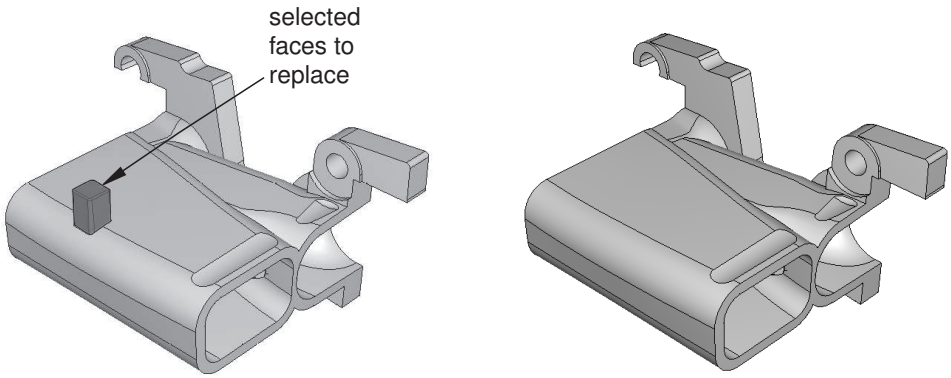


Figure 2: The effect of replacing selected faces and extending neighboring faces.

Repair small

You can repair selected small faces. Abaqus/CAE removes the small faces and edits the adjoining faces to create a closed geometry. *Figure 3* illustrates the effect of repairing small faces.

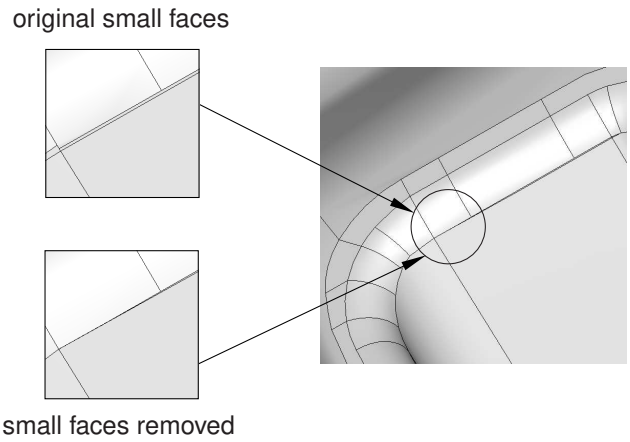


Figure 3: Repairing small faces.

Repair sliver

A sliver can be thought of as a small, sharp piece of extra material. You can remove an unwanted sliver from a face of a three-dimensional solid or shell or from a two-dimensional planar part. You must select the face containing the sliver to remove and two points from the face. Abaqus/CAE draws a line between the two points that divides the selected face into two regions. The first region is the face that will remain; the second region is the sliver that will be removed. *Figure 4* illustrates the effect of removing a sliver.

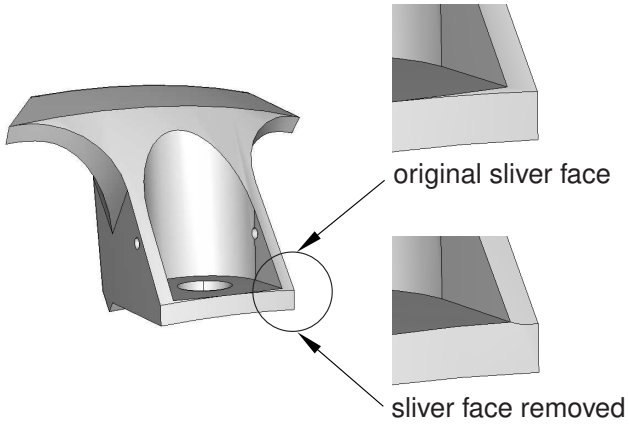


Figure 4: Removing a sliver.

Repair normals

You can repair the face normals of shell and solid imported parts, and you can perform these repairs on either manifold parts (parts in which every edge is shared by only one or two faces) and non-manifold parts. The tool has different uses for solid and shell parts.

Solid

In rare cases the Query toolset reports that the volume of an imported solid part is negative because the face normals indicate it was inside out in the CAD system from which it originated. The **Repair face normals** tool will flip the normals and turn the solid right side out.

Shell

An imported shell part can contain faces that have normals pointing in opposite directions. The **Repair face normals** tool will align all the normals on a shell part. If the face normals are already aligned, this tool will flip all the normals so that they remain aligned but point in the opposite direction.

Note:

The element normal orientation, which is specified using the **Element Normal** assignment in the Property module, specifies the relative element normal with respect to the geometry normal. These element normals will be updated using the new geometry normals after the **Repair face normals** operation.

When you perform repairs of non-manifold shell parts, you cannot align the normals for all of the faces in the part in a single operation; you must select faces individually to flip the direction of their normals. You can perform a query using the **Shell element normals** option from the **Query** dialog box to assess the orientation of the face normals before you repair your shell part.

Offset

You can create faces on a three-dimensional part by selecting the faces to be offset then specifying an offset distance or selecting target faces and a distance calculation method. The offset direction is opposite to the face

normal direction for the source face, and offset can be positive or negative. Abaqus/CAE creates the new face using the same process as that used for offsets in the Sketcher (for more information, see *Offsetting objects*).



Note:

regardless of the offset distance calculation method, Abaqus/CAE applies a constant offset to create the new face. You cannot use this method to create a face equidistant from two converging or diverging faces.

Extend 

You can extend existing faces by specifying an extension distance or selecting target faces to control the extension distance. Abaqus/CAE replaces the existing faces with the extended face feature.

Blend 

You can create new faces that blend the contours of existing edges in the model to form faces between those edges. You can choose to create the new faces by using tangency to existing faces at each edge, by calculating the shortest path between the edges, or by specifying a wire shape as the path.

Face from element faces 

You can create a new geometric face from orphan element faces. Abaqus/CAE creates a new geometric face based on the node positions of the selected element faces. Vertices are created at edge nodes where there is a significant change in the element edge direction.

Methods for editing entire parts

When you choose **Part** from the **Geometry Edit** dialog box, the **Tool** list displays the following methods for repairing entire parts:

Convert to analytical 

Abaqus/CAE tries to change the internal definition of edges, faces, and cells into a simpler form that can be represented analytically. For example, a face that is nearly planar will be converted to an equation that represents the plane. Converting to an analytical representation usually provides the following advantages:

- Processing of the part is faster.
- The converted entity is available during feature operations. For example, the extrude operation requires a planar face and a linear edge.
- The geometry is improved.
- If you subsequently need to stitch the part, the stitching operation is more likely to be successful.

Convert to precise 

Abaqus/CAE offers two methods to convert entities to precise geometry:

- If you choose **Tighten Gaps**, Abaqus/CAE attempts to improve the precision of the faces, edges, and vertices in your model. This method is faster but does not perform a full computation of the geometry.
- If you choose **Recompute Geometry**, Abaqus/CAE tries to change neighboring entities so that their geometry matches exactly. Recomputing geometry usually results in precise geometry; however, this operation can be lengthy and increases the complexity of the imported part, which means that processing of the part is slower. Moreover, if the part contains many complex surfaces, converting to a precise representation is likely to fail. If possible, you should return to the CAD application that generated the original file and increase the precision.

What is stitching?

When you select this option, Abaqus/CAE tries to join adjoining faces along their free edges. The intent of stitching is usually to create a solid part. Stitching edges usually results in valid geometry. Abaqus/CAE stitches gaps between trimmed surfaces by adjusting the underlying surfaces until the edges intersect. *Figure 1* illustrates the stitching process.

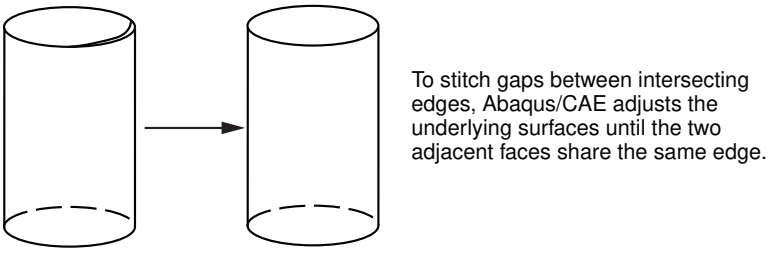


Figure 1: Stitching edges until they intersect.

Abaqus/CAE tries to recognize the edges of a solid from a set of trimmed surfaces. If the edges of each trimmed surface are close to an edge of an adjacent surface (within a user-specified tolerance), Abaqus/CAE stitches the trimmed surfaces. The end result of stitching is a B-rep solid in which the edge definition is shared between the two intersecting surfaces. For more information on B-rep solids, see *How do solid modelers represent a solid?*. Stitching fails in the presence of wires or internal faces.

When you import an IGES- or VDA-FS-format part, Abaqus/CAE stitches the part by default. However, because of the internal tolerances of IGES- and VDA-FS-format parts, the resulting representation of small features may not match the geometry that was intended in the original file.

In some cases stitching an IGES- or VDA-FS-format part results in Abaqus/CAE importing a single part that should have been imported as separate parts. If you copy the imported part to a new part, you can choose to separate disconnected regions into separate parts. For more information, see *Copying a part*.

A strategy for repairing geometry

The tools in the Geometry Edit toolset are often used to repair geometry that was imported into Abaqus/CAE. The goals of the repair operations are to create a part with no invalid entities and to create a part that has the smallest number of edges and faces to avoid impacting the mesh generation. If you are unable to create a valid part, you can ignore the invalidity. For more information, see *Working with invalid parts*. Ideally, the part will have no small edges or faces. You do not have to create a precise part, and you should continue to work with an imprecise part if possible. Parts with imprecise geometry can be meshed. However, in rare cases other operations may fail on parts with imprecise geometry; for example, other meshing functions, adding geometry features, and partitioning. If you cannot work with the imprecise part and you cannot make the part precise, you should return to the CAD application that generated the original file and increase the precision. For more information, see *What is a valid and precise part?*.

Abaqus/CAE provides many different editing tools, and it is often difficult to decide which of the tools to use and in which order to use them. In general, repairing the geometry of an imported part is an art that requires experimentation and experience. The best approach depends on the complexity of the part and how much detail you want to remain in the part when you mesh the part and analyze it.

The following guidelines will help you develop an efficient approach to repairing your parts:

Use the geometry diagnostic tools

Use the geometry diagnostic tools to query the part for invalid and imprecise entities, free edges, short edges, small faces, and sharp corner angles.

Use selection groups

You can use mouse button 3 to create temporary selection groups to make the combination of geometry diagnostic tools and repair tools easier to use. You can use the geometry diagnostic tools to highlight a region of invalid and imprecise geometry, and you can copy the region into a selection group. You can then paste the selection group into the Geometry Edit toolset when specifying the region to repair.

Use display groups

Use display groups to selectively remove faces and cells to help you see inside complex parts and to determine which entities are highlighted by the geometry diagnostic tools.

Take an incremental approach

Use the repair tools on small groups of selected faces and edges and check that the outcome of the repair process is correct before proceeding.

Keep solids intact

Try not to convert a solid into a shell by removing faces from the solid. To avoid removing faces, use the **Replace** tool for faces, and the **Repair small** tools for edges and faces.

Avoid creating wires

If your part is a solid or a shell, try not to perform operations that result in wires. It is difficult for Abaqus/CAE to reconstruct faces from wires.

Remove fillets and chamfers

Use the **Replace** tool and the **Extend neighboring faces** option to remove fillets and chamfers from your part and extend the neighboring faces to close the resulting gap.

Remove faces that meet tangentially

Use the **Replace** tool or the **Repair small** tool to repair faces that are approximately tangential. You should use the **Repair small** tool to remove faces that are relatively small. The **Repair small** tool simply extends the remaining faces until they meet. The small face is deleted in the process, although the result may be imprecise. Use the **Replace** tool for relatively large faces that meet tangentially. The **Replace** tool uses underlying points to replace the original faces with an approximate surface.

Check the validity of the part

If your part is invalid, you can check if the repair operations have made the part valid by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears. Checking the validity can be a lengthy operation, and you should wait until you have finished editing the part before you check its validity. If a part is already valid, none of the repair operations will make it invalid.

Create partitions after you repair

If your part contains partitions, the Geometry Edit toolset may delete them during a repair operation. To avoid this problem, you should wait to partition the part until after you have finished editing it.

Creating a part from orphan elements

You can use tools from the Geometry Edit toolset to create geometry from the faces of orphan mesh elements. This technique is useful when you have a meshed part that requires modifications, but you do not have some or all of its geometry available. The decision to add geometric features to an orphan mesh, recreate the part geometry from scratch, or reverse engineer geometry from the orphan elements depends on a number of factors, including:

- complexity of the existing part
- complexity of the features to be added
- degree of difficulty required to modify the existing orphan mesh
- desire to create a new mesh for the entire part

Other factors such as the time constraints for the project and the ability to export model geometry may also be involved in your decision.

The goal of creating a part from an orphan mesh is to create geometry that matches the original design intent while providing a platform for current and future modifications. The geometry must also be suitable for creation of a native mesh by either full association with the current mesh or creation of a new mesh.

The following guidelines will help you create geometry for an orphan mesh:

Create new geometric faces

Use the **Face from element faces** tool  to create the outer surface geometry for the model. Each use of the tool creates a single shell face from selected exterior orphan element faces. The tool includes several unique methods for selection of multiple orphan element boundary faces, and it automatically stitches the newly created face to any adjacent geometry. For more information, see *Create face from element faces*. When you are finished creating faces, all exterior orphan element faces should be covered by geometric faces.

Remove extra edges and faces

Use other tools from the Geometry Edit toolset to remove extra edges, small faces, and any other features that would restrict generation of a good quality mesh. If necessary, convert the part to precise geometry. For more information, see *A strategy for repairing geometry*.

Convert to solid

If you are working with a solid part, use the **Create solid from shell** tool  to fill the space inside the created shell faces.

Partition and mesh the part

Suppress or delete the orphan mesh, and create a new mesh for the part.

Create partitions after you repair

If your part contains partitions, the Geometry Edit toolset may delete them during a repair operation. To avoid this problem, you should wait to partition the part until after you have finished editing it.

Editing and repairing edges

This section describes the tools that you can use to repair edges.

In this section:

- *Stitching edges to create faces*
- *Repairing small edges*
- *Merging edges*
- *Removing redundant entities*
- *Repairing invalid edges*
- *Removing wire edges*

Stitching edges to create faces

You can create new faces by automatically stitching any gaps between free edges. Stitching gaps allows you to replace small faces, fillets, or chamfers that you previously deleted. The operation to stitch gaps is stored as a feature of the part; therefore, you can use the Model Tree to delete the stitch and revert to the original part. For more information, see *What is stitching?*.

You can stitch gaps in either manifold parts (parts in which every edge is shared by only one or two faces) or non-manifold parts.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also stitch edges using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Stitch** method.
3. From the warning dialog box that appears, click **OK**.
4. If desired, specify a stitch tolerance from the prompt area for the stitching operation. Abaqus/CAE will stitch only gaps smaller than the tolerance value that you specify.
5. Do one of the following:

   - If you want to select the edges for the stitching operation individually, toggle on **Pick edges**, click mouse button 2, then specify the edges you want to select. All the edges you select must be free edges because when Abaqus/CAE attempts only manifold stitching when **Pick edges** is selected. The stitching operation fails if you select any edges that are shared by more than one face.

     Tip: You can select edges individually, specify an existing set that includes edges, or select edges using the angle method. For more information about selecting geometry in the viewport, see *Selecting objects within the current viewport*.

   - If you want to stitch edges throughout the entire model, toggle off **Pick edges** and click mouse button 2. When **Pick edges** is toggled off, Abaqus/CAE can attempts both manifold and non-manifold stitching on the whole model. If any of the edges of the model are shared by more than one face, the resultant model will be non-manifold.

   Abaqus/CAE attempts to stitch the part according to your specifications. If the repair is successful, Abaqus/CAE prompts you to update the validity of the part.

6. If you have editing the part, you should check its validity by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Repairing small edges

You can repair selected small edges from a part. Abaqus/CAE removes the small edges and repairs the adjoining edges and faces to create a closed geometry. The operation to repair small edges is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also repair small faces using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Repair small** method.
3. Select the edges that you want to repair. You can select the edges from the viewport or select an existing set containing the edges.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one small edge to repair. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Click mouse button 2 to indicate that you have finished selecting small edges.

   Abaqus/CAE removes the selected small edges from the part and prompts you to update the validity of the part.

5. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you repair small edges.)
   - Click **No** to skip the update and continue your repairs.
   - Click **Cancel** to cancel the changes in the current repair operation.

   You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing edges*

Merging edges

You can merge selected edges from a part into a single edge. Merging edges also removes redundant vertices and edges. When you select an edge to merge, Abaqus/CAE searches for all connected edges until it encounters a branch and appends the connected edges to your selection. The operation to merge edges is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also merge edges using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Merge** method.
3. Select the edges that you want to merge. Abaqus/CAE searches for all connected edges until it encounters a branch and selects all of the connected edges.

   You can select the edges from the viewport or select an existing set containing the edges.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one edge to merge. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Click mouse button 2 to indicate that you have finished selecting edges.

   Where possible, Abaqus/CAE merges selected edges into single edges. Abaqus/CAE displays an error message if no edges can be merged; otherwise, Abaqus/CAE prompts you to update the validity of the part.

5. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you merge edges.)
   - Click **No** to skip the update and continue your repairs.
   - Click **Cancel** to cancel the changes in the current repair operation.

   You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

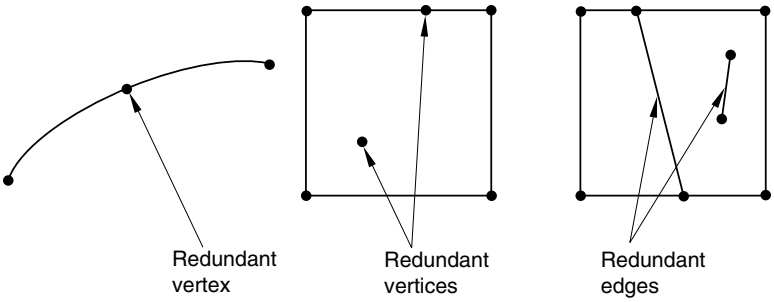
Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing edges*

Removing redundant entities

You can use the **Remove redundant entities** tool to remove redundant vertices and edges from a selected region.

An imported part can contain redundant vertices that are not attached to an edge or are positioned along a straight edge. Similarly, an imported part can include redundant edges that are not connected to a face or are internal edges. Redundant vertices and edges do not change the shape or the area of a part and are not required for a complete definition, as shown in the following figure:



The Query toolset provides a set of geometry diagnostic tools that allow you to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*. The operation to remove redundant entities is stored as a feature of the part; therefore, you can use the Model Tree to restore vertices and edges that you deleted.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: If you are using the Part module, you can also remove redundant entities using the tool, located with the edit tools in the module toolbox. For a diagram of the edit tools in the Part module toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Remove redundant entities** method.
3. Select the region from which Abaqus/CAE should remove redundant edges and vertices. You can select the region from the viewport or select an existing set containing the region.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. From the prompt area, toggle off **Remove redundant edge vertices** if you do not want to remove vertices.
5. From the prompt area, click **Done**.

   Abaqus/CAE removes the redundant entities from the selected region and prompts you to update the validity of the part.

6. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you remove redundant entities.)
   - Click **No** to skip the update and continue your edits.

- Click **Cancel** to cancel the changes in the current editing operation.

You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing edges*

Repairing invalid edges

In rare cases after you import a part, Abaqus will report that all of its edges are invalid. The **Repair invalid** tool attempts to repair the invalid edges by recomputing the data that define them, which can help repair some types of invalidity. You should also use this tool if only invalid edges remain after automatically repairing a part.

The Query toolset provides a set of geometry diagnostic tools that determine if a part contains invalid edges. For more information, see *Using the Query toolset in the Part module*. The operation to repair invalid edges is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also repair invalid edges using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Repair invalid** method.

   Abaqus/CAE repairs invalid edges and prompts you to update the validity of the part.

3. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you repair invalid edges.)
   - Click **No** to skip the update and continue your repairs.
   - Click **Cancel** to cancel the changes in the current repair operation.

   You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing edges*

Removing wire edges

You can remove selected wire edges from a part. Wire edges are wires, as opposed to edges of a solid or shell feature. The operation to remove wire edges is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also remove wire edges using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Edge** category and the **Remove wire** method.
3. Select the wire edges that you want to remove. You can select the edges from the viewport or select an existing set containing the edges.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one wire edge to remove. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Click mouse button 2 to indicate that you have finished selecting wire edges.

   Abaqus/CAE removes the wire edges and prompts you to update the validity of the part.

5. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you remove wire edges.)
   - Click **No** to skip the update and continue your repairs.
   - Click **Cancel** to cancel the changes in the current repair operation.

   You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing edges*

Editing and repairing faces

This section describes the tools that you can use to repair faces.

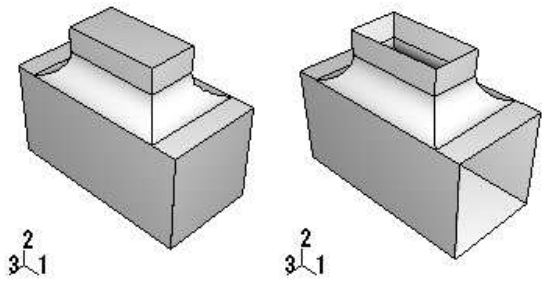
In this section:

- *Removing faces*
- *Covering edges with a new face*
- *Replace faces*
- *Repairing small faces*
- *Repairing a sliver*
- *Repairing face normals*
- *Offset faces*
- *Extend faces*
- *Blend faces*
- *Create face from element faces*

Removing faces

To make a part valid or precise, you can use the remove faces tool to remove selected faces from a three-dimensional solid or shell or from a two-dimensional planar part.

After you have selected the faces to remove, Abaqus/CAE looks for adjacent faces that define a feature. You can remove the entire feature, or you can remove only the selected faces. When you remove one or more faces from a solid part, Abaqus/CAE converts all of the cells in the part to a shell, as shown in the following figure:



The Query toolset provides a set of geometry diagnostic tools that allow you to locate areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*. The operation to remove faces is stored as a feature of the part; therefore, you can use the Model Tree to restore faces that you deleted.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also remove faces using the tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Remove** method.
3. Select the faces that you want to remove. You can select the faces from the viewport or select an existing set containing the faces.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to remove. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Click mouse button 2 to indicate that you have finished selecting faces.

   Note:

   If you are removing one or more faces from a solid part, Abaqus/CAE indicates that removing the faces will result in the solid being converted to a shell and prompts you to continue.

Abaqus/CAE removes the selected faces.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Covering edges with a new face

You can create a face by selecting the edges that the new face will cover. The selected edges must form a closed loop. Abaqus/CAE creates the new face as a shell. If the new shell forms a closed part, you can use the solid-from-shell tool to convert the part to a solid. For more information, see *Creating a solid feature from a shell*. The cover edges operation is stored as a feature of the part; therefore, you can use the Model Tree to delete faces that you created. Abaqus/CAE creates the new face by projecting a face on the plane formed by the selected edges. Therefore, you should not use the **Cover edges** tool if the resulting face would be very convex.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also create a face using the tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Cover edges** method.
3. Select the edges that form a loop and define the edges of the new face. You can select the edges from the viewport or select an existing set containing the edges. You can select more than one loop, and Abaqus/CAE will create more than one face. However, the loops should not be concentric.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. From the prompt area, click **Done**.

   Abaqus/CAE creates the new face.

5. If you have finished editing the part, you should check its validity by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Replace faces

If faces from an imported part do not appear to have been formed correctly, you can replace the faces with a new face generated by Abaqus/CAE based on the underlying geometry. Because Abaqus/CAE relies on the underlying geometry, you should not use the **Replace** tool if the resulting face would be very convex. If you select more than one face to replace, the faces must be connected.

Alternatively, you can replace a face by extending neighboring faces. This tool is useful for removing bosses or small bumps from your parts.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also replace faces using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Replace** method.
3. Select the faces that you want to replace. The faces must be connected. You can select the faces from the viewport or click mouse button 3 to paste faces into your selection from a temporary selection group (for details, see *What is a selection group?*).

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to replace. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. From the prompt area, toggle on **Extend neighboring faces** to replace the selected faces with a single face that is formed by extending neighboring faces.
5. Click mouse button 2 to indicate that you have finished selecting faces.

   Abaqus/CAE replaces the selected faces.

6. If you have finished editing the part, you should check its validity by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Repairing small faces

You can repair selected small faces of a part. Abaqus/CAE removes the small faces and repairs the adjoining edges and faces to create a closed geometry. The operation to repair small faces is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also repair small faces using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Repair small** method.
3. Select the faces that you want to repair. You can select the faces from the viewport or select an existing set containing the faces.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one small face to repair. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Click mouse button 2 to indicate that you have finished selecting small faces.

   Abaqus/CAE repairs the small faces and prompts you to update the validity of the part.

5. Do one of the following:

   - Click **Yes** to update the validity of the part. (You should update the validity of the part whenever you repair small faces.)
   - Click **No** to skip the update and continue your repairs.
   - Click **Cancel** to cancel the changes in the current repair operation.

   You can also update the validity of a part at any time by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Repairing a sliver

You can think of a sliver as a piece of extra material in a face. You can repair an unwanted sliver and remove it from a face of a three-dimensional solid or shell or from a two-dimensional planar part. You must select the face containing the sliver to repair and two points from the face. Abaqus/CAE uses the two points to divide the selected face into two regions. It then deletes the smaller region (the sliver) and retains the larger region. If the smaller region is relatively large compared with its neighbors, the operation fails. The operation to repair a sliver is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also remove a sliver using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Repair sliver** method.
3. Select the face that you want to repair. You can select the face from the viewport or select an existing set containing the face.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   Tip: If you are unable to select the desired face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. Select a point that defines a corner of the sliver.
5. Select a second point that defines the opposite corner of the sliver.
6. Abaqus/CAE draws a line between the two selected points and removes the resulting small face.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Repairing face normals

You can repair the face normals of shell and solid imported parts, and you can perform these repairs on either manifold parts (parts in which every edge is shared by only one or two faces) or non-manifold parts.

If Abaqus reports that the volume of an imported part is negative, it means that the solid has been turned inside out. The **Repair normals** tool will flip the normals and turn the solid right side out.

An imported shell part can contain faces that have normals pointing in opposite directions. The **Repair normals** tool will align all the selected normals on a shell part. If the face normals are already aligned, this tool will flip all the selected normals so that they remain aligned but point in the opposite direction.



Note:

Abaqus/CAE does not flip normals that have been assigned to shell faces manually using the **Element Normal** assignment option in the Property module.

When you perform repairs of non-manifold shell parts, you cannot flip the normals for all of the faces in the part in a single operation; you must select each face individually to flip the direction of its normal. Performing a query using the **Shell element normals** option from the **Query** dialog box can help you assess the orientation of the face normals before you repair your shell part.

The Query toolset provides a set of geometry diagnostic tools that calculate the volume of a solid part and display the normals of shells and membranes. For more information, see *Using the Query toolset in the Part module*. The operation to repair face normals is stored as a feature of the part; therefore, you can use the Model Tree to delete or suppress the operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also repair face normals using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Repair normals** method.
3. Select one of the following options from the prompt area:
   - Click **Entire Part** to repair face normals for every incorrectly oriented face in your part.
   - Click **Select Faces** to select individual faces in your part.
4. If you are selecting individual faces in your part, choose the face normals that you want to repair from the viewport, then click **Done** from the prompt area.

   You can use a combination of drag-select, **[Shift] + Click**, and **[Ctrl] + Click** to select faces in your part, and you can also use the angle method or the face curvature method to select faces. For more information, see *Selecting objects within the current viewport*.

   Abaqus/CAE repairs the face normals.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Offset faces

You can create faces by offsetting them from existing faces. Abaqus/CAE creates the new faces as a shell. The offset operation is stored as a feature of the part; therefore, you can use the Model Tree to delete faces that you created. You can use offset faces when creating a midsurface model; for information, see *Midsurface modeling*."

You can use target faces to compute an offset distance as shown in *Figure 1*, or you can specify an offset distance.

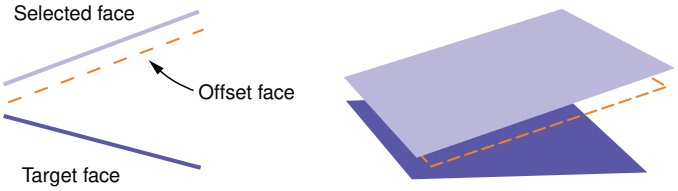


Figure 1: A constant offset distance between diverging faces.

Regardless of the offset distance method that you choose, Abaqus/CAE creates all new faces at a constant offset distance from the selected faces, as shown in the figure. However, if you use target faces to compute the offset distance, Abaqus/CAE uses the selected faces and the target faces to calculate the thickness and offset distance at each node in the offset faces such that the behavior of the offset faces approximates the behavior of the original sections. You can view the result of these calculations after the part is meshed by showing the native mesh and toggling on **Render shell thickness** with a **Scale factor** of 1 in the **Part Display Options** dialog box.

You can review and edit the shell thickness values with the **Assign thickness and Offset** tool (for more information, see *Assigning thicknesses and offsets*), or you can specify a thickness for the new faces when you assign a section to them.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also offset faces using the tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Offset** method.
3. Select the faces to offset. You can select faces individually, by face angle, or by face curvature; and you can select an existing set containing the faces. For more information on selecting objects in the viewport, see *Selecting objects within the current viewport*.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. From the prompt area, click **Done**.

   Abaqus/CAE displays the **Offset Faces** dialog box.

5. Choose one of the available options to determine the offset distance:

   Note:

   Regardless of the calculation method, Abaqus/CAE applies a constant offset to the faces as shown in *Figure 1*. You must use another method to create new faces centered between converging or diverging faces.

Use target faces to compute distance

Select a target face or faces. Abaqus/CAE computes the distance between the faces to be offset and the target faces by sampling the faces at several points. Use one of the following distance options:

- Select **Half the average distance** to offset the new faces by this amount.
- Select **Fraction distance to closest point on face**, and enter a value.

  Abaqus/CAE offsets the new faces by the distance to the closest point on the target faces times the entered value.
- Select **Fraction distance to farthest point on face**, and enter a value.

  Abaqus/CAE offsets the new faces by the distance to the farthest point on the target faces times the entered value.

You can select the target faces using the same methods you used in Step 3. If your original faces are part of the reference representation for a midsurface model, you can click **Auto Select** to have Abaqus/CAE select appropriate faces from the opposite side of the reference representation. Varying distances between potential target faces may result in selections that only partially match the original faces. If necessary, click **Edit** to modify the selections in the viewport.

Distance

Enter an offset distance, or select  to measure and enter a distance between objects in the viewport.

Negative values for the fraction or actual distance create an offset in the face normal direction; positive values create an offset in the opposite direction. If you are uncertain of the normal directions, use the **Shell element normals** general query to view the normal directions for shells in the model. For more information, see *Obtaining general information about the model*. The normals for solids point toward the outside, so a positive offset will create a face inside the part unless the offset distance is greater than the part thickness.

6. If the model contains a reference representation, toggle **Auto trim to reference representation** to control whether Abaqus/CAE matches the offset faces to the reference representation.

   If this option is toggled on, Abaqus/CAE extends the offset faces along their free edges and trims them where they intersect with the reference representation.

   Note:

   Face extension may fail for complex faces. If face extension fails, Abaqus/CAE indicates the error and trims any unextended faces that intersect with the reference representation.

7. Click **OK**.

   Abaqus/CAE creates new faces at the prescribed distance from the originals, and the procedure restarts at Step 3.

8. To exit the offset procedure, either

   - click the cancel button  in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or

- select another operation from the Geometry Edit toolset or from the tools in the Part module.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Using the offset face tool for midsurface modeling*
- *Methods for editing faces*

Extend faces

You can edit faces by extending existing faces. The extend operation is stored as a feature of the part; therefore, you can use the Model Tree to edit the distance parameter if the distance method is used or to delete the extended faces.

Figure 1 shows the use of target faces to extend a face. The top images show the complete part, which includes a central planar face separated from the surrounding faces by a small gap. The surrounding faces enclose about two and a half sides of the planar face. The planar face is selected for extension, and the surrounding faces are used as target faces. The selected face is extended along the three sides that would intersect with the target faces. The two lower images detail how you can control the extension where the selected face extends beyond the end of the target face on the left side. In the lower left image the **Trim to extended underlying target surfaces** option was used such that the entire left edge of the selected face is extended uniformly—as if the target face extended to the full height of the selected face. In the lower right image the face is extended only up to where the target face ends.

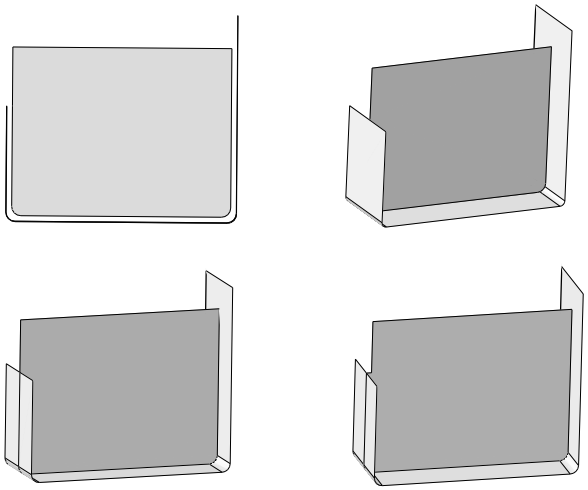


Figure 1: The effect of toggling the Trim to extended underlying target surfaces option when extending a face: initial model with gap (top left and right), on (lower left), and off (lower right).

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also extend faces using the tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Extend** method.

   Abaqus/CAE displays the **Extend Faces** dialog box.

3. Choose one of the following methods to extend faces:

   - Toggle on **Specify edges of faces to extend**, and pick one or more edges to extend the faces that are bounded by those edges along only the selected edges.
   - Toggle on **Specify faces to extend**, and pick faces to extend them along all their free edges.

   Click **Select** to specify the edges or faces to be extended.

You can select edges individually or by edge angle; and you can select faces individually, by face angle, or by face curvature. For more information on selecting objects in the viewport, see *Selecting objects within the current viewport*. You can also select an existing set of edges or faces.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

4. From the prompt area, click **Done**.
5. Choose one of the available methods to determine the extension distance:

Distance

Enter an extension distance, or select  to measure and enter a distance between objects in the viewport.

Up to target faces

Select a target face or faces to limit the extension.

Abaqus/CAE highlights the target faces in yellow.

If desired, toggle on **Trim to extended underlying target surfaces** to have Abaqus/CAE temporarily extend the target faces to create intersections with the faces that you want to extend.

Up to reference representation

Abaqus/CAE uses the faces of the reference representation to limit the extension. This option is not available if the model does not contain a reference representation.

6. Click **OK**.

   If you selected target faces in Step 5 that do not completely surround the faces being extended, Abaqus/CAE indicates the edges that will intersect with the target faces—a subset of your selection from Step 3—and gives you the option to extend only these edges instead of your original selection.

   Abaqus/CAE creates the extended face feature. If you used target faces or a reference representation to limit the extension, Abaqus/CAE trims the extended faces where they intersect with the limiting faces. The extended face feature replaces the original faces in the viewport, and the procedure restarts at Step 3 using the most recently used selection method.

7. To exit the extend procedure, either

   - click the cancel button  in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or
   - select another operation from the Geometry Edit toolset or from the tools in the Part module.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Using the extend faces tool for midsurface modeling*
- *Methods for editing faces*

Blend faces

You can create blended faces to join existing faces in the model.

Abaqus/CAE closes the space between two sets of connected edges with new faces. You can choose whether to make the new faces tangent to the selected edges, connect the edges using the shortest path between them, or specify a path that the new faces must follow between the selected edges. *Figure 1* shows two faces with the selected edges highlighted. In this example, the tangent method creates a face with a smooth radius tangent to both original faces, the shortest path method creates a planar face, and the selected path method creates a face with the profile of the selected wire edge.

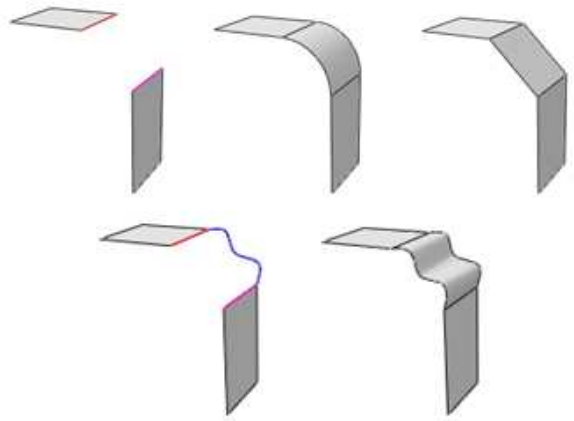


Figure 1: Creating blended faces using the tangent, shortest path, and specify path methods.

You can also create a blended face to join edges and faces. Abaqus/CAE creates blended faces between the selected edges and faces using the shortest path method. When you create a blended face feature between edges and an existing face, the new faces are always perpendicular to the existing face. The free edges are extended or shortened as needed to intersect with the edges of the face. *Figure 2* shows the selected edges and the feature created by blending between them and a planar face. The ends of the center edge are a perpendicular projection from the planar face, and the outer ends of the left and right edges are extended to meet the edges of the planar face.

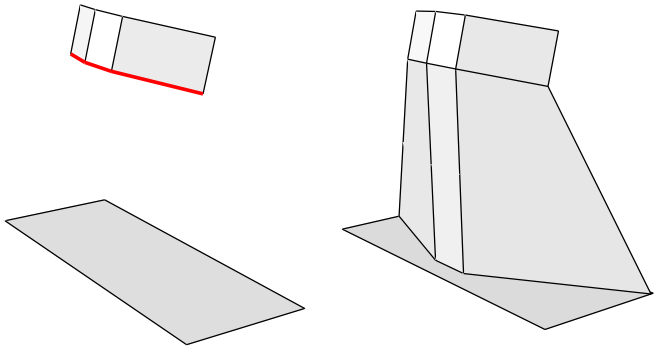


Figure 2: Creating a blended face between edges and a face.

The blend operation is stored as a feature of the part; therefore, you can use the Model Tree to delete faces that you created.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.



Tip: You can also blend faces using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **Blend** method.
3. Select edges for the first side of the blended face.

   You can select edges individually or by edge angle. For more information on selecting objects in the viewport, see *Selecting objects within the current viewport*. You can also select an existing set of edges.

   Note:

   The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Sets** on the right side of the prompt area.

   From the prompt area, click **Done**.
4. Select edges or faces for the second side of the blended face.

   If you select faces for the second side, the edges selected in Step 3 must have a perpendicular projection onto the faces or Abaqus/CAE will not be able to create a blended face for the selections.

   From the prompt area, click **Done** to complete your selection and to open the **Blend Faces** dialog box.
5. Choose one of the available blend methods:

Tangent

Abaqus/CAE will create blended faces tangent to the faces containing the first and second side edges. To use this method, the edges selected in Step 3 and Step 4 must be free edges—edges that belong to only one feature face—so that Abaqus/CAE can calculate tangency.

Shortest path

Abaqus/CAE will create blended faces that connect the first and second side edges using the shortest path between them. If you selected a face or faces for the second side of the blend, this is the only blend method available.

Specify path

Specify a path between the edges. Click **Select** to choose an existing edge or wire feature that joins the end of one of the first side edges to the end of one of the second side edges. Abaqus/CAE will create blended faces using the profile of the selected path.

6. Click **Preview** to preview the blended face that would result from the current selections.
7. Click **OK**.

   Abaqus/CAE creates the blended face feature. The blend procedure restarts at Step 3 using the most recently used selection method.
8. To exit the blend procedure, either

   - click the cancel button  in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or

- select another operation from the Geometry Edit toolset or from the tools in the Part module.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing faces*

Create face from element faces

You can create a geometric face from orphan element faces in a part. Abaqus/CAE creates a new geometric face based on the nodal positions of the selected element faces. Vertices are created at edge nodes where there is a significant change in the element edge direction. The new face appears like other geometric faces in the model, and Abaqus/CAE stitches the new face to adjacent faces in the part. By default, the new face is associated with the mesh; you have the option to change this behavior. In the Model Tree, the new face appears as a **Face from mesh** feature. You cannot change the group of element faces that comprise the geometric face once it has been created; however, you can delete the face and create a new one, and you can stitch faces that were not stitched together in the original creation of the geometric face.

The new geometric face shares space with the existing orphan element faces in the model. This may result in some areas appearing with the gray unmeshed geometry coloring and others appearing with the dark green and visible element edges of the orphan mesh. This is similar to the appearance of a bottom-up meshed region, where the tan region coloring and the cyan mesh coloring appear together. You can suppress the orphan mesh in the Model Tree to view only the new geometry.

When creating geometry from orphan mesh faces, keep in mind your intended use for the resulting geometry. You want to create faces that will be a good basis for modification and meshing. For example, when you select orphan element faces, your selection should end at any sharp corners or other features so that the geometric face will also end there, creating a logical feature edge. If you do select element faces that span sharp corners, Abaqus/CAE will create a single geometric face that includes the corner—this geometry may be difficult or impossible to mesh using the available top-down meshing techniques.



Note:

When you are working with solid orphan elements, selections that include element faces on both sides of a corner (multiple selections from the same orphan element) are not acceptable for creation of a single geometric face.

When Abaqus/CAE creates a geometric face, it stitches together small gaps between edges and small gaps between analytical surfaces by default. You can customize the tolerance values that determine whether small gaps will be stitched in either case, and you can turn off stitching completely for small gaps between edges. Deferring most or all of the stitching until late in the modeling process can be a more efficient modeling option because each stitching operation is processor intensive. Fitting of analytical surfaces cannot be deferred; you must perform this step before the new geometric face is created.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also create geometric faces from element faces using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Face** category and the **From element faces** method.
3. Select element faces for the new geometric face.

   You can select element faces individually, by angle, by limiting angle, by layer, or by analytic shape. For more information on selecting objects in the viewport, see *Selecting objects within the current viewport*. You can also select an existing surface. You cannot select element faces from two-dimensional or axisymmetric parts.



Note:

The default selection method is based on the selection method you most recently employed. To revert to the other method, click **Select in Viewport** or **Surfaces** on the right side of the prompt area.

4. If you want to control stitching, analytical surface fitting, and the mesh association options, click **Options** in the prompt area, and do any of the following from the dialog box that appears:

   - Toggle on **Stitch with tolerance** to apply stitching of nodes in the geometric face you create, and specify the stitching tolerance value.
   - In the **Fit analytic surfaces with tolerance** field, specify the stitching tolerance value you want to use for fitting together analytical surfaces in your selection.
   - By default, Abaqus/CAE associates the geometric face with the mesh. To change this behavior, toggle off **Associate face with mesh**.

   Click **OK** to close the dialog box.

5. From the prompt area, click **Done**.

   Abaqus/CAE creates the geometric face feature. The procedure restarts at Step 3 using the most recently used selection method.

6. To exit the procedure, either

   - click the cancel button  in the prompt area, or
   - click mouse button 2 anywhere in the Abaqus/CAE window, or
   - select another operation from the Geometry Edit toolset or from the tools in the Part module.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Creating a part from orphan elements*
- *Methods for editing faces*

Editing parts

This section describes the tools that you can use to repair entire parts.

In this section:

- *Converting to analytical*
- *Converting to precise*

Converting to analytical

When you convert a part to its analytical representation, Abaqus/CAE tries to change the internal definition of edges, faces, and cells into a simpler form that can be represented analytically. For example, a plane that is nearly planar will be converted to an equation that represents the plane. Converting to an analytical representation usually provides the following advantages:

- Processing of the part is faster.
- The converted entity is available during feature operations.
- The geometry is improved.
- If you subsequently need to stitch edges, the stitching operation is more likely to be successful.

After the operation is complete, you can use the geometry diagnostic tools in the Query toolset to locate any areas of invalid and imprecise geometry. For more information, see *Using the Query toolset in the Part module*.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also convert to analytical using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Part** category and the **Convert to analytical** method.
3. From the warning dialog box that appears, click **OK**.

   Abaqus/CAE tries to convert the part to its analytical representation.
4. If you have finished editing the part, you should check its validity by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing entire parts*

Converting to precise

When you convert a part to precise, Abaqus/CAE tries to change neighboring entities so that their geometry matches exactly. Converting to a precise representation usually reduces the number of imprecise entities in a part, though for some parts the conversion can increase the number of imprecise entities. Even when the number of imprecise entities increases, the maximum tolerance of the healed part will still usually decrease; however, you should consider suppressing or deleting the convert to precise operation when the number of imprecise entities increases.

The convert to precise operation can be a lengthy one, and it can increase the complexity of the imported part; so subsequent processing of the part may be slower. Invalid entities may also result from this operation.

1. From the main menu bar, select **Tools->Geometry Edit**.

   Abaqus/CAE displays the **Geometry Edit** dialog box.

   Tip: You can also convert to precise using the  tool, located with the edit tools in the Part module toolbox. For a diagram of the edit tools in the toolbox, see *Editing techniques*.

2. From the dialog box, select the **Part** category and the **Convert to precise** method.
3. From the warning dialog box that appears, click **OK**.
4. Select one of the following methods for the conversion:

   ### Tighten Gaps

   Abaqus/CAE attempts to reduce the gaps between the nodes and edges in your part but does not perform a full computation of the geometry. This option provides a faster but less complete conversion.

   ### Recompute Geometry

   Abaqus/CAE tries to change neighboring entities so that their geometry matches exactly. Recomputing geometry usually results in precise geometry; however, this operation can be lengthy and increases the complexity of the imported part, which means that processing of the part is slower. Moreover, if the part contains many complex surfaces, converting to a precise representation is likely to fail. If possible, you should return to the CAD application that generated the original file and increase the precision.

   After you select a method, Abaqus/CAE tries to convert the part to its precise representation. If the conversion is successful, Abaqus/CAE prompts you to update the validity of the part.

5. If you have finished editing the part, you should check its validity by clicking mouse button 3 on the part in the Model Tree and selecting **Update validity** from the menu that appears.

Additional information

- *Valid parts, precise parts, and tolerance*
- *Methods for editing entire parts*

The Partition toolset

You use the Partition toolset to divide a part or assembly into regions.
Regions are used throughout the modeling process; for example, to indicate the location of a load, a change in material properties, or a mesh boundary.

This chapter explains how you use the Partition toolset to create and position partitions on an edge, a face, or a cell.

In this section:

- *Understanding the role of partitions*
- *Using the Partition toolset*
- *Understanding partitions*
- *Partitioning techniques*
- *Partitioning edges*
- *Partitioning faces*
- *Partitioning cells*

Understanding the role of partitions

As you proceed through the modeling process, you may find that you need to select a particular region that does not exist in your model. Such regions might be used to define material boundaries, indicate the location of loads and constraints, and help refine your mesh. You use the Partition toolset to partition your model into regions.

Partitions can be created on edges, faces, and cells. A partition along an edge creates a new vertex, while partitions through faces and cells create new edges and faces, respectively. In all cases, partitioning serves to subdivide the geometry being partitioned. *Figure 1* illustrates partitioning an edge, a face, and a cell.

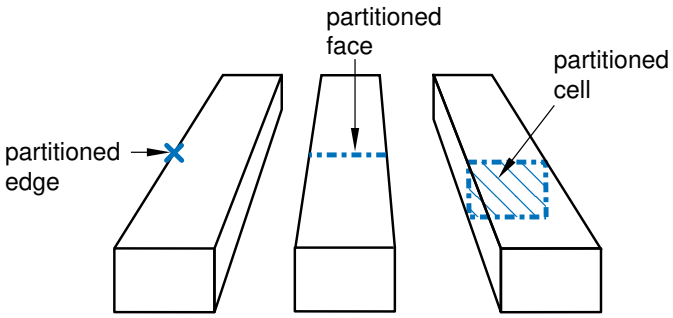


Figure 1: Partitioning an edge, a face, and a cell.

You partition edges, faces, and cells by defining partitions that refer to existing geometry. You can partition a part in either the Part module or the Property module, or you can partition an assembly in the modules that operate on the assembly. For example, you can partition a face of the assembly in the Mesh module, and you can seed the resulting internal edge to refine your mesh. A partition is a feature; and, like all features, it can be edited, deleted, suppressed, resumed, and queried. Similarly, a partition is regenerated when the assembly or a part is regenerated.

Using the Partition toolset

You can access the Partition toolset by selecting **Tools->Partition** from the menu bar. The **Create Partition** dialog box appears, and you choose the type of geometry to partition—edge, face, or cell—from the buttons in the **Type** region at the top of the dialog box. The **Method** list changes to reflect the partitions you can create. Select the desired partition tool from the **Method** list, and follow the prompts in the prompt area to create the partition. Most of the tools allow you to partition multiple edges, faces, or cells in one operation. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select the edges, faces, or cells to partition. For more information, see *Selecting objects within the current viewport*.

Partitioning techniques, provides an overview of the methods available for each type of partition. More information on creating partitions on parts and assemblies is provided in *Using the Partition toolset in the Part module* and *Partitioning the assembly*.

You can also access the Partition toolset from the module toolbox; *Figure 1* shows the hidden icons for all the partition tools in the module toolboxes.

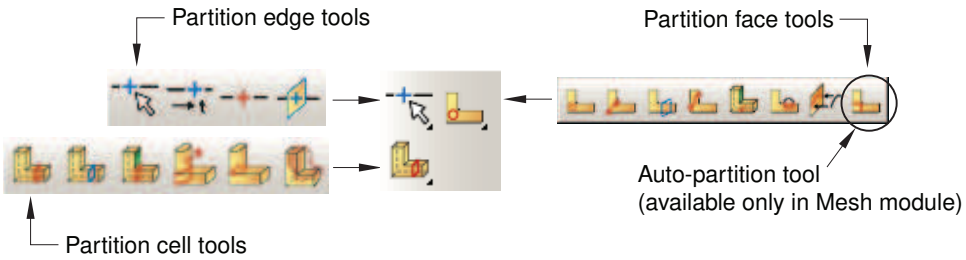


Figure 1: The partition tools.

To see a tooltip containing a short description of each partition tool, hold the mouse over the tool for a moment. For more information, see *Using toolboxes and toolbars that contain hidden icons*.

Understanding partitions

This section describes basic concepts you should understand before using the Partition toolset.

In this section:

- *Why partition?*
- *Partitions as features*

Why partition?

Deciding when to use the Partition toolset depends on how you will use the resulting regions, as described in the following list:

Partitioning parts

Use the Partition toolset in the Part or Property module to divide a part into regions. You might use the resulting regions in the Property module to apply section definitions, which include cross-sectional geometry information as well as material property information. Partitions created in the Part module and Property module are associated with the part and are available in every instance of the part in the assembly; however, you cannot modify or delete the partitions in a part instance.

Partitioning the assembly

Use the Partition toolset in the assembly-related modules to divide the part instances in the assembly into regions. You might use the resulting regions to do the following:

- Apply loads, boundary conditions, and predefined fields to regions.
- Associate an output request with regions.
- Gain finer control over the meshing process by adding internal edges that can be seeded.
- Subdivide complex three-dimensional regions into simpler regions that the automatic mesh generator can mesh.

Partitions created in an assembly are associated only with the assembly, not with the original parts, and are available in all the modules that operate on the assembly. They will not appear in other instances of the same part, and they are not available in the Part or Property module.

Additional information

- *Understanding partitions*
- *Partitioning techniques*

Partitions as features

The geometry of each part is constructed from a set of features; a partition is simply an additional geometric feature that you add to the part. Because partitions are features, they can be edited, deleted, suppressed, and resumed with the Feature Manipulation toolset.

If you create a partition and then modify the geometry of the underlying part or assembly, Abaqus/CAE regenerates the partition along with all the other features. Furthermore, the geometry of the regenerated partition is dependent on the method you used to create the partition. The following example illustrates that changes in underlying geometry can cause partitions to move or to change shape when the assembly is regenerated.

1. The user partitions a face using a line between two selected points (the centers of the two circles), as shown in *Figure 1*.

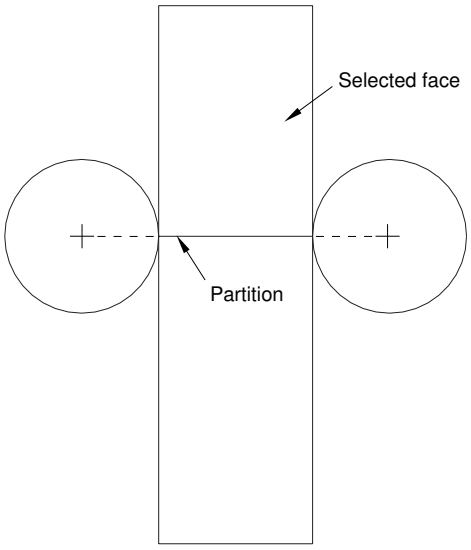


Figure 1: A partition between two selected points.

2. The user modifies the assembly, and the position of the selected points that defined the partition changes. When Abaqus/CAE regenerates the assembly, the partition is still defined as the line between the center of the two circles, as shown in *Figure 2*.

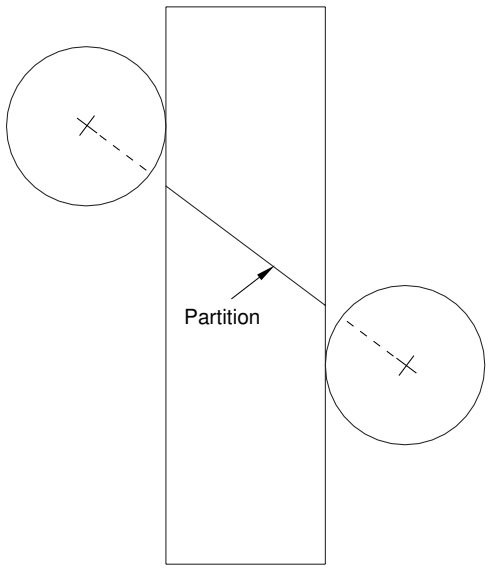


Figure 2: The partition after regeneration.

You can use the Feature Manipulation toolset to make limited modifications to some partitions:

- You can enter a parameter to partition an edge directly and to position a Bézier curve that partitions a face. The parameter must be between zero and one and represents a fraction of the length of an edge; the Feature Manipulation toolset allows you to edit the value you provided.
- You can sketch a partition on a face; the Feature Manipulation toolset allows you to edit the sketch.

If you need to change a partition that you cannot modify with the Feature Manipulation toolset, you must delete the partition and recreate it.

Additional information

- *Modifying and manipulating features*
- *Understanding partitions*
- *Partitioning techniques*

Partitioning techniques

This section provides an overview of the different partitioning techniques.

In this section:

- *Methods for partitioning edges*
- *Methods for partitioning faces*
- *Methods for partitioning cells*

Methods for partitioning edges

Select **Tools->Partition** from the main menu bar to display the **Create Partition** dialog box.
When you choose **Edge** from the **Create Partition** dialog box, the **Method** list displays the following methods for partitioning edges:

Specify parameter by location

Pick a point anywhere along the edge. For detailed instructions, see *Using the specify parameter by location method to partition edges*.

Enter parameter

Enter a parameter in the prompt area, as shown in *Figure 1*. An arrow along the edge indicates the direction of increasing parameter value from the start vertex (corresponding to an edge parameter value of zero) to the end vertex (corresponding to a value of one). For detailed instructions, see *Using the enter parameter method to partition edges*.

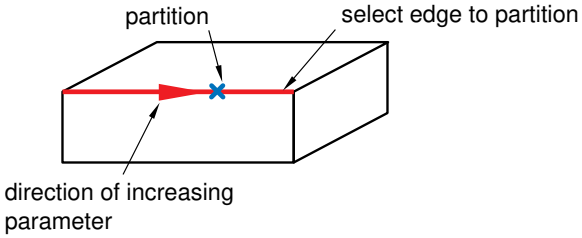


Figure 1: Entering a parameter to partition an edge.

Select midpoint/datum point

Select the midpoint of the edge or a datum point along the edge, as shown in *Figure 2*. For detailed instructions, see *Using the pick midpoint/datum point method to partition an edge*.

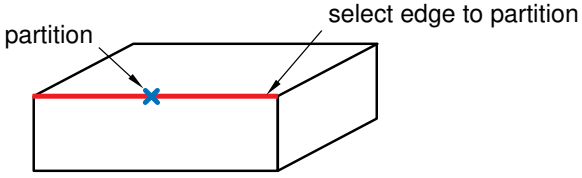


Figure 2: Selecting the midpoint or a datum to partition an edge.

Use datum plane

Select a datum plane. Abaqus/CAE creates the partition where the datum plane intersects the edges, as shown in *Figure 3*. Partitioning a group of selected edges with a datum plane is a useful technique for aligning a group of partitions. For detailed instructions, see *Using the datum plane method to partition edges*.

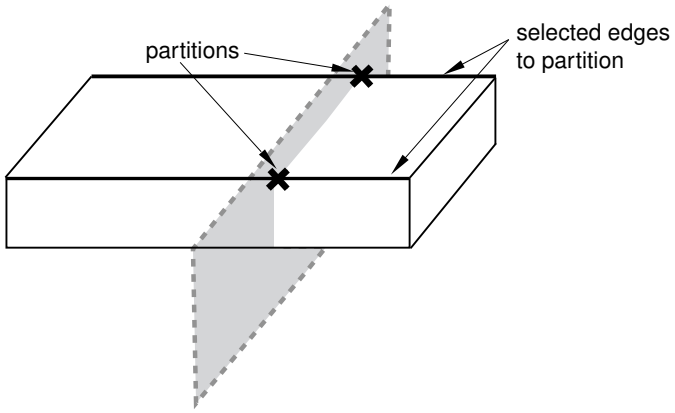


Figure 3: Selecting a datum plane to partition edges.

Additional information

- *Partitioning techniques*
- *Partitioning edges*
- *Understanding partitions*

Methods for partitioning faces

When you choose **Face** from the **Create Partition** dialog box, the **Method** list displays the following methods for partitioning faces:

Sketch planar partition

Partition a selected face by sketching a partition with the Sketcher, as shown in *Figure 1*. For detailed instructions, see *Using the sketch method to partition faces*.

You can sketch directly on the face to be partitioned, or you can sketch on a second face or datum plane and then project the sketch onto the face that you want to partition. For an example of projecting a sketch from a datum plane, see *Using the Datum toolset in the Part module*.

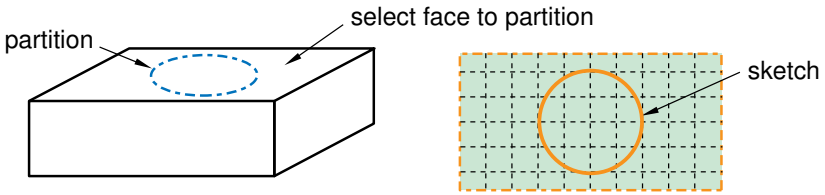


Figure 1: Partitioning a face using the Sketcher.

Shortest path between 2 points

Partition the face along the shortest path connecting two selected points; the resulting partition will be curved if the face being partitioned is curved, as shown in *Figure 2*. You can select points that are not associated with the face being partitioned; for example, the points can be located on a different face or even a different part instance. For detailed instructions, see *Using the shortest path method to partition faces*.

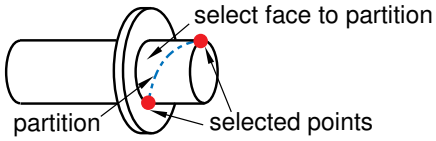


Figure 2: Partitioning a face using the shortest path between two points.

Use datum plane

Partition a face using the intersection with the extension of a datum plane, as shown in *Figure 3*. For detailed instructions, see *Using the datum plane method to partition faces*.

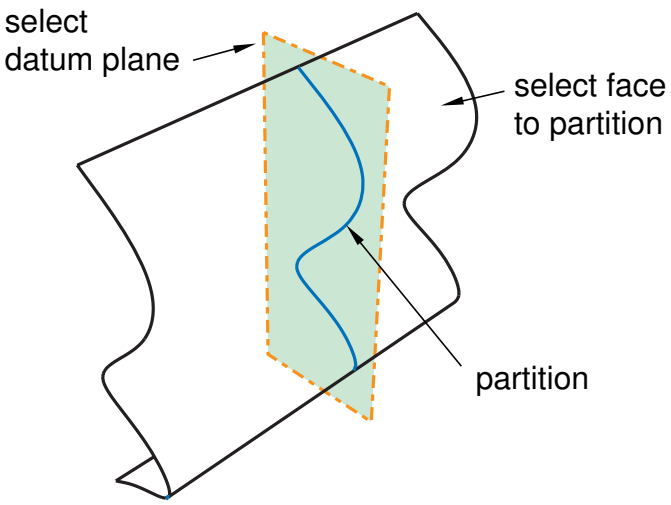


Figure 3: Partitioning a face using a datum plane.

Curved path normal to 2 edges

Partition the face along a Bézier curve that is normal to two of the face's edges, as shown in *Figure 4*. Position the curve by selecting two points anywhere along the two edges. The arc subtended by the two edges must be less than 180°. For detailed instructions, see *Using the curved path method to partition a face*.

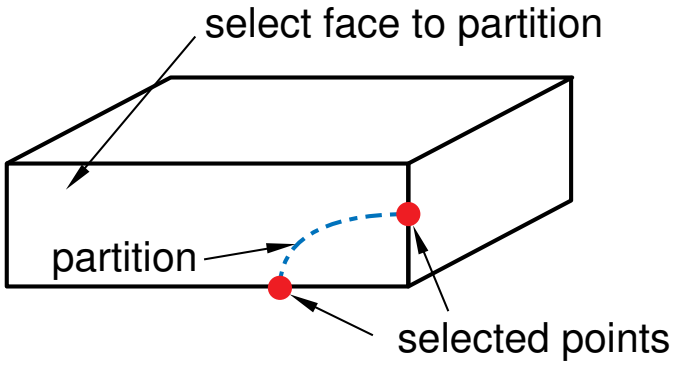


Figure 4: Partitioning a face using a Bézier curve.

Extend another face

Partition the face using the intersection with the extension of another face, as shown in *Figure 5*. The face being extended can be either planar, cylindrical, conical, or spherical; and it need not belong to the part containing the face to be partitioned. For detailed instructions, see *Using the extended face method to partition faces*.

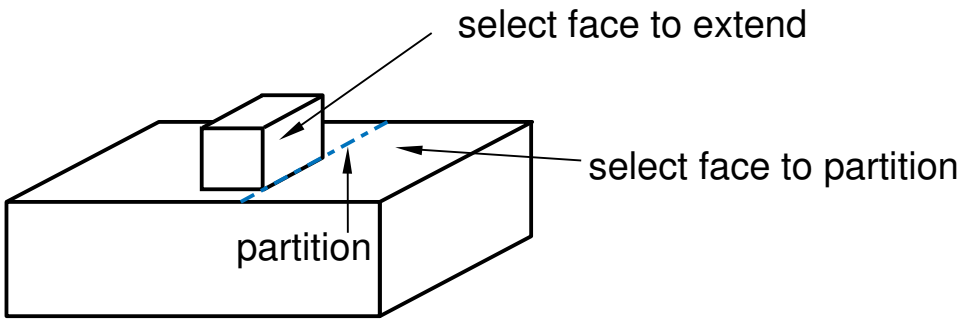


Figure 5: Partitioning a face using the extension of another face.

Intersect by other faces

Partition the face using the intersection of the target face with one or more other faces, as shown in *Figure 6*. The faces can be intersecting or tangential. For detailed instructions, see *Using the intersection method to partition faces*.

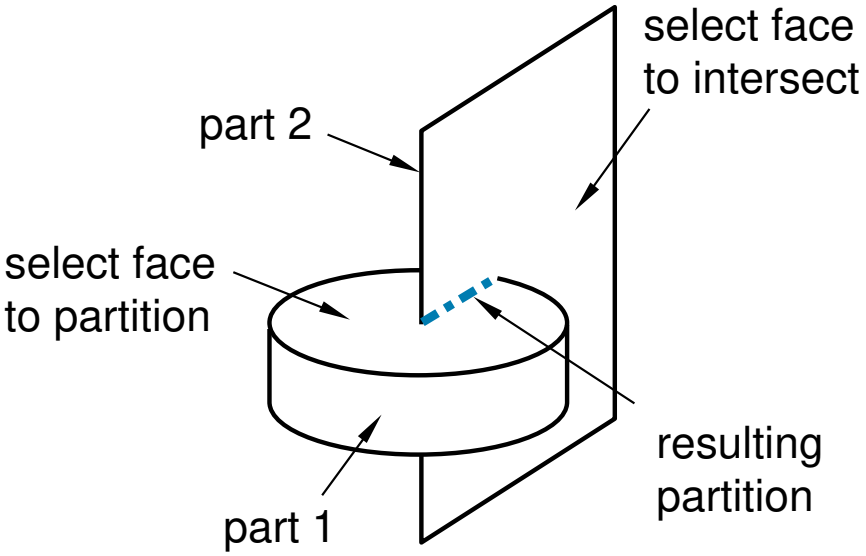


Figure 6: Partitioning a face using an intersection of faces.

Project edges

Partition the face by projecting edges in the model, as shown in *Figure 7*. The partition is created using a perpendicular projection from the face being partitioned to the partitioning edge. You can choose to use only the projection or, if necessary, to extend the ends of the projected edge to complete the face partition. For detailed instructions, see *Using the project edges method to partition faces*.

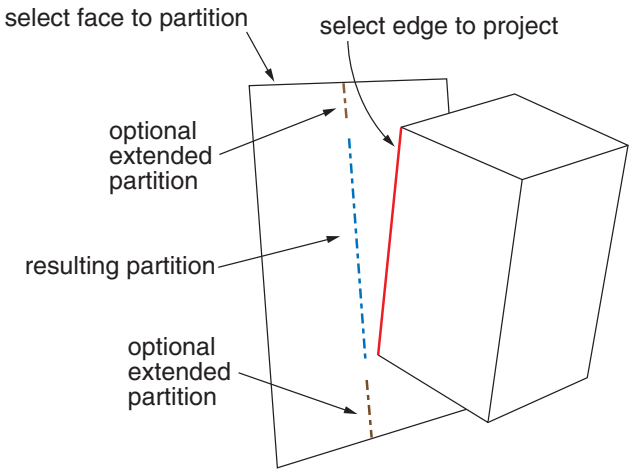


Figure 7: Partitioning a face by projecting an edge.

Auto-partition

When you mesh a face with quadrilateral elements using the free meshing technique, the Mesh module internally partitions the face into regions with three to five logical sides before meshing the face. For more information, see *Free meshing with quadrilateral and quadrilateral-dominated elements*. However, if you want to view and perhaps modify the automatically generated regions before generating the mesh, you can use the auto-partitioning tool to partition the face without meshing it. This tool is available only in the Mesh module. For detailed instructions, see *Using the automatic generation method to partition faces*.

Additional information

- *Partitioning techniques*
- *Understanding partitions*
- *Partitioning faces*

Methods for partitioning cells

When you choose **Cell** from the **Create Partition** dialog box, the **Method** list displays the following methods for partitioning cells:

Define cutting plane

Partition a cell by cutting it with a plane; the plane will pass completely through the cell. Use one of the following three methods to define the cutting plane:

- Select a point on the cutting plane; then pick an edge or datum axis that defines the normal to this plane, as shown in *Figure 1*.

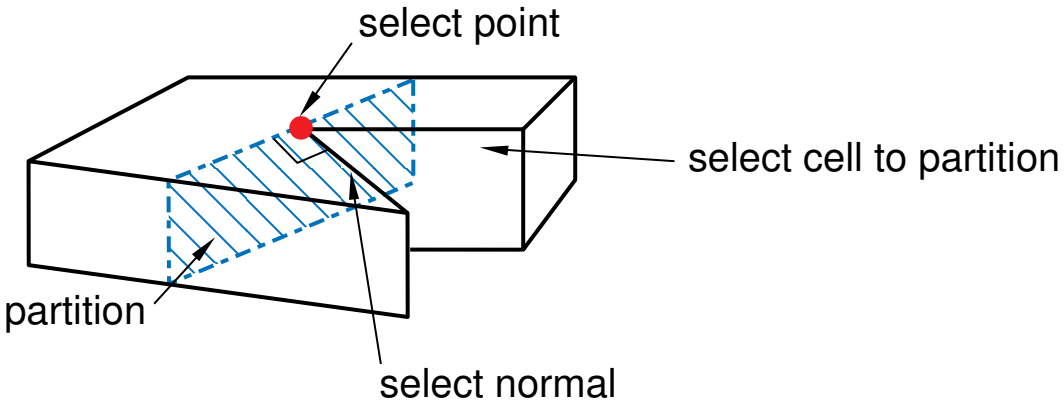


Figure 1: Defining the cutting plane with a point and a normal.

- Select three distinct and noncolinear points, as shown in *Figure 2*.

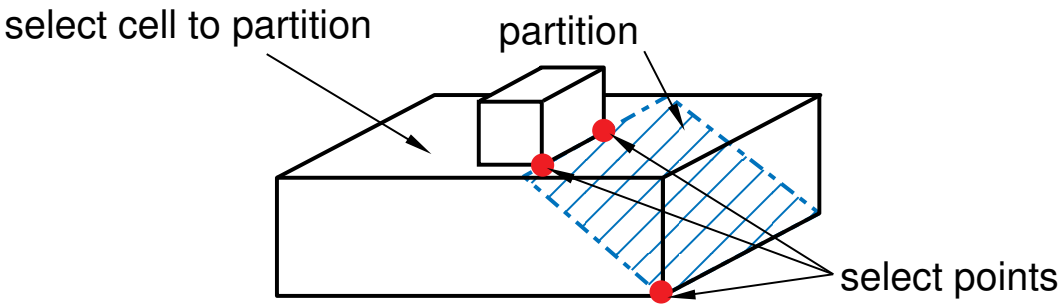


Figure 2: Defining the cutting plane with three points.

- Select an edge and a point along the edge; the cutting plane will be normal to the edge at the selected point, as shown in *Figure 3*.

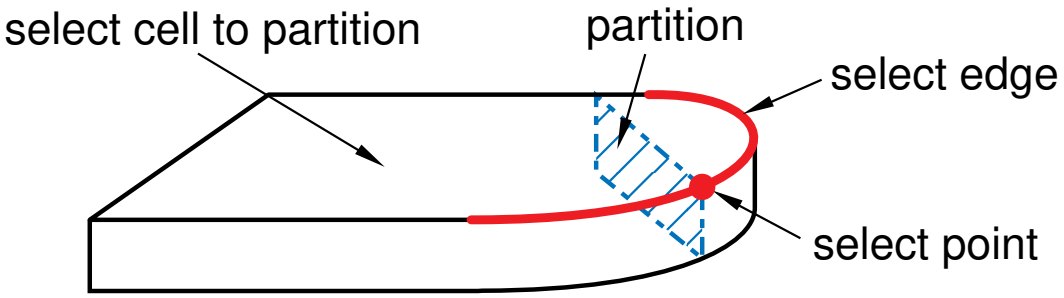


Figure 3: Defining the cutting plane with an edge and a point.

For detailed instructions, see *Using the cutting plane method to partition cells*.

Use datum plane

Partition a cell using the intersection with the extension of a datum plane, as shown in *Figure 4*. For detailed instructions, see *Using the datum plane method to partition cells*.

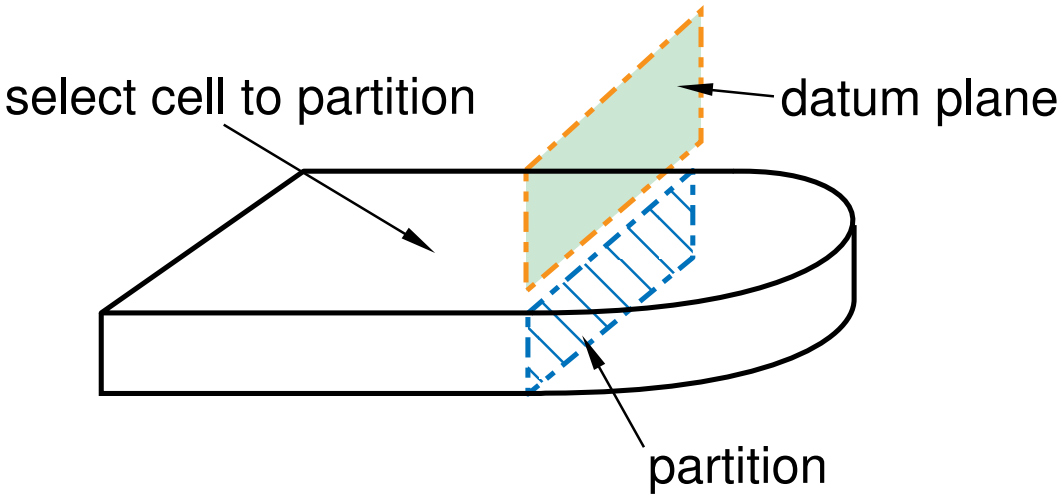


Figure 4: Partitioning a cell using a datum plane.

Extend face

Partition a cell by cutting it with a shell, where the shell is the extended geometry of a face, as shown in *Figure 5*. The face being extended can be planar, cylindrical, conical, or spherical. For detailed instructions, see *Using the extended face method to partition cells*.

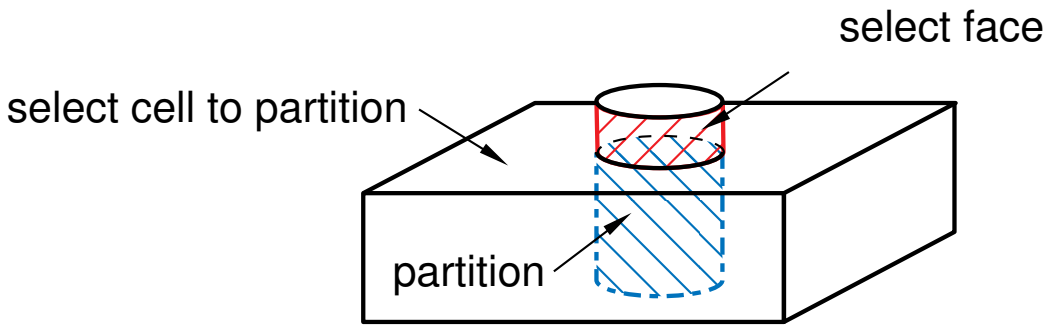


Figure 5: Partitioning a cell using an extension of a face.



Extrude/Sweep edges

Partition a cell by sweeping selected edges (that form the sweep profile) along a selected path (known as the sweep path). You can select any number of edges to be swept, although all the edges must lie on the same plane and must belong to the same part instance.

Use either of the following two methods to define the sweep path:

- Create a straight partition through the cell by extending the sweep profile infinitely in a direction parallel to a selected straight edge or datum axis that acts as a sweep path; the partition is created where the swept edge(s) pass through the selected cell, as shown in *Figure 6*. The sweep path must be straight and perpendicular to the set of edges being swept.

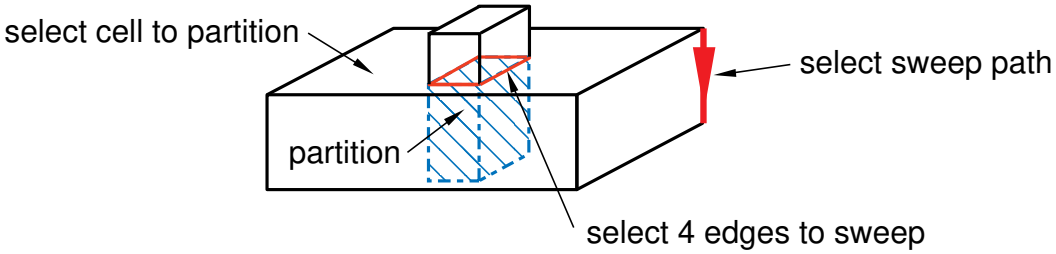


Figure 6: Sweeping a profile along a direction.

- Create a straight or curved partition through the cell by extending the sweep profile along or parallel to a selected edge. The partition extends only as far as the selected edge; and the partition is created where the swept edge(s) pass through the selected cell, as shown in *Figure 7*. The sweep path must begin in the plane containing the edges to be swept, and its tangent must be perpendicular to the same plane.

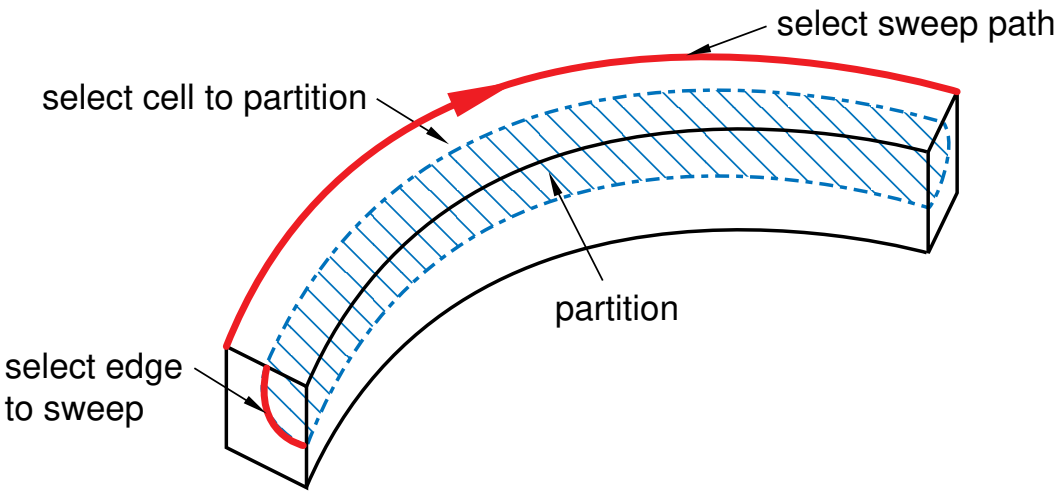


Figure 7: Sweeping a profile along an edge.

For detailed instructions, see *Using the extrude/sweep method to partition cells*.

Use n-sided patch

Partition a cell by dividing it with a surface patch formed from a loop of connected edges. The edges can be curved or straight, must be connected, and must belong to the same part as the cell to be partitioned. In addition, the patch must pass completely through the cell. Choose from the following methods to define the patch:

Select Edges

You can choose from the following methods to select the edges that form the N-sided patch:

Loop

Select a single edge, and allow Abaqus/CAE to search for a continuous loop of connected edges that will partition the cell, as shown in *Figure 8*. The resulting patch can have any number of edges.

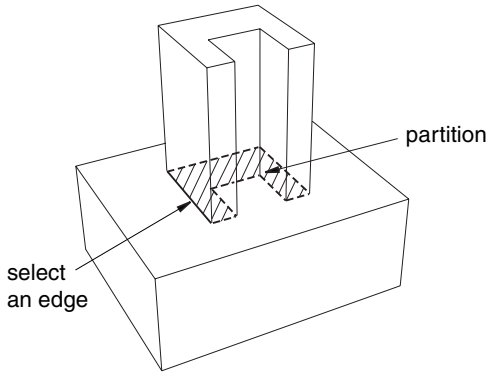


Figure 8: Allowing Abaqus/CAE to define a patch after selecting an edge.

Edges

Manually select the edges that will partition the cell. You can select any number of edges, and the selected edges must form a closed loop.

Select Corner Points

Select three, four, or five points that define the corners of the patch. If two of the points are connected by an existing edge, the resulting partition will follow the curve of the edge, as shown in *Figure 9*. The points must be on the boundary edges of the cell being partitioned.

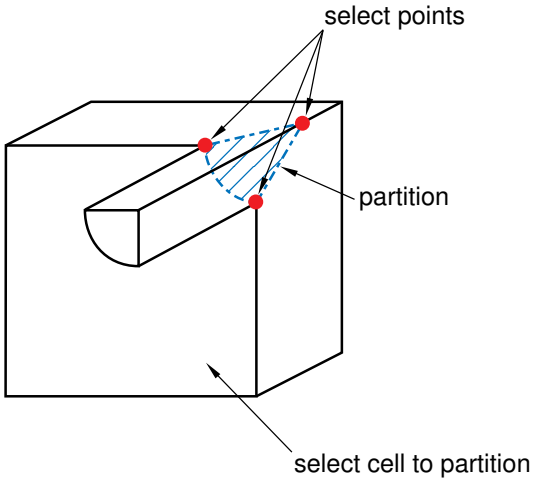


Figure 9: Defining a patch with corner points.

For detailed instructions, see *Using the N-sided patch method to partition a cell*.

Sketch planar partition

Partition a selected cell by sketching a partition with the Sketcher, as shown in *Figure 10*. In most cases you will sketch on a datum plane that intersects the selected cell. You can also select an existing face on which to sketch and draw the sketch outside the boundaries of the face. Abaqus/CAE creates the partition wherever the sketch intersects the cell.

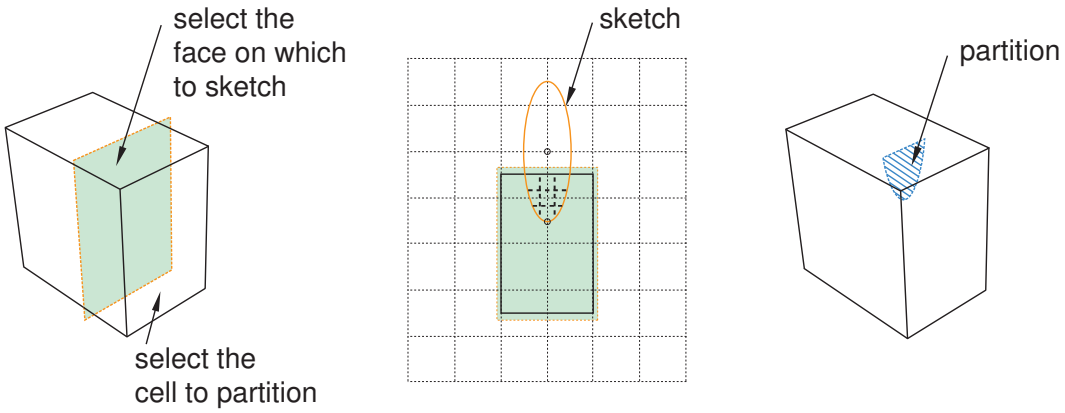


Figure 10: Partitioning a cell using the Sketcher.

For detailed instructions, see *Using the sketch planar partition method to partition a cell*.

Additional information

- *Partitioning techniques*
- *Understanding partitions*
- *Partitioning cells*

Partitioning edges

This section describes the tools that you can use to partition a selected edge.

In this section:

- *Using the specify parameter by location method to partition edges*
- *Using the enter parameter method to partition edges*
- *Using the pick midpoint/datum point method to partition an edge*
- *Using the datum plane method to partition edges*

Using the specify parameter by location method to partition edges

You can partition a selected edge anywhere along its length by clicking on the edge at the location of the partition. Abaqus/CAE converts the location to a parameter. As a result, you can use the Feature Manipulation toolset to modify the location of the partition.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the specify parameter method to partition edges using the  tool, located with the partition edge tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Edge**.

   The **Method** list displays the methods that you can use to partition an edge.

3. From the list of methods, select **Specify parameter by location**.
4. Click anywhere along the edge.

   Abaqus/CAE highlights the edge and the location along the edge that you picked.

5. If the location of the partition is not correct, click along the edge again.
6. If the location of the partition is correct, click **Create Partition** in the prompt area.

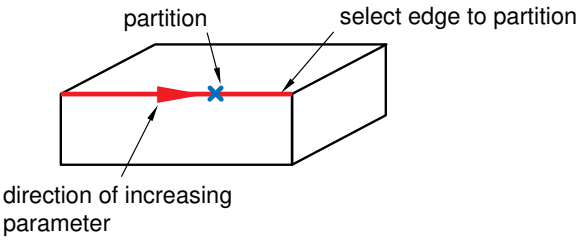
   Abaqus/CAE creates the partition.

Additional information

- *Partitioning techniques*
- *Methods for partitioning edges*

Using the enter parameter method to partition edges

An arrow along the selected edges points from the start vertex, corresponding to an edge parameter value of zero, to the end vertex, corresponding to a value of one. Partitions are created along the selected edges at the location determined by the parameter, as shown in the following figure:



To modify a partition that was created by entering a parameter, use the Feature Manipulation toolset to change the parameter.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the enter parameter method to partition edges using the tool, located with the partition edge tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Edge**.

   The **Method** list displays the methods that you can use to partition an edge.

3. From the list of methods, select **Enter Parameter**.
4. If the part or assembly contains more than one edge, select the edges to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one edge to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected edges, and arrows indicate the direction of increasing parameter value.

5. In the prompt area, click **Done** to indicate you have finished selecting edges.
6. In the prompt area, enter the desired edge parameter as a value between zero and one.
7. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

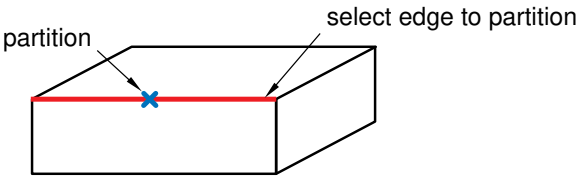
Additional information

- *Partitioning techniques*
- *Methods for partitioning edges*

Using the pick midpoint/datum point method to partition an edge

You can partition a selected edge by selecting the midpoint of the edge or a datum point along the edge.

The figure below shows an example of partitioning a selected edge by selecting the midpoint of the edge or a datum point along the edge.



You cannot directly modify a partition that was created by selecting an existing point. However, if you subsequently move the point, the partition will move along with the point.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the pick midpoint/datum point method to partition an edge using the tool, located with the partition edge tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Edge**.

   The **Method** list displays the methods that you can use to partition an edge.

3. From the list of methods, select **Select midpoint/datum point**.
4. Select the edge to partition.

   Abaqus/CAE highlights the selected edge.

5. Select the midpoint of the edge or a datum point along the edge.

   Abaqus/CAE highlights the location of the partition.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

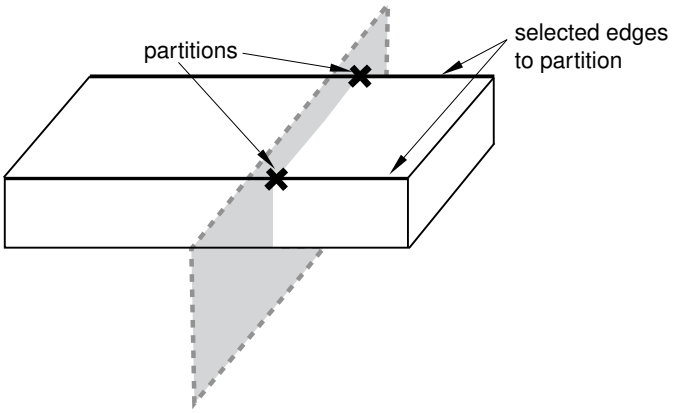
Additional information

- *Selecting objects within the viewport*
- *Partitioning techniques*
- *Methods for partitioning edges*

Using the datum plane method to partition edges

You can partition selected edges by cutting them with a datum plane

The figure below shows an example of partitioning selected edges by cutting them with a datum plane.



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the datum plane method to partition edges using the tool, located with the partition edge tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Edge**.

   The **Method** list displays the methods that you can use to partition an edge.

3. From the list of methods, select **Use datum plane**.
4. If the part or assembly contains more than one edge, select the edges to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one edge to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected edges.

5. Select an existing datum plane. If a datum plane does not exist in the desired location, create one using the Datum toolset.

   Abaqus/CAE highlights the selected datum plane.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

Additional information

- *Partitioning techniques*
- *Methods for partitioning edges*

Partitioning faces

This section describes the tools that you use to partition a selected face.

In this section:

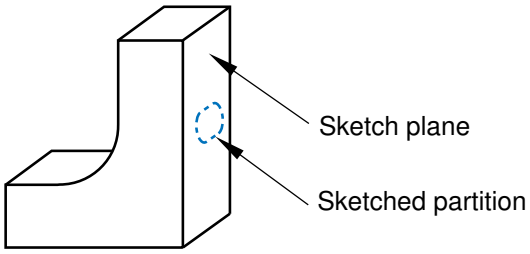
- *Using the sketch method to partition faces*
- *Using the shortest path method to partition faces*
- *Using the datum plane method to partition faces*
- *Using the curved path method to partition a face*
- *Using the extended face method to partition faces*
- *Using the intersection method to partition faces*
- *Using the project edges method to partition faces*
- *Using the automatic generation method to partition faces*

Using the sketch method to partition faces

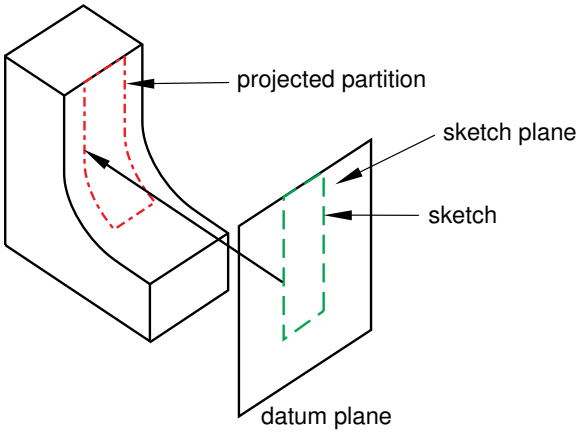
You can partition selected faces by sketching the geometry of the partition using the Sketcher. You use the Sketcher when you want to create a partition whose shape cannot be achieved with any of the other partition tools. Sketching a partition offers other advantages:

- You can use the Feature Manipulation toolset to modify the sketch if you want to change the shape or the position of the partition.
- You can dimension the sketch precisely, and you can use the Feature Manipulation toolset to modify the dimensions.

If the faces to be partitioned are planar and lie in the same plane, you can sketch directly on that plane. Your sketch can extend beyond the boundaries of the selected faces, but the partition will not extend beyond the edge of the faces. The following figure illustrates a sketched partition on a planar face:



If any of the faces to be partitioned are nonplanar or do not lie in the same plane, you must select a planar face or datum plane on which to sketch. Abaqus/CAE creates the partition by projecting the sketch onto the faces to be partitioned. You can specify the direction of the projection and the distance of the projection, normal to the sketch plane. The following figure illustrates a sketched partition on a nonplanar face using the projection of a sketch drawn on a datum plane:



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the sketch method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Sketch**.

   If the part or assembly contains only one face, Abaqus/CAE enters the Sketcher.

4. If the part or assembly contains more than one face, select the faces to partition and click mouse button 2. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

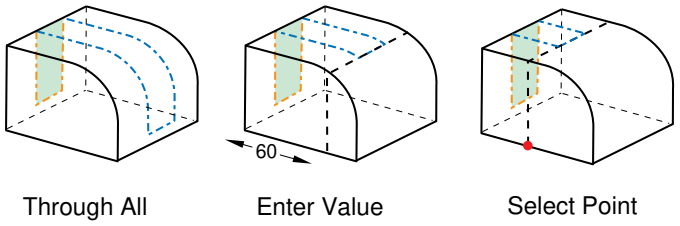
   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   - If the part or assembly is two-dimensional or axisymmetric, Abaqus/CAE enters the Sketcher and highlights the selected faces.
   - If the part or assembly is three-dimensional and the selected faces are planar, select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the sketch plane. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

     Tip: If the part or assembly does not have an edge that will provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

     Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected planar face or datum plane aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation. Abaqus/CAE highlights the selected faces.
   - If the part or assembly is three-dimensional and the selected faces are nonplanar or do not lie in the same plane, do the following:

     1. Select a planar face or a datum plane on which to sketch. Abaqus/CAE highlights the selected face.
     2. From the buttons in the prompt area, select a method to determine how far Abaqus/CAE projects the sketch normal to the sketch plane. The partition is created wherever the projected sketch extends through the selected faces. The three methods are illustrated in the following figure:



        - If you selected the **Through All** method, Abaqus/CAE asks you to choose which direction the sketch will be projected.
        - If you selected the **Enter Value** method, Abaqus/CAE asks you to choose the distance and direction that will be used to project the sketch. The default distance that Abaqus/CAE provides generates a partition that extends completely through the selected faces (resulting

in the same partition as selecting **Through All**). The Feature Manipulation toolset allows you to modify the projection distance after you create the partition.

- If you selected the **Select Point** method, Abaqus/CAE asks you to choose the point to which the sketch will be projected; the point defines both the direction and the projection distance.

3. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the sketch plane. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.

   Tip: If the part or assembly does not have an edge that will provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

   Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected planar face or datum plane aligns with the plane of the Sketcher grid with the projection direction into the screen and the selected edge aligns with the grid in the desired orientation. Abaqus/CAE highlights the selected faces.

5. Use the Sketcher to sketch the partition. From the main menu bar, click **Done** to indicate you have finished sketching the partition.
6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

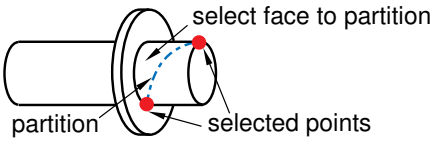
Additional information

- *The Sketch module*
- *Partitioning techniques*
- *Methods for partitioning faces*

Using the shortest path method to partition faces

You can partition selected faces with a line or curve that forms the shortest possible path between two selected points.

The points can be vertices, midpoints, arc-centers, or datum points. The path will be curved if a face being partitioned is not planar, as shown in the following figure:



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the shortest path method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Shortest path between 2 points**.
4. If the part or assembly contains more than one face, select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected faces.

5. Select the vertices, midpoints, arc-centers, or datum points defining the start and end of the path.

   Abaqus/CAE highlights the points.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

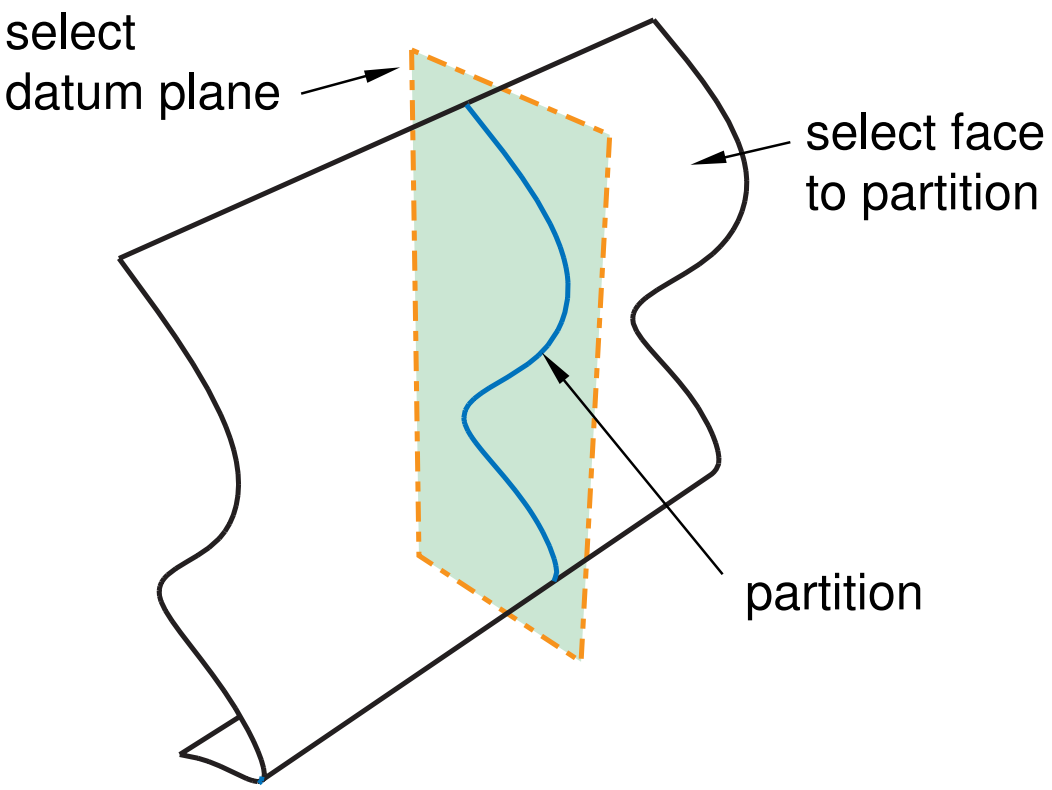
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Using the datum plane method to partition faces

You can partition selected faces by cutting them with a datum plane.

The figure below shows an example of partitioning selected faces by cutting them with a datum plane.



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the datum plane method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Use datum plane**.

4. If the part or assembly contains more than one face, select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected faces.

5. Select an existing datum plane. If a datum plane does not exist in the desired location, create the datum using the Datum toolset.

   Abaqus/CAE highlights the selected datum plane.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

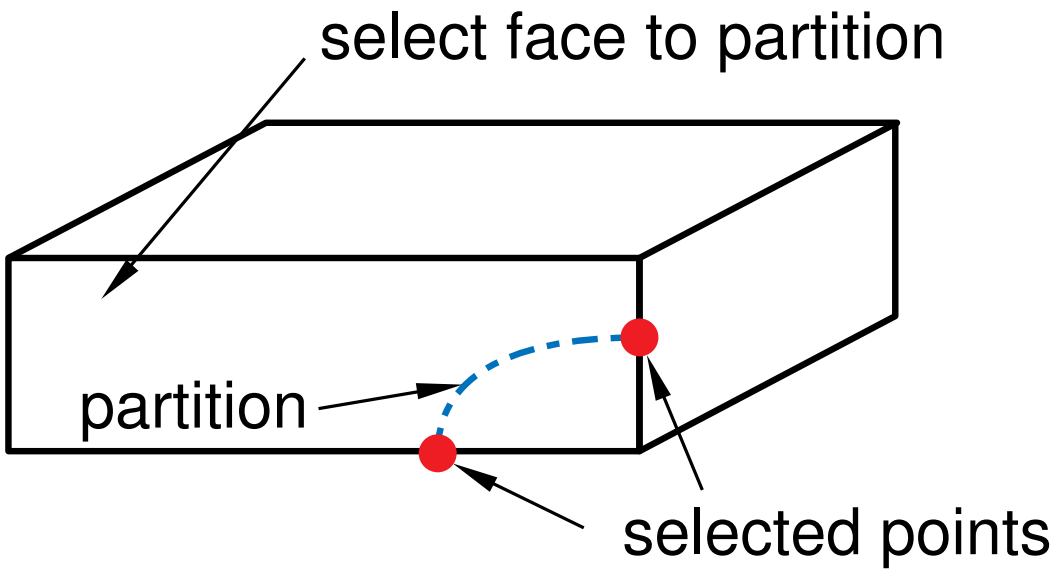
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Using the curved path method to partition a face

You can partition a face with a curve normal to two edges that bound the face.

The edges need not be contiguous, but they must bound the face and subtend an angle less than 180 degrees, as shown in the following figure:



You first select the face to partition and then select one of two methods to define the position of the partition along the two edges; Abaqus/CAE draws a Bézier curve connecting the two points. You can locate the partition along the selected edges by either:

- Entering a parameter value between zero and one, where zero represents the start vertex of the edge and one the end vertex. This method allows you to precisely position the partition anywhere along the edges. In addition, you can subsequently modify the partition using the Feature Manipulation toolset to edit the two parameters that you entered to locate the partition.
- Clicking the midpoint or a datum point along the edge. If you use this method to position a curved partition, you will not be able to modify the partition later.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the curved path method to partition a face using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Curved path normal to 2 edges**.
4. Select the face to partition.

   Abaqus/CAE highlights the selected face.

5. From the buttons in the prompt area, select one of the following methods to locate the curve:
   - **Enter Parameter**
   - **Pick**
6. Select any of the edges bounding the selected face.

   Tip: If you are unable to select the desired edges, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected edge.
7. Do one of the following:
   - If you selected the **Enter Parameter** method, enter a value between zero and one. An arrow on the selected edge indicates the direction of increasing parameter value.
   - If you selected the **Pick** method, click the midpoint or a datum point along the selected edge.
8. Select a second edge that bounds the face and repeat the previous step. The angle subtended by the two selected edges must be less than 180 degrees.
9. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

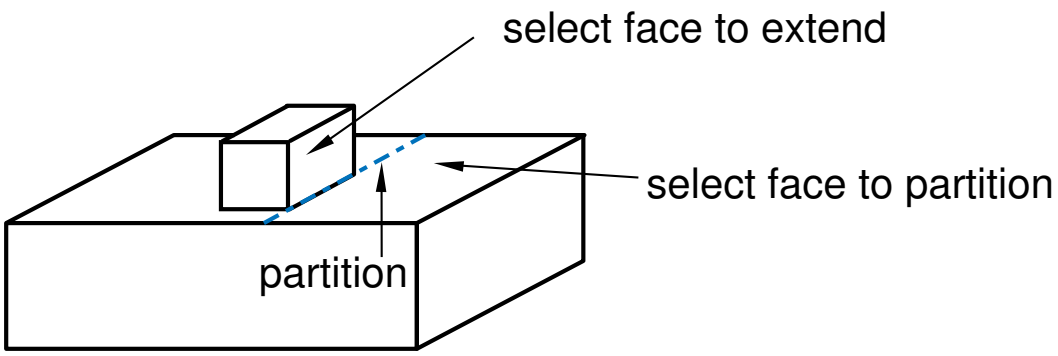
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Using the extended face method to partition faces

You can partition selected faces by extending another face so that it cuts through the selected faces.

Extending a face is analogous to creating an infinite plane that is coincident with that face. Abaqus/CAE creates a partition wherever this infinite plane slices the selected faces you are trying to partition, as shown in the following figure:



If the infinite plane cannot intersect any of the selected faces, Abaqus/CAE displays an error message. The extended face need not belong to the same part as the faces to be partitioned; for example, in the Assembly module you can partition the face of one part instance by extending the face of a second. Abaqus/CAE displays an error message if the part or assembly contains only one face.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the extended face method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Extend another face**.
4. Select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected faces.

5. Select the face whose extension creates the desired partition.

   Abaqus/CAE highlights the selected face.

6. In the prompt area, click **Create Partition**.

Abaqus/CAE creates the partition.

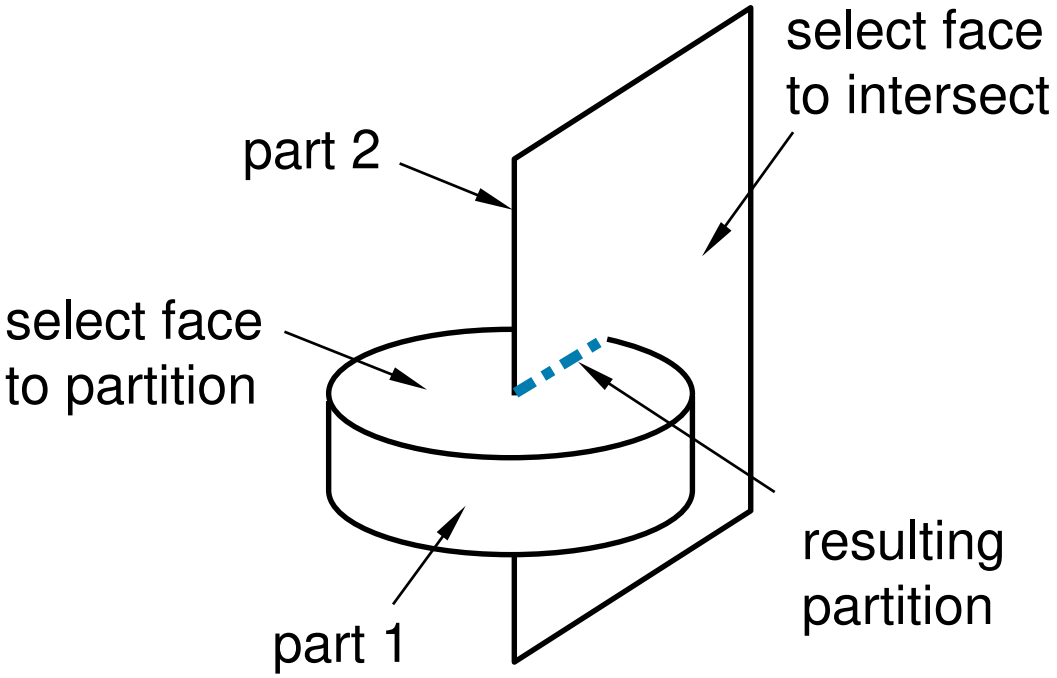
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Using the intersection method to partition faces

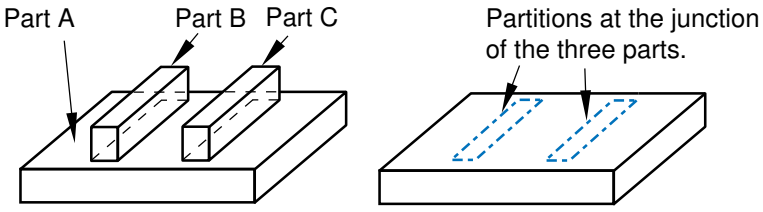
You can partition selected faces using their intersection or junction with one or more other faces.

The figure below shows an example of partitioning selected faces using their intersection or junction with one or more other faces.



The faces need not belong to the same part; for example, in the Assembly module you can partition the face belonging to one part instance using its intersection with a face belonging to a second part instance. The resulting regions are especially useful for defining contact or constraint interactions between part instances. Abaqus/CAE displays an error message if the part or assembly contains only one face.

The following figure shows two partitions that you might create at the intersection of three part instances:



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.



Tip: You can also use the intersection method to partition faces using the  tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Intersect by other faces**.
4. Select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected faces.

5. Select the faces that join or intersect the faces to be partitioned.

   Abaqus/CAE highlights the selected faces.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

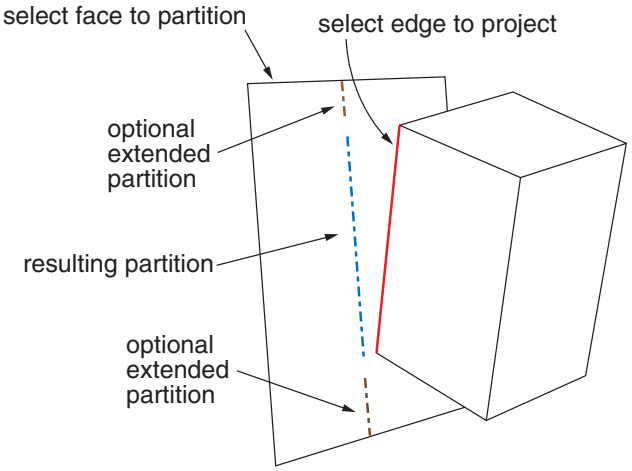
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Using the project edges method to partition faces

You can partition selected faces using their intersection or junction with the projection of one or more edges.

The figure below shows an example of partitioning selected faces using their intersection or junction with the projection of one or more edges.



Abaqus/CAE creates the partition by taking a perpendicular projection from the faces being partitioned to the selected edges; nonplanar faces or faces joined at a sharp edge such that there is a discontinuity in the perpendicular projection may significantly change the shape of the projected edges. As shown in the figure, you can create a partition that ends at the endpoints of the projected edges, or you can extend the projected edges to reach the face boundaries. If you extend the partition, Abaqus/CAE extends the free ends of the projected edges along their tangents to reach the boundary edges of the faces being partitioned.

The faces and edges need not belong to the same part; for example, in the Assembly module you can partition faces belonging to one part instance using edges projected from a second part instance. The faces to be partitioned must be part of the active model representation. The edges can be part of the active representation or part of the reference representation for a midsurface model.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the project edges method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Project edges**.
4. Select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

Abaqus/CAE highlights the selected faces.

5. Select the edges to be projected onto the faces. If desired, toggle on **Extend edges** to have Abaqus/CAE extend the projected edges, preventing gaps in the new face partitions.

   Note:

   You can also toggle the **Extend edges** parameter on or off by editing the completed partition feature in the Model Tree.

   Abaqus/CAE highlights the selected edges.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE projects the selected edges to create the partition.

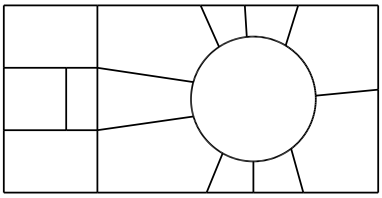
Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

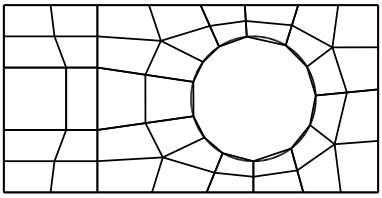
Using the automatic generation method to partition faces

The Mesh module is capable of meshing any two-dimensional face automatically.

Automatic meshing with quadrilateral elements begins by internally partitioning the face into regions having between three and five sides, as shown in the following figure:



Then, a mesh is generated on each of these regions, as shown in the following figure:



Normally, when you mesh a face, these two steps take place in what appears to be a single step. However, to gain more control over the meshing process, you can perform the partitioning step separately using the automatic partitioning tool in the Partition toolset.

Since the automatic partitioning tool is associated with meshing, it is available only within the Mesh module. After you automatically partition the face, you can gain additional control over the mesh by seeding or by adding or deleting partitions manually.

1. From the main menu bar, select **Tools**->**Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the automatic generation method to partition faces using the tool, located with the partition face tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Face**.

   The **Method** list displays the methods that you can use to partition a face.

3. From the list of methods, select **Auto-partition**.
4. If the part or assembly contains more than one face, select the faces to partition. You can use a combination of drag select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to partition. For more information, see *Selecting objects within the current viewport*.



Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

Abaqus/CAE highlights the selected faces.

5. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

Additional information

- *Partitioning techniques*
- *Methods for partitioning faces*

Partitioning cells

This section describes the partition tools that you use to partition a cell.

In this section:

- *Using the cutting plane method to partition cells*
- *Using the datum plane method to partition cells*
- *Using the extended face method to partition cells*
- *Using the extrude/sweep method to partition cells*
- *Using the N-sided patch method to partition a cell*
- *Using the sketch planar partition method to partition a cell*

Using the cutting plane method to partition cells

You can partition selected cells by cutting them with a plane. You first select the cells to partition and then select one of three methods to define the plane that partitions the cells. The three methods are:

- Select a point and any straight edge. The cutting plane will pass through the point, normal to the edge.
- Select three points. The cutting plane will pass through all three points.
- Select a straight or curved edge and a point on the edge. The cutting plane will pass through the point, normal to the edge at the selected point.

The following procedure illustrates the three methods.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the plane method to partition cells using the tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

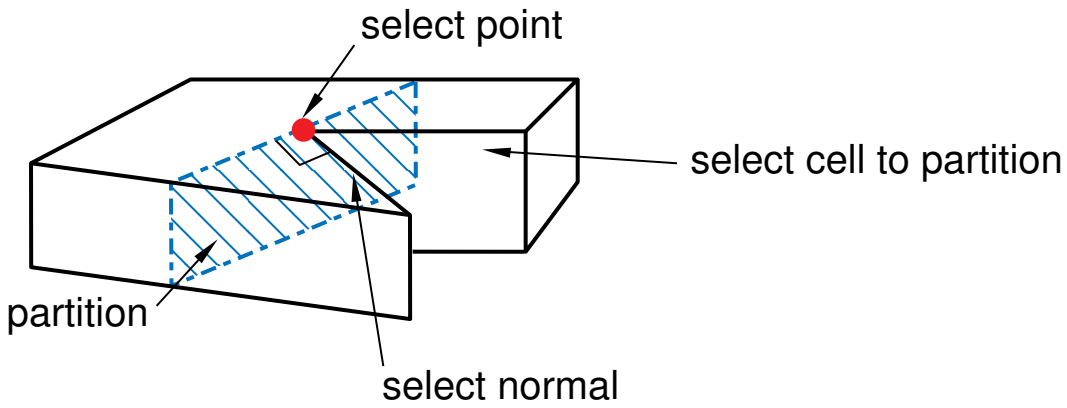
3. From the list of methods, select **Define cutting plane**.
4. If the part or assembly contains more than one cell, select the cells to partition. You can use a combination of drag select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one cell to partition.

   Tip: If you are unable to select the desired cells, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

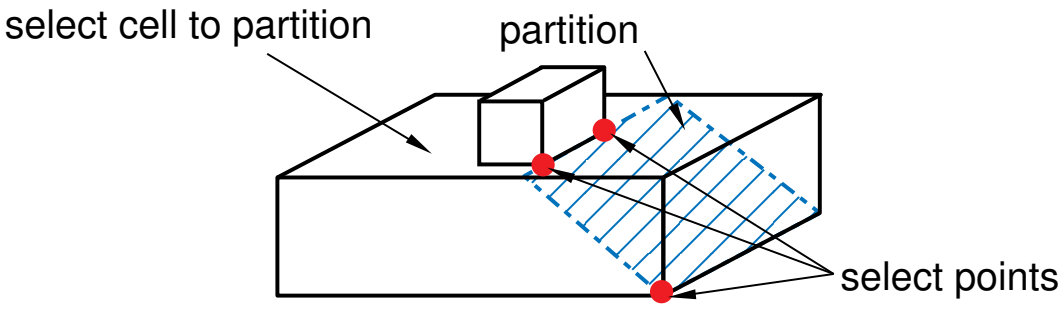
   Abaqus/CAE highlights the selected cell.

5. From the prompt area, click one of the following methods to define the cutting plane:

   - Use the **Point & normal** method to partition the selected cells along a plane that passes through a selected point and is normal to a selected straight edge or datum axis, as shown in the following figure:

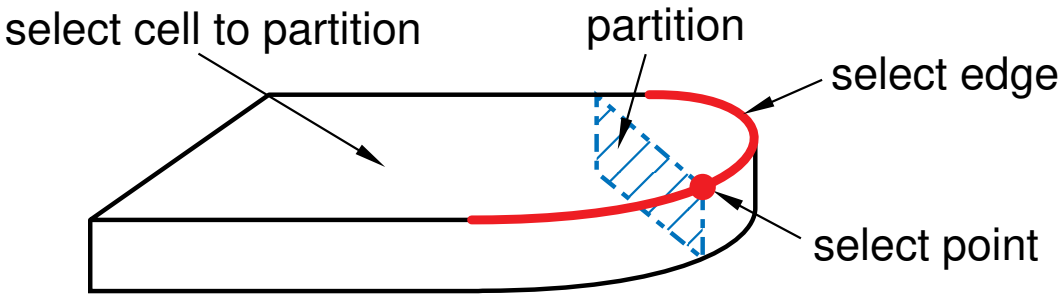


- Use the **3 Points** method to partition the selectes cells along a plane passing through three selected points, as shown in the following figure:



The three points can be anywhere on the part or assembly, but they must be distinct and they must not be colinear.

- Use the **Normal to edge** method to partition the selected cells along a plane that is normal to a selected edge and passes through a selected point on the edge. The edge can be straight or curved, and need not be part of the cells being partitioned, although you cannot select a datum axis. This method is useful when you want to partition a cell normal to a curved edge, as shown in the following figure:



6. Select points and edges as directed by the prompts in the prompt area. Points can be vertices, midpoints, arc centers, or datum points.
7. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

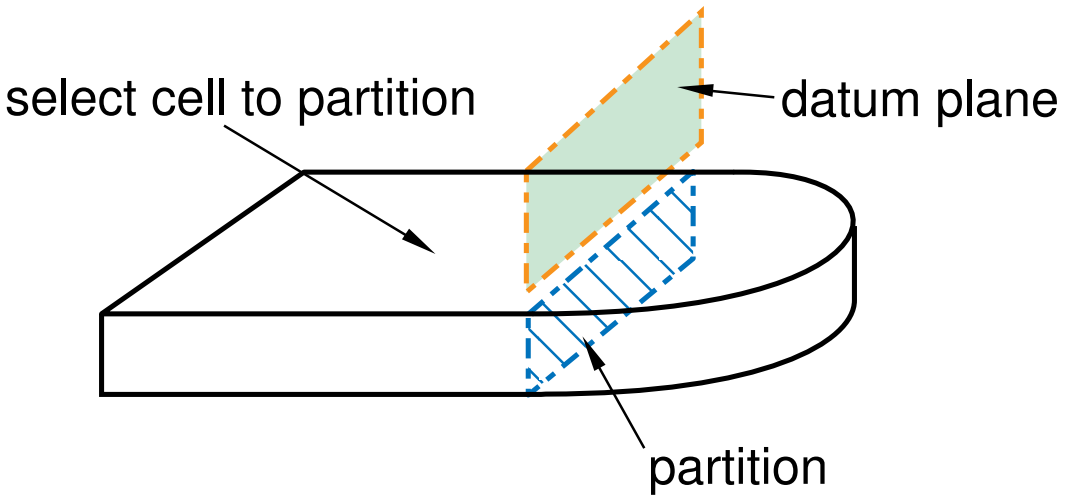
Additional information

- *Partitioning techniques*
- *Methods for partitioning cells*

Using the datum plane method to partition cells

You can partition selected cells by cutting them with a datum plane.

The figure below shows an example of partitioning selected cells by cutting them with a datum plane.



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the datum plane method to partition cells using the tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

3. From the list of methods, select **Use datum plane**.
4. If the part or assembly contains more than one cell, select the cells to partition. You can use a combination of drag select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one cell to partition.

   Tip: If you are unable to select the desired cells, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected cell.

5. Select an existing datum plane. If a datum plane does not exist in the desired location, create the datum using the Datum toolset.

   Abaqus/CAE highlights the selected datum plane.

6. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

Additional information

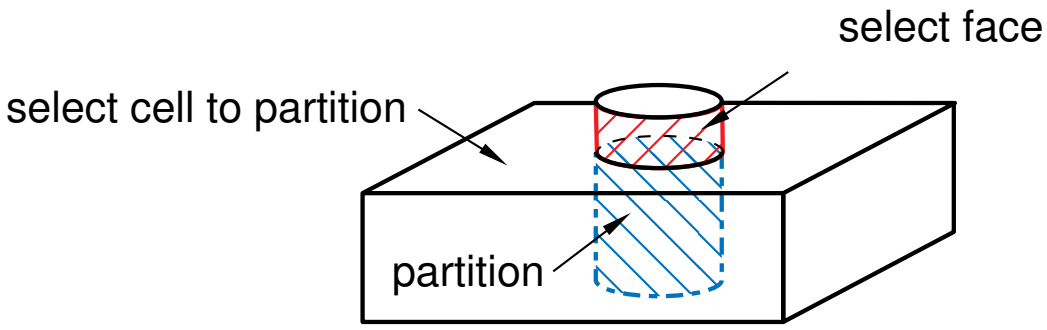
- *Partitioning techniques*

• *Methods for partitioning cells*

Using the extended face method to partition cells

You can partition selected cells by extending another face until it cuts through the selected cells.

Extending a face is analogous to creating an infinite plane that is coincident with the face. Abaqus/CAE creates a partition wherever this infinite plane slices the cells that you are partitioning, as shown in the following figure:



If the infinite plane cannot intersect the cells, Abaqus/CAE displays an error message. The extended face need not belong to the same part as the cells to be partitioned; for example, in the Assembly module you can partition the cell of one part instance by extending the face of a second.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the datum plane method to partition cells using the tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

3. From the list of methods, select **Extend face**.

4. If the part or assembly contains more than one cell, select the cells to partition. You can use a combination of drag select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one cell to partition.

   Tip: If you are unable to select the desired cells, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected cell.

5. Select the face that will be extended to create the desired partition.

   Abaqus/CAE highlights the selected faces.

   Tip: If you are unable to select the desired faces, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

6. In the prompt area, click **Create Partition**.

Abaqus/CAE creates the partition.

Additional information

- *Partitioning techniques*
- *Methods for partitioning cells*

Using the extrude/sweep method to partition cells

You can create complex three-dimensional partitions using the sweep tool to sweep two-dimensional profiles (known as sweep profiles) through three-dimensional parts or part instances. Using the sweep tool is a two-stage operation. First you define the sweep profile by selecting the edges to sweep, and then you select the path along which to sweep the selected edges (known as the sweep path). You can choose between two methods to define the sweep path:

- **Extrude along direction**
- **Sweep along edge**

These two methods are illustrated below.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the extrude/sweep method to partition cells using the  tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

3. From the list of methods, select **Extrude/sweep edges**.
4. If the part or assembly contains more than one cell, select the cells to partition. You can use a combination of drag select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one cell to partition.

   Tip: If you are unable to select the desired cells, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

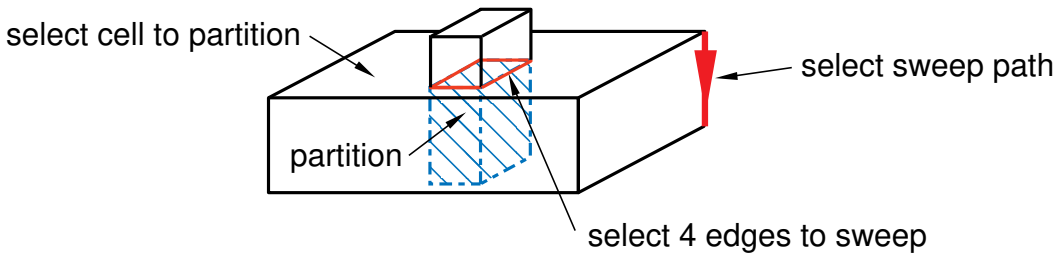
   Abaqus/CAE highlights the selected cell.

5. To create the sweep profile, select the edges to sweep through the cells. The edges must all belong to the same part, lie on the same plane, and be connected to each other.

   Abaqus/CAE highlights the selected edges.

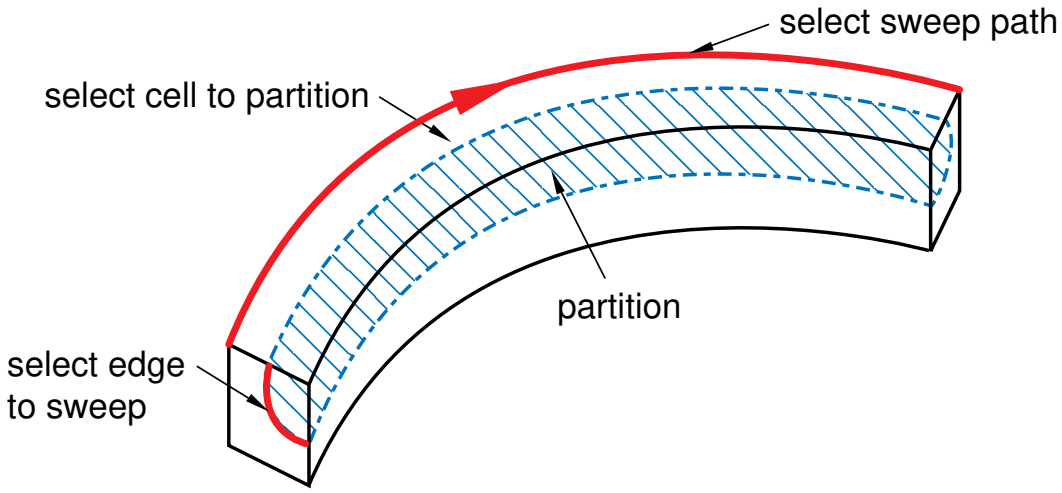
6. From the buttons in the prompt area, click on one of the following methods to define the sweep path:

   - Use the **Extrude along direction** method to partition a cell by extruding selected edges—the sweep profile—along a straight edge or datum axis—the sweep path—in a selected direction. The sweep path must be perpendicular to the plane containing the sweep profile. The partition extends infinitely in the selected direction through the selected cell. Extruding a sweep profile along a direction to partition a cell is illustrated in the following figure:



Abaqus/CAE highlights the sweep profile.

- Use the **Sweep along edge** method to partition a cell by sweeping selected edges—the sweep profile—along a straight or curved edge—the sweep path. The sweep path must begin in the plane containing the sweep profile, and its tangent must be perpendicular to that same plane. The resulting partition will extend only as far as the sweep path. Sweeping a sweep profile along an edge to partition a cell is illustrated in the following figure:



Abaqus/CAE highlights the sweep path.

7. In the prompt area, click **Create Partition**.
   Abaqus/CAE creates the partition.

Additional information

- *Partitioning techniques*
- *Methods for partitioning cells*

Using the N-sided patch method to partition a cell

You can partition a cell by creating a patch that divides the cell into regions. You can define the patch using either edges or corner points. If you select an edge, Abaqus/CAE searches for a continuous loop of connected edges that will define a viable patch. If you select corner points, you can select either three, four, or five points to define the partition. In contrast with the other cell partition tools, the partition that results from an N-sided patch does not extend beyond the patch. As a result, the N-sided patch method is useful for creating an isolated partition that does not extend into the rest of the cell.

If you select a curved edge or if you select two points connected by a curved edge, the corresponding edge of the resulting N-sided patch follows the contour of the edge. However, if you select points that are not connected by an edge, the patch simply connects the points; the patch will not follow the contour of the cell, and the partition may be incomplete.

1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the N-sided patch method to partition a cell using the  tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

3. From the list of methods, select **N-sided patch**.
4. If the part or assembly contains more than one cell, select the cell to partition.

   Tip: If you are unable to select the desired cell, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE highlights the selected cell.

   Tip: Use the **Previous** button () to undo the steps in a procedure.

5. From the buttons in the prompt area, select the method to define the N-sided patch:

   - Use the **Select Edges** method to partition a cell by defining an N-sided patch from a sequence of selected edges. Use the text field in the prompt area to select the method of choosing edges:

     ### Loop

     Select a single edge, and allow Abaqus/CAE to search for a continuous loop of connected edges that will partition the cell, as shown in the following figure:

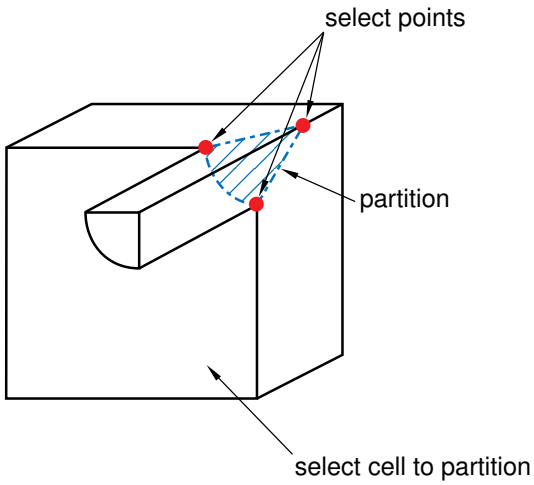
     

     The edge that you select must be on the boundary of the cell being partitioned, and Abaqus/CAE must be able to form a closed loop from a series of connected edges. The resulting patch can have any number of edges. Abaqus/CAE highlights the loop that will form the patch.

Edges

Manually select the edges that will partition the cell. The edges that you select must be on the boundary of the cell being partitioned. You can select any number of edges, and the selected edges must form a closed loop.

- Use the **Select Corner Points** method to partition a cell by defining an N-sided patch using three to five selected points. Abaqus/CAE creates the patch by connecting the points in the order that you select them; consequently, the order in which you select the points is significant. The points can be vertices, midpoints, or datum points and must be on the boundary edges of the cell being partitioned. If a curved edge connects two of the points, the patch follows the contour of the curve, as shown in the following figure:



Abaqus/CAE highlights the selected points along with the edges that will form the patch.

6. Do one of the following:

   - If the highlighted edges indicate that Abaqus/CAE will create the desired partition, click **Create Partition** in the prompt area.

     Abaqus/CAE creates the partition.

   - To change the selection of edges, click the **Previous** button () to undo the previous step.

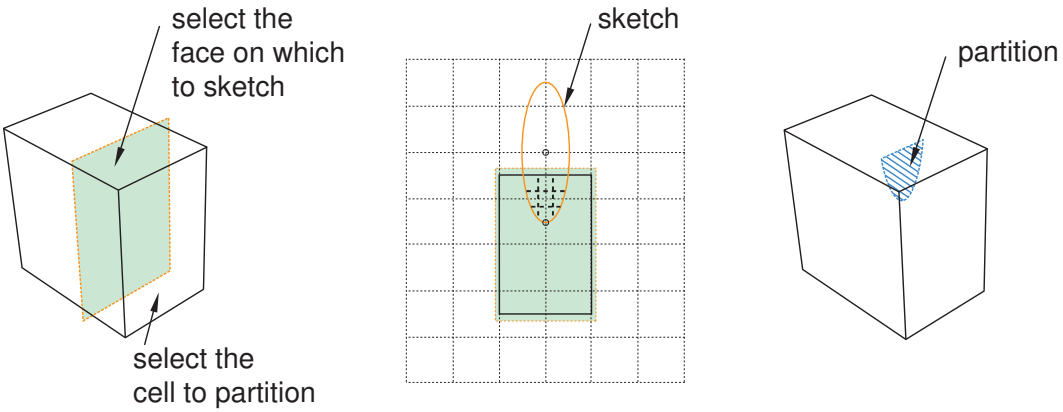
Additional information

- *Partitioning techniques*
- *Methods for partitioning cells*

Using the sketch planar partition method to partition a cell

You can partition a selected cell by sketching the geometry of a planar partition using the Sketcher. You use the Sketcher when you want to create a partition whose shape cannot be achieved with any of the other partition tools. A sketched partition is useful for creating a crack face that will be used in a fracture mechanics analysis.

You must sketch on a planar face. In most cases you will sketch the partition on a datum plane that intersects the selected cell. The following figure illustrates a sketched planar partition on a datum plane:



1. From the main menu bar, select **Tools->Partition**.

   The **Create Partition** dialog box appears. Abaqus/CAE displays prompts in the prompt area to guide you through the procedure.

   Tip: You can also use the sketch planar partition method to partition cells using the tool, located with the partition cell tools in the module toolbox. For a diagram of the partition tools in the toolbox, see *Using the Partition toolset*.

2. From the **Type** radio buttons at the top of the dialog box, choose **Cell**.

   The **Method** list displays the methods that you can use to partition a cell.

3. From the list of methods, select **Sketch planar partition**.

   If the part or assembly contains only one cell, Abaqus/CAE enters the Sketcher.

4. If the part or assembly contains more than one cell, select the cell to partition and click mouse button 2.

   Tip: If you are unable to select the desired cell, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

5. Select the planar face on which to sketch. If no suitable face exists, you can select a datum plane.

   The selected face is highlighted in the viewport.

6. Select an edge and the orientation of the edge on the Sketcher grid. The edge must not be perpendicular to the sketch plane. By default, the selected edge will appear vertical and on the right side of the Sketcher grid. To choose a different orientation for the edge, click the arrow on the right side of the dialog box and choose an orientation from the list that appears.



Tip: If the part or assembly does not have an edge that will provide the desired orientation, you can create a datum axis. You can then select the datum axis to control the orientation of the part on the Sketcher grid.

- Abaqus/CAE highlights the selected edge, enters the Sketcher, and rotates the part until the selected planar face or datum plane aligns with the plane of the Sketcher grid and the selected edge aligns with the grid in the desired orientation. Abaqus/CAE highlights the selected faces.

7. Use the Sketcher to sketch the planar partition. From the main menu bar, click **Done** to indicate you have finished sketching the partition.
8. In the prompt area, click **Create Partition**.

   Abaqus/CAE creates the partition.

Additional information

- *The Sketch module*
- *Partitioning techniques*
- *Methods for partitioning cells*

The Query toolset

The Query toolset allows you to obtain information about your model.

In this section:

- *Understanding the role of the Query toolset*
- *Querying the model*

Understanding the role of the Query toolset

The Query toolset allows you to obtain information about your model.

In most cases Abaqus/CAE displays the requested information in the message area, and the same information is written to the replay file. Select **Tools**->**Query** from the main menu bar to use the Query toolset, or select the query tool in the Query toolset.

The **Query** dialog box is split into two sections. The top section of the dialog box contains general queries that are available in each module except the Job module, where the Query toolset is not available at all. The bottom portion of the dialog box contains module-specific queries; as you switch between modules, Abaqus/CAE displays queries that are appropriate for the contents of the current module.

In this section:

- *General queries*
- *Module-specific queries*

General queries

You can always use the Query toolset to obtain general information about the model, regardless of which module you are using, although the Query toolset is not available in the Job module.
The **General Queries** in the Visualization module are similar to those in the other modules, but they are discussed separately in *Querying the model in the Visualization module* due to differences in some selection methods and the inclusion of undeformed and deformed model data in the results. For all other modules, the items under **General Queries** in the Query toolset provide the general information discussed below.

In most cases Abaqus/CAE displays the requested information in the message area, and the same information is written to the replay file. Select **Tools->Query** from the main menu bar to use the Query toolset, or select the  tool in the Query toolset.

The **Query** dialog box is split into two sections. The top section of the dialog box contains general queries that are available in each module except the Job module, where the Query toolset is not available at all. The bottom portion of the dialog box contains module-specific queries; as you switch between modules, Abaqus/CAE displays queries that are appropriate for the contents of the current module.

For tools that require meshes to be available, you can use the **View->Part Display Options** or **View->Assembly Display Options** menu items to display the native mesh.

Point/Node

Coordinates of a selected point or node. You can select a local CSYS in which to display the coordinates.

Distance

Distance between two selected points, nodes, edges, faces, or a combination of these objects. You can select a local CSYS in which to display the coordinates of the picked points and the components of the result.

Angle

The angle between two edges or faces, between an edge and a face, or between three points

Feature

For a selected feature:

- Feature name
- Description
- Status (if the feature is suppressed or if it failed to regenerate)
- Parent feature names
- Child feature names
- Parameters

Shell element normals

Display shell/membrane normal directions

Beam element tangents

Display beam/truss tangent directions

In addition, if the current viewport contains a mesh, the Query toolset provides the following information:

Mesh stack orientation

For hexahedral, wedge, and quadrilateral elements that you can use in a continuum shell, cohesive, cylindrical, or gasket mesh, Abaqus/CAE indicates the mesh stack orientation. For hexahedral and wedge elements, Abaqus/CAE colors the top face brown and the bottom face purple. For quadrilateral elements, arrows indicate the orientation of the elements. In addition, Abaqus/CAE highlights any element faces and edges that have inconsistent orientation.



Note:

The query results do not account for changes made in the mesh stack orientation while defining a solid composite layup or a composite shell layup in the Property module.

Mesh

For an assembly, part or part instance, geometric region, or element:

- The total number of nodes and elements in the selected area
- The number of elements for each element shape

By default, Abaqus/CAE displays mesh information in the message area, but you can display this information in tabular format in the **Mesh statistics** dialog box by toggling on **Display detailed report** in the prompt area. The **Mesh statistics** dialog box also enables you to display mesh information by part instance or by element type.

Element

For a selected element:

- Element label
- Element topology; for example, linear hexahedron
- Abaqus element name; for example, C3D8I
- Nodal connectivity

Mesh gaps/intersections

For a selected part or part instance:

- Display element edges of boundary faces with incompatible interfaces
- Display element edges of boundary faces with cracks or gaps
- Display element edges of boundary faces that intersect other faces

Mass properties

For an assembly, selected part or part instance, geometric region, solid element, shell, solid face, beam, or truss, Abaqus/CAE returns some or all of the following information:

- Surface area (displayed only for shell elements and for solid faces)
- Area centroid
- Volume
- Volume centroid
- Mass
- Center of mass
- Moments of inertia about the center of mass or about a specified point

For more information about this query, see *Querying mass properties*.

Geometry diagnostics

- Invalid, imprecise, or small geometry
- Topology

Sets

For a selected set:

- Geometry set: components, index, instance name, set name, set type, total number of components
- Element set: element label, element type, connectivity, instance name, set name, set type, total number of elements
- Node set: node label, nodal coordinates, elements sharing the node, instance name, set name, set type, total number of nodes

Surfaces

For a selected surface:

- Geometry surface: index, side, instance name, surface name, surface type, total number of surfaces
- Mesh surface: element label, element type, connectivity, side, instance name, surface name, surface type, total number of faces

Geometry face normal

The surface normal components in the global Cartesian coordinate system of a planar face or, for a curved surface, at a selected point on the face. You can select a local CSYS in which to display the normal. If typing coordinates of an evaluation point on a curved surface, you should enter the coordinates in the local CSYS, if selected.

Element face normal

The surface normal components in the global Cartesian coordinate system of the element face. You can select a local CSYS in which to display the normal.

Advanced distance

Whereas the regular distance query allows you to find the distance between two selected single entities, the advanced distance query allows you to find the minimum distance between groups of entities. These entities can be nodes, elements, vertices, edges, faces, or cells, and you can select them from the viewport or by set. You can select a local CSYS in which to display the coordinates of the closest points and the components of the result.

For more information, see *Obtaining general information about the model*.

- *Customizing mesh display*

Module-specific queries

In addition to general queries, you can query the model for information specific to the module you are using. Abaqus/CAE displays these module-specific queries at the bottom of the Query toolset under, "***Modulename*** **Module Queries**." The Query toolset can provide the following module-specific information:

Part module

The items under **Part Module Queries** provide the following module-specific information about the current part:

Part attributes

- Name
- Modeling space
- Type (deformable or rigid body)

Regeneration warnings

If any sets or surfaces in the selected part cannot be regenerated because their underlying geometry has been modified or deleted, Abaqus/CAE displays the set or surface name, the original number of faces, and the number of faces found during the query.

Substructure statistics

Abaqus/CAE displays the following information about the selected substructure part: the number of retained nodes, eigenmodes, and substructure loads in the part; the availability of the recovery matrix, gravity load vectors, reduced mass matrix, reduced structural damping matrix, and reduced viscous damping matrix in the substructure; and mass properties of the substructure.

For more information, see *Using the Query toolset in the Part module*.

Property module

The items under **Property Module Queries** provide the following module-specific information about the current part:

Section assignments

Sections assigned to a selected region

Regions missing sections

Regions that require a section assignment

Beam orientations

Beam orientations assigned to a selected wire region (Abaqus/CAE displays the ($\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{t}$) axis system on the selected wire region)

Material orientations

Material orientations assigned to a selected region

Rebar orientations

Rebar reference orientations assigned to a selected region

Ply stack plot

Abaqus/CAE creates a new viewport and displays a graphical representation of a core sample through a region of a composite layup or composite section. The image shows the plies in the layup along with details of each ply, such as its fiber orientation, thickness, reference plane, and integration points.

Disjoint ply regions

Abaqus/CAE displays in the message area the names of the composite layups and the plies within them that contain disjoint regions.

For more information, see *Using the Query toolset to obtain assignment information*, and *Viewing a ply stack plot*.

Assembly module

The items under **Assembly Module Queries** provide the following module-specific information about a selected part instance:

Instance attributes

Name, type, and modeling space

Instance position

- Position of the origin relative to the global coordinate system
- Sum of the translations and rotations applied to the instance
- Number of translational and rotational constraints applied

For more information, see *Using the Query toolset to query the assembly*.

Step module

The Query toolset provides only general information in the Step module.

Interaction module

The item under **Interaction Module Queries** provides the following module-specific information about a selected wire:

- Connector assignment information

For more information, see *Using the Query toolset to obtain connector assignment information*.

Load module

The Query toolset provides only general information in the Load module.

Mesh module

The items under **Mesh Module Queries** provide the following module-specific information about a part or part instance:

- Free/Non-manifold edges
- Unmeshed regions
- Unassociated geometry

For more information, see *Obtaining mesh information*.

Job module

None of the Abaqus/CAE toolsets are available in the Job module.

Visualization module

The items under **Visualization Module Queries** provide the following module-specific information:

- Probe values. Abaqus/CAE displays information in the **Probe Values** dialog box as you move the cursor around the current viewport. Probing a model plot displays model data and analysis results; probing an *X*–*Y* plot displays *X*–*Y* curve data. For more information, see *Probing the model*.
- Stress linearization. Stress linearization is the separation of stresses through a section into constant membrane and linear bending stresses. Abaqus/CAE performs stress linearization calculations and displays the results in the form of an *X*–*Y* plot. For more information, see *Calculating linearized stresses*.
- Active elements or nodes. Abaqus/CAE displays the label numbers of all of the active nodes or active elements in the current viewport. For more information, see *Querying active node or element labels*.
- Ply stack plot. Abaqus/CAE creates a new viewport and displays a graphical representation of a core sample through a region of a composite layup or composite section. The image shows the plies in the layup along with details of each ply, such as its fiber orientation, thickness, reference plane, and integration points. For more information, see *Viewing a ply stack plot*.

Sketch module

The items under **Sketch Module Queries** provide the following information about a selected constraint or sketch:

Constraint

- Constraint type
- Constrained entity names

In addition, the constrained entities are highlighted in the sketch.

Detail

- Number of geometries

- Number of vertices
- Number of constraints
- Number of dimensions
- Number of unconstrained degrees of freedom

Querying the model

This section describes how you use the Query toolset to obtain information about your model.

In this section:

- *Using the Query toolset to query the model*
- *Obtaining general information about the model*
- *Querying mass properties*
- *Using the geometry diagnostic tools*

Using the Query toolset to query the model

You can use the Query toolset to obtain general information about the geometry of the model in the current viewport and the features that define the model.
In addition, you can obtain information specific to the module you are using; for example, if you are using the Mesh module, the Query toolset can provide information on the mesh, nodes, and elements.

1. From the main menu bar, select **Tools->Query**.

   Tip: You can also query the model by clicking the tool in the Query toolset.

   Abaqus/CAE displays the **Query** dialog box.

2. From the **Query** dialog box, select one of the following:

   - **General Queries**
     - **Point/Node**
     - **Distance**
     - **Angle**
     - **Feature**
     - **Shell element normals**
     - **Beam element tangents**
     - **Mesh stack orientation**
     - **Mesh**
     - **Element**
     - **Mesh gaps/intersections**
     - **Mass properties**
     - **Geometry diagnostics**
     - **Sets**
     - **Surfaces**
     - **Geometry face normal**
     - **Element face normal**

     For more information, see *Obtaining general information about the model*.
   - Module-specific queries. For a complete list of the information available in each module, see *Understanding the role of the Query toolset*.

3. For shell normals, beam tangents, mesh gaps, and mesh stack orientation, Abaqus/CAE uses arrows and highlighting to display the requested information in the viewport.

   For all other queries, Abaqus/CAE displays prompts in the prompt area to guide you through the rest of the procedure. Abaqus/CAE displays the requested information in the message area. To resize the message area, drag the top edge; to see information that has scrolled out of the message area, use the scroll bar on the right side. The same information that appears in the message area is also written to the replay file.

4. Click the cancel button  in the prompt area when you have finished requesting information.

Additional information

- *Obtaining general information about the model*
- *Understanding the role of the Query toolset*

Obtaining general information about the model

You can use the Query toolset to obtain general information about your model including point/node coordinates; element labels, topology, and nodal connectivity; and mass properties.

To obtain general information about your model, select **Tools->Query** from the main menu or click the  tool in the Query toolset. For tools that require meshes to be available, you can use the **View->Part Display Options** or **View->Assembly Display Options** menu items to display the native mesh. Select one of the following from the **General Queries** field in the **Query** dialog box that appears:

Point/Node

Select a point or node. Abaqus/CAE displays the *X*-, *Y*-, and *Z*-coordinates of the point. You can select a local CSYS in which to display the coordinates.

Distance

Select two points or nodes. In the Part, Property, and Mesh modules you can select points, edges, or faces. You can select a local CSYS in which to display the coordinates of the picked points and the components of the result. Abaqus/CAE displays the following:

- The *X*-, *Y*-, and *Z*-coordinates of each entity.
- The absolute distance between the entities.
- The *X*-, *Y*-, and *Z*-components of the vector between the two entities.

The distance between non-point entities is always the shortest distance between two points within the entities.

Angle

Select **Face/Curve** or **3 Points** in the prompt area.

- If you selected **Face/Curve**, select two edges or two faces or select an edge and a face. Abaqus/CAE displays one of the following, depending on your selection:
  - The angle between the two edges.
  - The angle between the normals to the face.
  - The angle between the edge and the normal to the face.

  When you select an edge, Abaqus/CAE displays an arrow along the edge. When you select a face, Abaqus/CAE displays an arrow on the face that indicates the normal to the face. The angle is defined as the angle that must be swept by one edge or face to align the two arrows. The angle is always positive and less than or equal to 180°.
- If you selected **3 Points**, select three points/nodes or enter node labels.

Feature

Click mouse button 3 on a feature in the Model Tree, and select **Query** from the menu that appears. Abaqus/CAE displays the following information about the selected feature:

- Name and description; for example, solid extrude.
- Status, if the feature is suppressed or if it failed to regenerate.

- The name of its parent, if any.
- The names of its children, if any.
- The value of any parameters that define the feature.

Shell element normals

- For parts with shell regions, Abaqus/CAE displays the part or assembly using the shaded render style. The side of the shell where the surface normal coincides with the shell normal (top face) is colored brown; the opposite side (bottom face) is colored purple.
- For axisymmetric parts with wire regions, Abaqus/CAE displays cyan arrows indicating the directions of the normals.

You can use the **Assign** menu in the Property module or the **Mesh->Orientation->Normal** menu in the Mesh module to reverse the normal directions. For more information, see *Assigning shell/membrane normal directions*.

Beam element tangents

Abaqus/CAE displays cyan arrows indicating the directions of the beam tangents.

You can use the **Assign** menu in the Property module or the **Mesh->Orientation->Normal** menu in the Mesh module to reverse the tangent directions. For more information, see *Assigning beam/truss tangent directions*.

Mesh stack orientation

You can use this tool to query only hexahedral, wedge, and quadrilateral elements because these are the only elements that can be stacked to form a continuum shell, cohesive, cylindrical, or gasket mesh. For hexahedral and wedge elements, Abaqus/CAE colors the top face brown and the bottom face purple. Similarly, arrows indicate the orientation of quadrilateral elements. In addition, Abaqus/CAE highlights any element faces and edges that have inconsistent orientation. For more information, see *Creating a model with cohesive elements using geometry and mesh tools*, *Continuum shells*, *Swept meshing of cylindrical solids*, and *Gaskets*.

If the region is a native Abaqus mesh, you can change the mesh stack orientation by changing the direction of the sweep path. For solid regions, you can assign a new mesh stack orientation independent of the sweep path. If the region is an orphan mesh, you can use the Edit Mesh toolset to change the mesh stack orientation. For more information, see *Specifying the sweep path*, *Applying a mesh stack orientation*, and *Orienting the stack direction*, respectively.



Note:

The query results do not account for changes made in the mesh stack orientation while defining a solid composite layup or a composite shell layup in the Property module.

Mesh

For an assembly, part or part instance, geometric region, or element, Abaqus/CAE displays the following:

- The total number of nodes and elements in the selected area
- The number of elements for each element shape

By default, Abaqus/CAE displays mesh information in the message area, but you can display this information in tabular format in the **Mesh statistics** dialog box by toggling on **Display detailed report** in the prompt area.

The **Mesh statistics** dialog box also enables you to display mesh information by part instance or by element type.

Element

- Element label
- Element topology; for example, linear hexahedron
- Abaqus element name; for example, C3D8I
- Nodal connectivity.

Mesh gaps/intersections

Select the mesh parts or part instances, and enter the maximum distance between a node and an element face. Abaqus/CAE highlights element edges that intersect the boundary faces and element edges that are closer than the specified distance to boundary faces.

Mass properties

This query displays information about the surface area, area centroid, volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point for your selection. For more information, see *Querying mass properties*.

Geometry diagnostics

- Invalid, imprecise, or small geometry
- Topology

Sets

Select a set from the list of sets provided. If desired, toggle on **Highlight set** to highlight the set in the viewport. Click **Done**.

Abaqus/CAE displays the following information about the selected set in the message area, depending on your selection:

- Geometry set: components, index, instance name, set name, set type, total number of components
- Element set: element label, element type, connectivity, instance name, set name, set type, total number of elements
- Node set: node label, nodal coordinates, elements sharing the node, instance name, set name, set type, total number of nodes

Surfaces

Select a surface from the list of surfaces provided. If desired, toggle on **Highlight surface** to highlight the surface in the viewport. Click **Done**.

Abaqus/CAE displays the following information about the selected surface in the message area, depending on your selection:

- Geometry surface: index, side, instance name, surface name, surface type, total number of surfaces
- Mesh surface: element label, element type, connectivity, side, instance name, surface name, surface type, total number of faces

Geometry face normal

Select a face. Select CSYS, if required. For a curved surface, you are prompted to select the point or enter the *X*-, *Y*-, and *Z*-coordinates of the point at which to evaluate the normal. If typing coordinates of an evaluation point on a curved surface, you should enter the coordinates in the local CSYS, if selected. You can select reference points, attachment points, or vertices. The surface normal components in the global Cartesian coordinate system appear in the message area. If you selected a CSYS, the normal is displayed in the given CSYS.

Element face normal

Select an element. Select CSYS, if required. The surface normal components in the global Cartesian coordinate system appear in the message area. If you selected a CSYS, the normal is displayed in the given CSYS.

Additional information

- *Using the Query toolset to query the model*
- *Understanding the role of the Query toolset*
- *Customizing mesh display*

Querying mass properties

The **Mass properties** query displays the surface area and area centroid, volume and volume centroid, mass and center of mass, and moments of inertia about the center of mass or about a specified point for your selection.
The data displayed depend on your selections. For example, if you select a single face of a solid part, Abaqus/CAE displays the surface area and area centroid of the face in the message area. If you select the entire part, Abaqus/CAE displays the volume and volume centroid, mass and center of mass, and its moments of inertia about the center of mass or about a specified point for the part. If the part is a shell, Abaqus/CAE also includes the surface area and area centroid of one side of the shell in the mass properties output.

The mass properties calculated in Abaqus/CAE account for mass being distributed in a continuum throughout the model, whereas the Abaqus solver calculates the mass properties according to the discretization of the model. When compared to the mass properties provided in the data (`.dat`) file of an Abaqus analysis, the values of the moments of inertia obtained in the mass properties query may be substantially different, especially for finite element models with coarse meshes.

In the Visualization module, the mass properties are always calculated based on model data (undeformed shape) even if the deformed shape is displayed in the viewport.

If the region that you query contains mesh occupying the space where geometry exists but that mesh is unassociated with any geometry, the volume/area will effectively be counted twice in the calculations. You should query unmeshed regions and/or unassociated geometry to avoid encountering this behavior.

You cannot query for the mass properties of a part to which an Eulerian section is assigned. However, you can query for the mass properties of an Eulerian part, provided it does not have an Eulerian section assignment.

You can query for the mass properties of a model even when portions of the model are not completely defined. For example, if your selection includes a part instance that does not have a section assignment, Abaqus/CAE includes the area of that part instance in mass properties calculations but excludes the part instance from calculations of volume, volume centroid, mass, center of mass, and moments of inertia. Abaqus/CAE also appends a warning message to the data indicating that the data are incomplete; in this example, the message would indicate that a portion of your selection was missing a section assignment. This approach enables you to investigate aspects of your model at various points during the modeling process without displaying misleading information about mass-related quantities.

Abaqus/CAE returns a warning message with the mass properties data in the following situations:

Geometry not considered by the mass properties query

Abaqus/CAE computes mass properties for shells, solids, trusses, and rotary inertia elements; and in modules other than the Visualization module. Abaqus/CAE does not compute mass properties for beam general section with a generalized section but it does compute mass properties for other profiles. Abaqus/CAE also includes point and nonstructural mass elements in its mass properties calculations. If your selection includes other geometry such as axisymmetric elements, springs, connectors, or gaskets (or, in the Visualization module, if your selection includes point and nonstructural mass elements), Abaqus/CAE excludes that geometry from its calculations. The exclusion of point and nonstructural mass elements from mass properties results in the Visualization module means that mass properties results may be different in the Visualization module than they are in other modules.

Ineligible thickness values

If your selection includes a section definition with a missing thickness value or a thickness value of zero, Abaqus/CAE excludes the selection from calculations of mass, center of mass, and moments of inertia. Similarly, Abaqus/CAE does not consider regions with a variable thickness (defined using a nodal thickness or field thickness) or regions using a thickness value that is not applicable to the corresponding sections specified.

Density not defined or improperly defined

If your selection includes a region in which a value for density is not available or is defined improperly, Abaqus/CAE excludes that region from all mass-related calculations in the mass properties query. The region's area and volume data are included in the final calculations.

Missing section assignment

If your selection includes a region that does not have a section assigned to it, Abaqus/CAE excludes that region from all mass-related calculations in the mass properties query. The region's area and volume data are included in the final calculations.

Inclusion of shell offsets or reinforcements

Abaqus/CAE does not consider shell offset values when calculating the center of volume or surface area, and Abaqus/CAE also excludes any reinforcements in your model from all mass property calculations. If your query includes shell offsets or reinforcements, Abaqus/CAE returns the query results with an appropriate warning message.

Composite sections

When your mass properties query selection includes a part or region with a composite section, Abaqus/CAE calculates mass- and volume-related values in the mass properties query using smeared properties located on the element.

Inclusion of nonstructural mass entities

Abaqus/CAE does not consider nonstructural mass on wires, such as mass-per-length values. In addition, when geometry is not meshed, Abaqus/CAE cannot compute center of mass and moment of inertia values as accurately as it can for meshed geometry.

1. Select **Tools->Query** from the main menu to use the Query toolset, or select the  tool in the Query toolset.
2. Click **Mass properties**.

   Additional options appear in the prompt area.
3. If desired, click **Options** to modify query options.
   a. In the **Mass Properties Query Options** dialog box, you can modify accuracy, density, thickness parameters, and the point about which the moment of inertia is computed. You can click **Reset** to return the options to the previously stored values.
   b. Click **OK** to save your changes and to close the dialog box.
4. If you include a native mesh in your selection, you can query for the mass properties of the geometry or of the native mesh. To query for the mass properties of the native mesh, click **Options** in the prompt area, then toggle on **Use mesh when available for accurate properties** in the dialog box that appears.
5. For selections that include complex geometry, Abaqus/CAE may take a long time to compute the volume properties. You can reduce the computation time by instructing Abaqus/CAE to stop the computation when a certain relative accuracy has been achieved. Click **Options** in the prompt area, then select a relative accuracy of **Low**, **Medium**, or **High** from the dialog box that appears. Higher relative accuracy yields a more accurate result but takes longer to calculate. Relative accuracy settings are available only for geometry selections, not for mesh selections.

6. If your selection includes regions in which thickness or density is either missing or improperly defined, you can supply values for the thickness or density of these regions for the mass properties calculations. Providing these values to the query does not change these values in the model itself.

   Click **Options** in the prompt area, then provide values greater than zero in the appropriate fields.

7. If you want to query mass properties for the entire assembly, select **Query entire assembly** and click **Done**.

   Abaqus/CAE displays the following information in the message area: the volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point. If the assembly includes shell features, Abaqus/CAE reports the surface area and area centroid for one side of the shell faces as well.

8. If you want to query mass properties for the entire part (when the part is displayed in the Part module or Property module), select **Query entire part** and click **Done**.

   Abaqus/CAE displays in the message area the volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point for the part. If the part includes shell features, Abaqus/CAE reports the surface area and area centroid for one side of the shell faces as well.

9. If you want to query mass properties for one or more part instances in an assembly, highlight **Select part instances**, highlight one or more part instances in the viewport, and click **Done**.

   Abaqus/CAE displays in the message area the volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point for the part instance or instances. If the part instance or part instances you query include shell features, Abaqus/CAE reports the surface area and area centroid for one side of the shell faces as well.

10. If you want to query mass properties for one or more geometric regions in an assembly, part, or part instance, highlight **Select geometric regions**, highlight one or more regions in the viewport, and click **Done**.

    Abaqus/CAE displays in the message area the volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point for the region or regions. If the region or regions you query include shell features, Abaqus/CAE reports the surface area and area centroid for one side of the shell faces as well.

11. If you want to query mass properties for one or more faces in an assembly, part, or part instance, highlight **Select geometric regions**, highlight one or more regions in the viewport, and click **Done**.

    a. From the prompt area, highlight **Select geometric regions**.
    b. From the **Selection** toolbar, select **Faces** from the list of object types.
    c. Select one or more faces from the viewport, and then click **Done**.

    Abaqus/CAE displays the total surface area of the selected faces. You can query for multiple shell surfaces or multiple faces, but you cannot perform a query for both shells and geometric faces.

12. If you want to query mass properties for one or more elements in an assembly, part, or part instance, highlight **Select elements**, highlight one or more elements in the viewport, and click **Done**.

    Abaqus/CAE displays in the message area the volume, volume centroid, mass, center of mass, and moments of inertia about the center of mass or about a specified point for the region or regions. If the region or regions you query include shell features, Abaqus/CAE reports the surface area and area centroid for one side of the shell faces as well.

13. If you want to query mass properties for one or more element faces in an assembly, part, or part instance, perform the following steps:

    a. From the prompt area, highlight **Select elements**.
    b. From the **Selection** toolbar, select **Element Faces** from the list of object types.
    c. Select one or more elements faces from the viewport, and then click **Done**.

Abaqus/CAE displays the total surface area of the selected element faces. You can query for multiple element faces and for element faces from multiple shell surfaces or multiple faces, but you cannot perform a query for element faces from both shells and geometric faces.

You can also use general queries, including querying mass properties, in the Visualization module. For more information, see *Querying the model in the Visualization module*.

Using the geometry diagnostic tools

This section describes how you can use the geometry diagnostic tools to locate the following in a part or part instance in the current viewport:

Geometry

- **Invalid entities**: Abaqus/CAE highlights the regions of the part or part instance that have invalid geometry.

  Toggle on **Update validity** to update the status of the selected part or part instance. If an invalid part has become valid, it is marked as valid in the Model Tree. If a valid part has become invalid, Abaqus/CAE displays a warning message that allows you to ignore the invalid status and continue using the part.
- **Imprecise entities**: Abaqus/CAE highlights the regions of the part that have imprecise geometry.

For more information on invalid geometry, see *What is a valid and precise part?*, and *Working with invalid parts*.

Topology

- **Free edges**: Abaqus/CAE highlights the free edges of the part or part instance, including shell and wire edges. A solid part should not contain any free edges.
- **Solid cells**: Abaqus/CAE highlights the solid cells of the part or part instance.
- **Shell faces**: Abaqus/CAE highlights the shell faces of the part or part instance.
- **Wire edges**: Abaqus/CAE highlights the wire edges of the part or part instance. Wire edges are wires, as opposed to edges of a solid or shell feature.

Small geometry

Small entities such as short edges, small faces, and faces with small face corner angles can affect the mesh quality. You can use the following tools to locate these regions:

- **Edges shorter than:**Abaqus/CAE highlights edges shorter than the specified length. The specified length must be 1×10^{-6} or greater.
- **Faces smaller than:**Abaqus/CAE highlights faces smaller than the specified area. The specified area must be 1×10^{-12} or greater.
- **Face corner angles less than:**Abaqus/CAE highlights faces where two edges meet at an angle less than the specified angle. The specified angle must be less than 90°. Abaqus/CAE meshes faces with small corner angles with elements that have a high aspect ratio that affect the mesh quality.

1. From the main menu bar, select **Tools->Query**, or click the  tool in the Query toolset to start the Query toolset.
2. From the **Query** dialog box, select **Geometry diagnostics**. If the viewport contains multiple part instances, select the one that you want to query.

   Abaqus/CAE displays the **Geometry Diagnostics** dialog box.
3. Toggle on the desired geometry diagnostic options and specify a new value, if desired.
4. + Click **Highlight**.

- Abaqus/CAE highlights any regions with the corresponding invalid, imprecise, or small geometry you selected. If the part size is relatively large, short edges and small faces, even though highlighted, may not be visible. To help you find these short edges and small faces, Abaqus/CAE displays markers around the highlighted regions.
- If you selected **Update validity**, Abaqus/CAE also checks and updates the validity status of the part or assembly. The validity checks can be computationally expensive for complex geometry.
- If no regions are found for a particular diagnostic tool, Abaqus/CAE displays a message in the message area.

5. If desired, click **Create Set** in the prompt area to create a geometry set containing the highlighted regions. In the dialog box that appears, enter a name for the set and + click **Continue**.

   The Geometry Edit toolset provides tools to repair regions that are invalid or imprecise. The Geometry Edit toolset allows you to specify a set when you select a region to repair.

   Note:

   You can view geometry diagnostics for the reference representation by toggling on **Apply to reference representation**. However, items from the reference representation will be excluded from created geometry sets because the reference representation cannot be repaired. For more information, see *Understanding the reference representation*.

Additional information

- *Importing and exporting geometry data and models*
- *Editing techniques*
- *The Set and Surface toolsets*

The Reference Point toolset

This chapter describes the Reference Point toolset.

In this section:

- *What is a reference point?*
- *What is a reference point used for?*
- *Creating a reference point*

What is a reference point?

A reference point is a point that you create on a part. You can also create reference points on the assembly. You can position a reference point anywhere in space, and a reference point is useful for creating a point in your model where a vertex is not available; for example, at the center of a hole. In contrast to a vertex a reference point is ignored by the Mesh module when the mesh is generated.

You can use the Reference Point toolset in the Part module to create a reference point that is associated with a part by selecting **Tools->Reference Point** from the main menu bar. When you create the assembly, the reference point appears on each instance of the part in the assembly. A part can include only one reference point, and Abaqus/CAE labels the reference point **RP**. Abaqus/CAE asks you if you want to delete the original point if you try to assign a second point.

Alternatively, you can use the Reference Point toolset in the Assembly, Interaction, or Load modules to create a reference point on the assembly by selecting **Tools->Reference Point** from the main menu bar. In the Interaction module you can use the 

tool in the module toolbox to create a reference point. The assembly can include more than one reference point, and Abaqus/CAE labels them **RP-1**, **RP-2**, **RP-3**, etc. For more information, see *Creating a reference point*. If desired, you can turn off the display of the reference point symbol and the reference point label; for more information, see *Controlling reference point display*.

What is a reference point used for?

You can use a reference point for the following:

- If the part is a deformable planar part that is modeled with generalized plane strain elements, you must create a reference point to indicate the required reference node. For more information, see *Creating generalized plane strain sections*. For more information on generalized plane strain elements, see *Choosing the Element's Dimensionality*.
- If the part is a discrete or analytical rigid part, you use the reference point to indicate the rigid body reference point. Constraints or motion that you apply to the reference point are applied to the entire rigid part.

  The location of the rigid body reference point affects how you prescribe moments or motion; in addition, the location of the rigid body reference point affects the interpretation of moment reactions. If your model includes a dynamic analysis involving rotations, the rotary inertia specification of a rigid body must be made consistent with the location of its rigid body reference point.
- You must refer to a reference point on the assembly when you create a rigid body constraint in the Interaction module. A rigid body constraint constrains the motion of regions of the assembly to the motion of a reference point. You can create and name a set containing the reference point and refer to the set, or you can select the reference point directly from the current viewport. You can also apply loads and boundary conditions to a reference point in the Load module.
- In some cases you will want to position the reference point at the center of mass, which you can find using the Query toolset. You can use the Property module to assign mass and rotary inertia section properties to the reference point. The section can also include optional damping data.
- You can apply constraints to a reference point; for example, equation, coupling, and display body constraints.
- You can use a reference point when you create an assembly-level wire feature in the Assembly module or the Interaction module. A reference point is useful if you want to attach the connector to a point in space. You can create a reference point at the desired location and use a rigid body constraint to attach a part instance to the reference point. When you attach a connector to such a reference point, the connector is effectively attached to the part instance.

Creating a reference point

From the main menu bar, select **Tools->Reference Point** to create a reference point on a part or on the assembly. A part can include only one reference point, and the reference point appears on each instance of the part in the assembly. Alternatively, you can use the Assembly, Interaction, or Load modules to create multiple reference points on the assembly. For more information, see *What is a reference point?*.

1. From the main menu bar, select **Tools->Reference Point**.

   Tip: In the Interaction module you can create a reference point using the  tool in the module toolbox.

2. Select a point to act as the reference point. You can use either of the following techniques to position the reference point:

   - Select any existing vertex from the part, including datum points. If you want to move the reference point, you must edit the vertex or datum point. Alternatively, you can delete the reference point and select a new vertex or datum point.
   - Type the components of a vector representing the *X*-, *Y*-, and *Z*-coordinates of the reference point. If you want to move the reference point, you can use the Feature Manipulation toolset to edit its coordinates.

   Abaqus/CAE displays the reference point at the desired location along with its label. If you create the reference point on a part in the Part module, Abaqus/CAE labels it **RP**. If you create the reference point on the assembly, Abaqus/CAE labels it **RP-1**, **RP-2**, **RP-3**, etc. You can change the reference point label by clicking mouse button 3 on the feature in the Model Tree and selecting **Rename** from the menu that appears. If desired, you can turn off the display of the reference point symbol and the reference point label; for more information, see *Controlling reference point display*.

Additional information

- *Rigid parts*
- *The reference point*
- *Renaming a feature*
- *What is a reference point?*
- *What is a reference point used for?*

The Set and Surface toolsets

When you need to specify a region of your model (for example, to apply a load or to define contact), you can select the region from the viewport or you can define a set or a surface that contains the region. You use the Set and Surface toolsets to create and manage sets and surfaces.

The Set and Surface toolsets are available from the **Tools** menu in the main menu bar in all modules except the Visualization module.

In this section:

- *Understanding the role of the Set and Surface toolsets*
- *Understanding sets and surfaces*
- *Using the Set and Surface toolsets*

Understanding the role of the Set and Surface toolsets

The Set and Surface toolsets are collections of tools that allow you to create and manage sets and surfaces. You can use these tools to perform the following tasks:

Create

Create a set or surface by selecting a group of entities, such as faces and edges.

Edit

Modify a set or surface by selecting entities to add to it or to delete from it.

Rename

Replace the current name of an existing set or surface with a new name.

Delete

Delete a set or surface from the model.

In addition, you can use the Model Tree to merge a group of selected sets and surfaces into a new set or surface. For detailed instructions on creating and manipulating sets and surfaces, see the following sections:

- *Creating, editing, copying, renaming, and deleting sets and surfaces*
- *Creating sets*
- *Creating surfaces*
- *Performing Boolean operations on sets or surfaces*
- *Editing sets and surfaces*
- *Associating objects (such as loads and sections) with sets and surfaces*

Understanding sets and surfaces

This section describes basic concepts concerning sets and surfaces.

In this section:

- *What is a set?*
- *How do part sets and assembly sets differ?*
- *What is a surface?*
- *Regeneration of geometry sets and surfaces*
- *Specifying a particular side or end of a region*

What is a set?

A set is a named region or collection of entities on which you can perform various operations.

For example, once you create a set, you can use it to perform the following tasks:

- Assign section properties in the Property module.
- Create contact pairs with contact node sets and surfaces in the Interaction module.
- Define loads and boundary conditions in the Load module.
- Request output from specific regions of the model in the Step module.

You can create the following types of sets:

Geometry sets

A geometry set contains geometric objects (cells, faces, edges, and vertices) that you have selected from one of the following types of parts or from instances of these parts:

- Native parts (those created using the tools in the Part module)
- Parts that you imported from a file.

You select the entities to include in the set from the current viewport. Depending on the shape and modeling space of the part, you can include any combination of cells, faces, edges, and vertices in a set. However, some procedures can be performed only with certain types of objects. As a result, the set that you select must include only object types that are valid for the procedure. For example, concentrated forces can be applied only to vertices; therefore, the sets to which you apply concentrated forces can include only vertices.

Node and element sets

Node and element sets contain nodes and elements that you have selected. You can create node and element sets from native Abaqus nodes and elements on a part that you have meshed in the Mesh module, from orphan mesh nodes and elements, or from nodes and elements on any instances of parts. A set can include nodes or elements from a single part or from multiple part instances. Sets that you create using the Set toolset can include either nodes or elements but not both. However, you can create sets containing a mixture of nodes and elements by merging sets or by importing an output database or an input file containing multiple sets with the same name. Native nodes and elements within sets allow you to request output or add loads to specific areas without deleting the mesh and partitioning the geometry. However, any changes to the mesh—including regenerating it from replay or journal files—may invalidate or change the native portion of mesh sets.

For more information about mesh parts and orphan mesh nodes and elements, see *What kinds of files can be imported and exported from Abaqus/CAE?*, *Importing a model from an Abaqus input file*, *Importing a model from an output database*, and *Creating a mesh part*.

If you rename or delete a set, any objects associated with the set, such as sections or loads, become invalid. However, if you change the name of a renamed set back to its original name or if you recreate a deleted set using its original name, objects associated with that set are restored.

Additional information

• *Using the Set and Surface toolsets*

How do part sets and assembly sets differ?

Sets that you create from a part or sets that you create from a part instance in the assembly are used differently:

Part sets

Part sets are sets that you created in the part-related modules—Part or Property. The Mesh module also acts as a part-related module when you select the **Part** object from the context bar. When you import an orphan mesh from an output database, you also automatically import any sets as part sets. Part sets appear in the Model Tree in a **Set** container under the part with which they are associated. Only part sets are visible in the **Set Manager** in the Part and Property modules. You do not use part sets directly in the Part module; however, in the Property module you can assign sections to regions specified by part sets. If you assign a section to a region defined by a part set, Abaqus applies the section assignment to all instances of that part in the assembly.

When you instance a part in the Assembly module, you can refer to any part sets that you previously created; however, the assembly-related modules provide only read-only access to these sets, and you cannot access part sets from the **Set Manager** in assembly-related modules. Sets from an instanced part appear in the Model Tree under the assembly. You can select an eligible part set during a procedure (while applying a load or boundary condition, for example) by clicking the **Set** button on the far right side of the prompt area and selecting the set from the **Region Selection** dialog box that appears. Abaqus names the sets *part_instance_name.set_name*. You cannot access part sets from the **Set Manager** in an assembly-related module.

Assembly sets

Assembly sets are sets that you created in the assembly-related modules—Assembly, Step, Interaction, or Load. The Mesh module also acts as an assembly-related module when you select the **Assembly** object from the context bar. Assembly sets appear in the Model Tree in a **Set** container under the assembly along with any sets from an instanced part. Only assembly sets are visible in the **Set Manager** in the assembly-related modules. You can use assembly sets to indicate, for example, regions of the assembly where you would like to apply a load or boundary condition or to obtain output. Assembly sets can include regions from multiple part instances.

An assembly set refers to the assembly itself and not to the individual part instances. As a result, Abaqus/CAE does not delete an assembly set if you delete a part instance contained in the set. You must manually delete the assembly set.

A single part cannot contain a geometry set, a node set, or an element set with the same name as an existing set; however, different parts can contain sets with the same name. All assembly set names must be unique.

Additional information

- *What is a set?*
- *Using the Set and Surface toolsets*

What is a surface?

Surfaces are collections of faces and edges or collections of element faces and edges. You can select a surface when a procedure is expecting a face; for example, when you are applying distributed loads, such as pressure loads, and defining contact interactions. You can select an interior surface; for example, when you are using the solid offset mesh tool.

When you create a surface on a shell model, you must select which side of the surface is the desired face; you can also select both faces. Similarly, if you create a surface on a wire model, you must select which end of the wire is the desired face; you can also select the circumference of the wire. For more information, see *Specifying a particular side or end of a region*.

You can define a surface that includes edges of a shell or element edges. You can use edge-based surfaces in Abaqus/Explicit general contact interactions. You can also use edge-based surfaces to tie two shells along two edges or to tie two shells that intersect to form a "T."

You can create two different types of surfaces:

Geometry surfaces

Create a geometry surface by selecting geometric objects (faces and edges) from native or imported geometry. When the analysis input file is created, the surface definition in the input file specifies the element edges (in the case of axisymmetric part instances) or faces that are associated with the surface geometry.

Mesh surfaces

Create a mesh surface by selecting element faces (for three-dimensional regions) or edges (for two-dimensional regions) from meshes. Adding surfaces to the mesh allows you to request output or add loads to specific areas without deleting the mesh and partitioning the geometry. However, native element faces or edges within mesh surfaces may be invalidated or the content may change if you make any changes to the mesh—including regenerating the mesh from replay or journal files.

If you rename or delete a surface, any objects associated with the surface, such as loads or interactions, become invalid. However, if you change the name of a renamed surface back to its original name or if you recreate a deleted surface using its original name, objects associated with that surface are restored.

If you use the Virtual Topology toolset to create virtual faces and edges, you can select those virtual faces and edges when you create sets and surfaces. In addition, if an existing set or surface contains an edge or vertex that you ignored using the Virtual Topology toolset, the edge or vertex is removed from the set or surface. Similarly, if an existing set or surface contains faces or edges that you combined, Abaqus/CAE replaces the original faces and edges with the new combined faces and edges. This is true only if all of the faces and edges that you combined are members of the same set. For more information, see *What can I do with a part or a part instance containing virtual topology?*.

Additional information

- *Selecting interior surfaces*
- *Using the Set and Surface toolsets*
- *Defining Contact Pairs in Abaqus/Standard*
- *Defining Contact Pairs in Abaqus/Explicit*
- *Mechanical Contact Properties*

Regeneration of geometry sets and surfaces

Geometry sets and surfaces are regenerated automatically when you change the underlying geometry of the model. However, if you change the geometry of the model significantly, objects that you had previously included in sets or surfaces may be unidentifiable.

For example, if you delete or suppress a feature of a part, sets and surfaces associated with that feature are altered:

- If components of the feature were the only objects in a set or surface, the set or surface still appears in the **Set Manager** but is empty. You can edit the set so that it includes new geometry.
- If components of the feature were included with other objects in a set or surface, the set or surface no longer contains the components of the feature but continues to contain all of the other objects.
- If you suppress and then resume a feature, the sets and surfaces associated with that feature are also restored, as well as any objects assigned to those sets and surfaces, such as loads or interactions.

Additional information

- *What is a set?*
- *What is a surface?*
- *Using the Set and Surface toolsets*

Specifying a particular side or end of a region

When you create a surface definition from a shell, a wire, or an internal face of a three-dimensional part, you must specify which side of the part you want to include in the surface definition. When you select sides from a shell or face, Abaqus/CAE uses different colored faces to indicate the two sides and prompts you to select the color corresponding to the side that you want to select. During this procedure, the model display becomes translucent to make obstructed or interior sides visible. When you select sides of a wire, Abaqus/CAE uses different colored arrowheads to indicate the two sides of the part.

When you edit a surface definition, the sides are colored based on your previous selections as described below.

Selecting sides from a shell

To distinguish the sides of a shell, Abaqus/CAE colors one side of the shell brown and the other side purple. You select a specific side by clicking either the **Brown** or **Purple** button in the prompt area (see *Figure 1*).



Figure 1: Select the side of the surface.

You also have the option of selecting **Both sides**, which defines a double-sided surface that includes both the brown and the purple sides of the shell. Double-sided surfaces can be used only in the following situations:

- Contact interactions using three-dimensional shells, membranes, and rigid bodies in an Abaqus/Explicit analysis
- Contact interactions using the small-sliding, surface-to-surface contact formulation in an Abaqus/Standard analysis

Consider the conical shell in *Figure 2*.

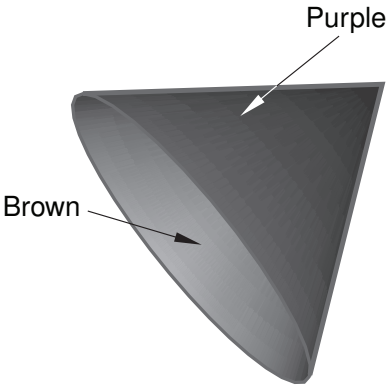


Figure 2: Colors indicate the side of a shell.

Abaqus/CAE displays the exterior side of the cone in purple and the interior side of the cone in brown. Therefore, to select the exterior of the cone, you would click **Purple** in the prompt area; to select the interior, you would click **Brown**.

If you select more than one shell face as part of a surface definition, you can control the color coding for each face individually. Abaqus/CAE includes a **Flip a surface** option that allows you to reverse the orientation of any individual face before creating the surface definition (see *Figure 3*).



Figure 3: The Flip a surface option becomes available for multi-faced shell surfaces.

Use this option to make all of the desired sides the same color, then select that color from the prompt area. If you select **Both Sides**, Abaqus/CAE selects both sides of all selected faces regardless of any surfaces that you previously flipped.

In the model in *Figure 4* both rounded ends are selected as part of the surface.

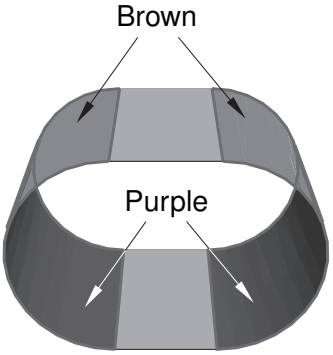


Figure 4: Two faces are included in the surface selection.

Initially, Abaqus/CAE colors the convex sides of both faces brown. However, the desired surface actually consists of one convex face and one concave face. To achieve this surface, click **Flip a surface**, and select the face on the right side of the model. Abaqus/CAE reverses the color-coding for only this face, as shown in *Figure 5*.

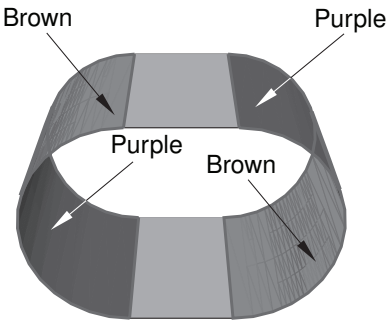


Figure 5: Flipping a surface.

Click **Brown** in the prompt area; Abaqus/CAE defines the surface on all of the brown face sides in the model.

When editing an existing surface definition, you can reselect sides by choosing either **Brown** or **Purple**. **Brown** indicates the side that you used to create the surface previously, and **Purple** indicates the other side. For example, if you selected the **Purple** side to create the surface previously, that side now appears as **Brown** during editing.

Selecting sides on interior surfaces

When you select an internal face to create a surface, the surface side is ambiguous. For example, the highlighted face in *Figure 6* could be either the right side of cell A or the left side of cell B.

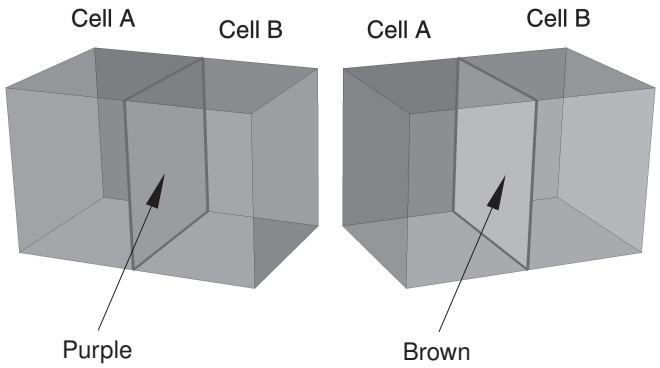


Figure 6: Defining a surface on an internal face.

Abaqus/CAE uses the same brown/purple color-coding interface discussed above to determine which side of the surface you intended to select. Click **Purple** to define a surface on the right side of cell A; click **Brown** to define a surface on the left side of cell B.

Selecting sides from a wire

If you are selecting a surface of a three-dimensional wire part instance, Abaqus/CAE displays the wire in red along with the arrowheads shown in *Figure 7*.

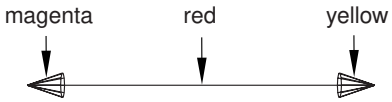


Figure 7: Arrows indicate the ends of a three-dimensional wire.

You can select the end with the magenta arrowhead, the end with the yellow arrowhead, or the circumferential surface of the part, which is indicated by the red wire (see *Figure 8*).



Figure 8: Select the end or the circumference of the surface.

If the wire part is two-dimensional, Abaqus/CAE distinguishes the two sides of the surface using the arrowheads shown in *Figure 9*.

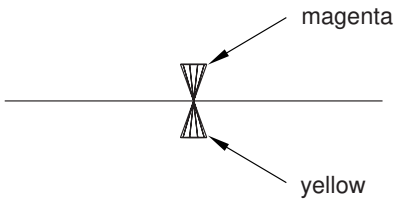


Figure 9: Arrows indicate the top and bottom surface of a two-dimensional wire.

You select an arrowhead color to determine the surface side, as shown in *Figure 10*.



Figure 10: Select the side of the surface.

If the part is an axisymmetric shell, the inside and the outside surface of the part are indicated with arrowheads (see *Figure 11*). The selection options are the same as in *Figure 10*.

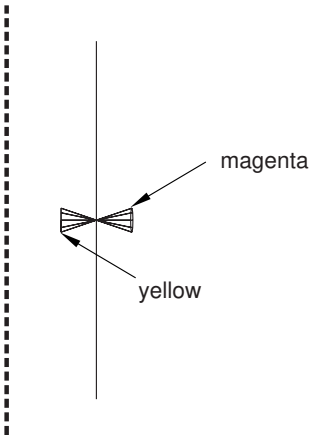


Figure 11: Arrows indicate the inside and the outside of an axisymmetric shell.

When editing an existing surface definition created on a wire, you can reselect sides by choosing either the **Magenta** or **Yellow** arrowheads. **Magenta** indicates the side that you used to create the surface previously, and **Yellow** indicates the other side. For example, if you selected the **Yellow** side to create the surface previously, that side now appears as **Magenta** during editing

Additional information

- *What is a surface?*
- *Using the Set and Surface toolsets*
- *Selecting interior surfaces*

Using the Set and Surface toolsets

This section outlines the various tasks you can perform with the Set and Surface toolsets and describes the procedures for each.

In this section:

- *Creating, editing, copying, renaming, and deleting sets and surfaces*
- *Creating sets*
- *Creating unsorted node sets*
- *Creating surfaces*
- *Performing Boolean operations on sets or surfaces*
- *Editing sets and surfaces*
- *Associating objects (such as loads and sections) with sets and surfaces*

Creating, editing, copying, renaming, and deleting sets and surfaces

You can create, edit, rename, and delete set or surface definitions.

To create, edit, rename, and delete set or surface definitions, use one of the following:

- The **Create**, **Edit**, **Copy**, **Rename**, and **Delete** items listed under the **Tools->Set** and **Tools->Surface** submenus in the main menu bar.

  The **Edit**, **Copy**, **Rename**, and **Delete** menu items contain submenus listing all of the sets or surfaces in the current model. The **Set** submenus vary depending on which module you are in; in the Part and Property modules, only part sets are listed. In the Mesh module, either part or assembly sets may be listed depending on whether a part or the assembly is selected in the context bar. In all other modules, only assembly sets are listed.
- The set and surface managers. These managers contain functions identical to those listed under the **Set** and **Surface** submenus but with convenient browsers that display sets and surfaces along with their type (such as geometry or mesh). The **Set Manager** displays part sets in the Part and Property modules, part or assembly sets—depending on the current object selection in the context bar—in the Mesh module, and assembly sets in all other modules.

  Note: Sets and surfaces that are empty as a result of feature or mesh modification will still appear in the lists. See *Regeneration of geometry sets and surfaces*, for more information.

  The Model Tree displays an exclamation point (!) next to the names of any empty sets or surfaces. You can also position the cursor over a set or surface name in the Model Tree to view a tooltip that includes the number of items it contains.

  To display a manager, select **Tools->Set->Manager** or **Tools->Surface->Manager** from the main menu bar.

  You can copy imported sets and surfaces through the Model Tree, but imported sets and surfaces cannot be copied through **Tools->Set** and **Tools->Surface** submenus or through set and surface managers.



Warning: When you delete a set or surface, any objects associated with that set or surface, such as loads or interactions, become invalid.

Additional information

• *Managing objects*

Creating sets

The Set toolset allows you to create geometry sets and node and element sets.

You create a geometry set when you are working on a native part created by Abaqus/CAE. You can create node and element sets from native Abaqus nodes and elements on a part that you have meshed in the Mesh module, from orphan mesh nodes and elements, or from nodes and elements on any instances of parts. For more information, see *What is a set?*.

In addition, you can create either a part set or an assembly set. You create a part set when you are working with a part in the Part module or the Property module. When you instance the part in the Assembly module, you can refer to any part sets that you previously created. Finally, you create an assembly set when you are working in the assembly-related modules—Assembly, Interaction, Load, and Mesh. For more information, see *How do part sets and assembly sets differ?*.

1. From the main menu bar, select **Tools->Set->Create**.

   Tip: You can also click **Create** in the **Set Manager**.

   The **Create Set** dialog box appears. The dialog box lists the types of sets that you can create based on the part or assembly displayed in the current viewport.

2. In the dialog box:
   a. Enter a name for the set. Part sets created on different parts can use the same name; however, assembly set names must be unique.
   b. If necessary, select the set type.

      By default, a node set is always sorted in the order of the node labels. You can toggle on **Make unsorted set** to create an unsorted node set and easily control the node set content. For more information, see *Creating unsorted node sets*.

   c. Click **Continue**.

   If you create a node or element set on a native part, Abaqus/CAE automatically displays the mesh if it is not already visible. If you create a node or element set on a part instance or on a part that contains both native and orphan mesh components, you must use the **Show native mesh** tool in the **Visible Objects** toolbar to display and select items from the native portion of the mesh.

3. Select the entities in the viewport that you want to include in your set.

   Tip: You can use the **Selection** toolbar to limit the types of entities that you can select in the viewport. For more information, see *Selecting objects within the viewport*.

   When you are finished selecting entities, click mouse button 2.

Additional information

- *What is a set?*
- *Selecting objects within the viewport*

Creating unsorted node sets

You can use the Set toolset to create unsorted node sets.
By default, a node set is always sorted in the order of the node labels. You can create an unsorted node set and easily control the node set content.

You can create unsorted node sets from native Abaqus nodes on a part that you have meshed in the Mesh module, from orphan mesh nodes, or from nodes on any instances of parts.

1. From the main menu bar, select **Tools->Set->Create**.

   Tip: You can also click **Create** in the **Set Manager**.

   The **Create Set** dialog box appears. The dialog box lists the types of sets that you can create based on the part or assembly displayed in the current viewport.

2. In the dialog box:
   a. Enter a name for the set. Part sets created on different parts can use the same name; however, assembly set names must be unique.
   b. Select **Node** as the set type, and toggle on **Make unsorted set**.
   c. Click **Continue**.

   If you create a node set on a native part, Abaqus/CAE automatically displays the mesh if it is not already visible. If you create a node set on a part instance or on a part that contains both native and orphan mesh components, you must use the **Show native mesh** tool in the **Visible Objects** toolbar to display and select items from the native portion of the mesh.

3. In the **Create Unsorted Node Set** dialog box that appears, specify the nodes for the set.

   Use either of the following techniques:

   Select the nodes from the viewport

   a. Select one of the following methods to insert selections in the set definition table:

      - Select **Individual** to select nodes individually.
      - Select **Along edges** to select nodes along a feature edge. Abaqus/CAE proposes a direction for the node selection (indicated by an arrow) and prompts you to confirm the direction. If necessary, you can flip the direction.
      - Select **Free** to select nodes using any of a variety of general selection methods as described in *Selecting objects within the current viewport*.

      Click **Add Before** or **Add After** to begin making selections.

   b. If desired, repeat the preceding step to add additional nodes.

      The **Add Before** and **Add After** buttons determine where your selections will be added in relation to the currently highlighted line in the set definition table.

   c. Click **Done** in the prompt area.

      Abaqus/CAE displays the **Create Unsorted Node Set** dialog box; the part instances and nodes in your set are listed in the set definition table.

   Enter the node labels directly into the table

   In the **Set Definition** table, use standard keyboard and mouse editing techniques to insert, modify, or delete part instance names and node label expressions. If you do not know the part instance names in the model, use the Query toolset to query nodes in the viewport. For special table editing options, press mouse button 3. (For more information, see *Entering tabular data*.) A node label expression is either of the following:

- A single node label; for example, 5.
- A range of nodes that you specify with a starting and ending node label optionally followed by an increment; for example, 5:10 or 5:10:2.

Select rows in the set definition table, and toggle on **Highlight nodes in selected row** to view node selections.

4. If desired, order some or all of the contents of the set definition table according to a sorting expression.
   a. Select one or more rows in the set definition table.
   b. Specify the following settings to order the contents of the selected rows:
      - **Sorting order**: select **Ascending** or **Descending**.
      - **Local system**: select or create a coordinate system, if desired.
      - An expression in the field provided. Any valid Python expression can be used, referencing one or more valid parameters: "Label", "X", "Y", "Z", "R", "Th", or "P". (The actual valid parameters depend on the local system selected, if any, and are indicated in the dialog.) For example, the simple expression "X" will sort the indicated nodes according to the X coordinate of each node. More complex examples include "X*Y" and "Z if Label < 1000 else Label".
   c. Click **Apply**. The contents of the selected rows are ordered according to the expression and sorting order. Rows themselves are not reordered; you can use the right mouse button menu in the table to move rows up or down as needed.
5. Click OK to commit the set definition.

Creating surfaces

The Surface toolset allows you to create geometry or mesh surfaces. Geometry surfaces can consist of geometric faces and edges; mesh surfaces can consist of element faces and edges.

1. From the main menu bar, select **Tools->Surface->Create**.

   Tip: You can also click **Create** in the **Surface Manager**.

   The **Create Surface** dialog box appears. The dialog box lists the type of surface you are creating.

2. In the dialog box:
   a. Enter a name for the surface. Surfaces created on different parts can use the same name; however, assembly surface names must be unique. For information on naming objects, see *Using basic dialog box components*.
   b. If necessary, select the surface type.
   c. Click **Continue**.

   If you create a mesh surface on a native part, Abaqus/CAE automatically displays the mesh if it is not already visible.

3. Select the entities in the viewport that you want to include in the surface. If you are creating a geometry surface, Abaqus/CAE allows you to select only faces and edges. Similarly, if you are creating a mesh surface, Abaqus/CAE allows you to select only element faces and element edges. You can select interior entities to include in the surface. For more information, see *Selecting interior surfaces*. To create an edge-based surface, you must select the edge of a shell or the edge of an element. For more information, see *What is a surface?*.
4. When you are finished selecting entities, click mouse button 2.
5. If you are creating a geometry surface from a native part, you must specify the side of the selected face or the end of the selected wire. For more information, see *Specifying a particular side or end of a region* and *Understanding the correspondence between geometric and physical objects*.

Additional information

- *What is a surface?*
- *Selecting objects within the viewport*
- *Defining Contact Pairs in Abaqus/Standard*
- *Defining Contact Pairs in Abaqus/Explicit*
- *Mechanical Contact Properties*

Performing Boolean operations on sets or surfaces

You can select a group of sets or surfaces from the Model Tree and use Boolean operators to combine them into a single set or surface. The following operators are available:

Union

Use the **Union** operator to create a set or surface that merges the contents of your selections.

Intersection

Use the **Intersection** operator to create a set or surface containing the items common to all your selections.

Difference

Use the **Difference** operator to create a set or surface by subtraction. You must designate the "first" set or surface, and Abaqus/CAE subtracts the remaining selections from it.

For example, you can use the **Union** operator to combine all of the top faces from a linear pattern of part instances into a single surface.

1. From the Model Tree, expand the **Sets** or **Surfaces** container associated with a part or with the assembly.
2. Select all of the desired sets or surfaces. You can use a combination of drag-select, **[Shift] + Click**, and **[Ctrl] + Click** to select the desired sets or surfaces. You cannot combine sets with surfaces. For more information, see *Drag-selecting multiple objects*.
3. Click mouse button 3 over the Model Tree, and select **Boolean** from the menu that appears.

   Abaqus/CAE displays the **Sets Boolean** or **Surfaces Boolean** dialog box.
4. In the dialog box, enter the name of the new set or surface and select the desired operator.
5. If you selected the **Difference** operator, click the arrow to the right of the **First set** or **First surface** field and specify the set or surface from which Abaqus/CAE will subtract your remaining selections.
6. Click **OK**.

   Abaqus/CAE creates the new set or surface, and it appears in the Model Tree.

Additional information

- *Using the Set and Surface toolsets*
- *Selecting objects within the viewport*

Editing sets and surfaces

You can edit an existing set or surface by reselecting entities in the viewport.

1. From the main menu bar, select **Tools->Set->Edit->*set of your choice*** or **Tools->Surface->Edit->*surface of your choice***.

   Note:

   You can also click **Edit** in the **Set Manager** or in the **Surface Manager** after selecting the name of the set or surface of interest.

   The set or surface is highlighted in the viewport, and Abaqus/CAE prompts you to reselect the objects for the set.

2. In the viewport, reselect entities to include in the set or surface. Use **[Shift] + Click** and **[Ctrl] + Click** to change which entities are selected in the viewport.

   Tip: You can use the **Selection** toolbar to limit the types of entities that you can select in the viewport. For more information, see *Selecting objects within the viewport*."

   When you are finished selecting entities, click mouse button 2.

   If you are editing a geometry set, you can select only geometry for the set. Similarly, if you are editing a node or element set, you can select only nodes or elements, respectively. If you are editing a set imported from an output database or from an input file that contains both nodes and elements or a set created using the Boolean union operator that contains nodes, elements, and geometry, Abaqus/CAE prompts you to choose the type of entity that you will be reselecting. If you reselect the nodes, Abaqus/CAE retains any original elements and geometry in the set. Repeat this step as needed to edit each type of entity.

3. If necessary, specify the side or end of the regions that you want to include in the surface and then click mouse button 2. (For more information, see *Specifying a particular side or end of a region*.)

Additional information

- *Using the Set and Surface toolsets*
- *Selecting objects within the viewport*

Associating objects (such as loads and sections) with sets and surfaces

When you select the region to which to apply an object, such as a load or a section, you can choose between the following techniques:

Highlight selections in viewport

Click on the entities in the current viewport; for example, a face or a group of vertices. Use **[Shift] + Click** and **[Ctrl] + Click** to append entities to your selection.



Tip: You can use the **Selection** toolbar to limit the types of entities that you can select. For more information, see *Selecting objects within the viewport*.

Sets or Surfaces

Use an existing set or surface that you have created with the Set or Surface toolsets. If you choose to use an existing set or surface, the **Region Selection** dialog box appears. This dialog box contains a list of the sets or surfaces that are appropriate for the object you are applying.

Click the appropriate button in the prompt area to toggle between the two techniques. For more information, see *Using the Set and Surface toolsets*. You can also use selection groups to speed up your selection procedures. You can copy selected entities into selection groups, and you can paste entities from selection groups into your selection. For more information, see *What is a selection group?*.

The Stream toolset

A streamline is a curve that is instantaneously tangent to the velocity vector of the flow, and a stream is a set of streamlines that enable you to visualize the velocity or vorticity data in a fluid flow analysis. The Stream toolset is available only in the Visualization module.

This chapter explains how to use the Stream toolset to create, modify, and delete streams; display or hide them in the viewport; and customize several aspects of their appearance.

In this section:

- *Understanding stream display*
- *Creating a stream*
- *Displaying and hiding streams*
- *Customizing stream display*

Understanding stream display

A streamline traces the path tangent to a nodal vector field, and a stream is a set of streamlines that enable you to visualize data emanating from a particular location in a vector field. *Figure 1* shows a sample stream of 12 streamlines displaying velocity data in a fluid flow analysis.

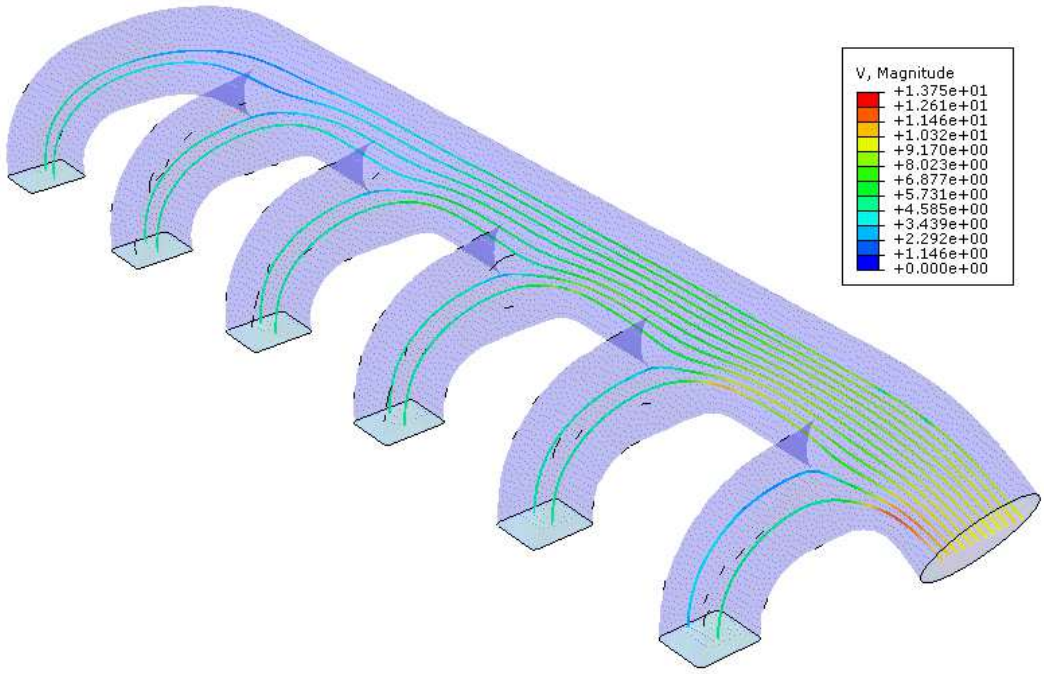


Figure 1: 12-pointed stream showing velocity data for a flow analysis through a manifold.

You can create streams by first defining a rake, which is a line segment with a series of points specified along its length that control the locations where Abaqus/CAE displays streamlines for a fluid flow analysis. Rakes are placed into an area of interest in the fluid flow; when you create and display the stream in the viewport, Abaqus/CAE displays a number of streamlines that corresponds to the number of points on the rake. Streamlines are evenly spaced along the rake, except for rakes defined using nontrivial path definitions; for more information about stream rake definition, see *Creating a stream*.

The streamline shape is driven by the currently selected stream variable, which is usually velocity or vorticity. To change the current stream variable, see *Selecting the stream field output variable*. The streamline color is driven by the currently selected primary variable. For more information on customizing streamline color options, see *Customizing stream colors*.

You must define separate stream rakes for each part instance for which you want to display flow data. Abaqus/CAE computes stream data for individual part instances only and does not display streamlines across separate part instances.

Creating a stream

A stream definition has two components: the location of the line segment or the path definition that you want to use as your “rake” for investigations of velocity or vorticity data, and the number of points along that rake for which you want to display streamlines of fluid flow data. You can specify the rake location by picking nodes from the viewport, by entering coordinates in the dialog box, or by selecting one of the path definitions in your session. Edge list paths cannot be used for stream definition.

You cannot save streams to the output database file. If you want to retain a stream that you have defined, record the process of defining that stream as a macro in Abaqus/CAE. For more information on macros, see *Managing macros*.

1. From the main menu bar in the Visualization module, select **Tools->Stream->Create**.

    **Tip:** You can also create a stream by using the  tool in the Visualization module toolbox.

   The **Create Stream** dialog box appears.
2. If desired, specify a custom name for the stream definition in the **Name** field.
3. If desired, increase or decrease the number of points specified in the **Number of points** field. By default, Abaqus/CAE creates a rake with five points.
4. From the **Specification** options, select **By nodes**, **By coordinates**, or **By path**.
5. To specify the location of the stream rake, do one of the following:

Select the nodes directly from the viewport

1. Click .

   Abaqus/CAE prompts you to select the first point.
2. Select the starting node for the rake.
3. Select the ending node for the rake.

Abaqus/CAE populates the **Start** and **End** fields with your selections.

Enter the coordinates into the dialog box

1. In the **Start (X, Y, Z)** field, enter the coordinates of the rake starting point.
2. In the **End (X, Y, Z)** field, enter the coordinates of the rake ending point.

Specify a path as the stream rake

From the **Path name** list, select the path that you want to use as a stream rake.

6. Click **OK**.

   Abaqus/CAE creates the stream definition and displays it in the viewport.

Additional information

- *Displaying and hiding streams*
- *Customizing stream display*

Displaying and hiding streams

You can display or hide streams individually in the current viewport by toggling their **Show** check boxes in the **Stream Manager**, and you can display or hide all of the streams in your session by clicking **Set All On** or **Set All Off**. The manager displays all streams defined in your session and indicates whether each one is currently displayed. In addition, you can display or hide the upstream or downstream component of each stream by toggling its **Upstream** or **Downstream** check boxes.

The **Stream Manager** also displays the number of points on the rake for each stream in your session. You can edit this number for a stream definition directly from the manager, except for streams defined on circular paths or streams defined using path definitions consisting of three or more points or nodes.

Customizing stream display

This section describes how you can customize the color, line style, and arrows of streamlines using settings in the **Stream Plot Options** dialog box.
You can open the **Stream Plot Options** dialog box by selecting **Options**->**Stream** from the main menu bar of the Visualization module of Abaqus/CAE.

The changes you make in the **Stream Plot Options** dialog box apply to all active streams in the current viewport only. Each viewport has separate stream options, which enables you to display streamlines differently in each viewport. For example, you can display streamlines using a uniform color in one viewport and display thicker streamlines that use banded contours based on the current primary variable in a different viewport.

You can also control the default settings that appear in the **Stream Plot Options** dialog box by editing the `onCaeStartup()` function in the Abaqus environment file (`abaqus_v6.env`). For more information, see *Using Abaqus Scripting Interface commands in your environment file*.

In this section:

- *Customizing stream colors*
- *Customizing streamline thickness and arrows*

Customizing stream colors

You can adjust the color that Abaqus/CAE uses to display streamlines for the active streams in the current viewport. The **Stream Plot Options** dialog box enables you to display streamlines with:

- a single, uniform color along their entire length,
- a continuous contour spectrum that varies along the length of the streamlines, or
- a banded contour spectrum that varies along the length of the streamlines.

Continuous contours display gradations without any solid bands of color, while banded contours display a number of contour intervals in which the color is constant. Contours for streamlines reflect values for the current primary variable and are subject to the contour settings specified in the **Contour Plot Options** dialog box. For more information about contour options, see *Customizing a contour plot*.

The color selected for uniform color display also determines the color with which Abaqus/CAE displays streamline arrows that indicate the direction of flow. For more information about toggling on arrow display, see *Customizing streamline thickness and arrows*.

1. Locate the **Color** options.

   Select **Options**->**Stream** from the main menu bar in the Visualization module. The **Color** options appear at the top of the page.
2. To specify the color option, choose **Uniform color**, **Continuous contour**, or **Banded contour**.
3. If you choose **Uniform color**, perform the following additional steps to select a color:

   1. Click the color sample ( ).

      Abaqus/CAE displays the **Select Color** dialog box.
   2. Use one of the methods in the **Select Color** dialog box to select a new color. For more information, see *Customizing colors*.
   3. Click **OK** to close the **Select Color** dialog box.

      The color sample and all streamlines in the viewport change to the selected color.
4. Click **OK** to implement your changes and to close the **Stream Plot Options** dialog box.

Customizing streamline thickness and arrows

You can adjust the display of streamlines in the current viewport by changing their thickness, adding directional arrows that indicate the direction of fluid flow, and by customizing the number of direction arrows when their display is enabled. Abaqus/CAE displays directional arrows using the uniform color selected in the **Color** options; for more information, see *Customizing stream colors*. Arrow sizes change as you increase or decrease streamline thickness.

1. Locate the **Style** options.

   Select **Options->Stream** from the main menu bar in the Visualization module. The **Style** options appear at the bottom of the dialog box.
2. Drag the **Line thickness** slider to make the streamlines thinner or thicker.
3. Toggle on **Uniform thickness across streams** to display all streamlines in the current viewport with a uniform thickness.
4. Toggle on **Show direction arrow** to display directional arrows along the streamlines.
5. Drag the **Approx. number of arrows on each streamline** slider to increase or decrease the number of directional arrows displayed on each streamline.
6. Click **OK** to implement your changes and to close the **Stream Plot Options** dialog box.

The Virtual Topology toolset

The Virtual Topology toolset allows you to ignore details, such as very small faces and edges, when you mesh a part or a part instance.

In this section:

- *What is virtual topology?*
- *What can I do with the Virtual Topology toolset?*
- *What can I do with a part or a part instance containing virtual topology?*
- *Why repair a part if I can use virtual topology?*
- *Creating virtual topology based on geometric parameters*
- *Using the Virtual Topology toolset*

What is virtual topology?

In some cases parts or part instances contain details such as very small faces and edges. These features can be important for the detailed machining and packaging design of the component; however, they may be redundant in the numerical analysis if they have little impact on the mechanics of the problem being studied. Indeed, these small details may unduly constrain the generation of the mesh, resulting in a poor mesh or a finer mesh density. In some circumstances small faces and edges can be significant if they occur in a region of interest where you need a fine mesh. However, in most cases the details add nothing to the analysis, and you want Abaqus/CAE to ignore them during the mesh generation process and during the analysis.

The topology of the model is its composition from faces, edges, and vertices. The Virtual Topology toolset allows you to manipulate this topology and create a simplified form for the purpose of meshing. This simplified form is different from the real model and is called “virtual topology.” The faces and edges that result from the manipulation are said to be “virtual.” Similarly, parts and part instances that contain virtual faces and edges are said to be “virtual.”

The Virtual Topology toolset allows you to remove small details by combining a small face with an adjacent face or by combining a small edge with an adjacent edge. The faces or edges to be combined can be specified directly, or you can choose the edges and vertices to ignore.

For example, *Figure 1* shows a piston that has some small details where the blended surfaces intersect. These details result in a dense mesh that includes some sliver-shaped elements. You can use virtual topology to ignore the small details, resulting in the more uniform mesh of better quality that is illustrated in *Figure 2*.

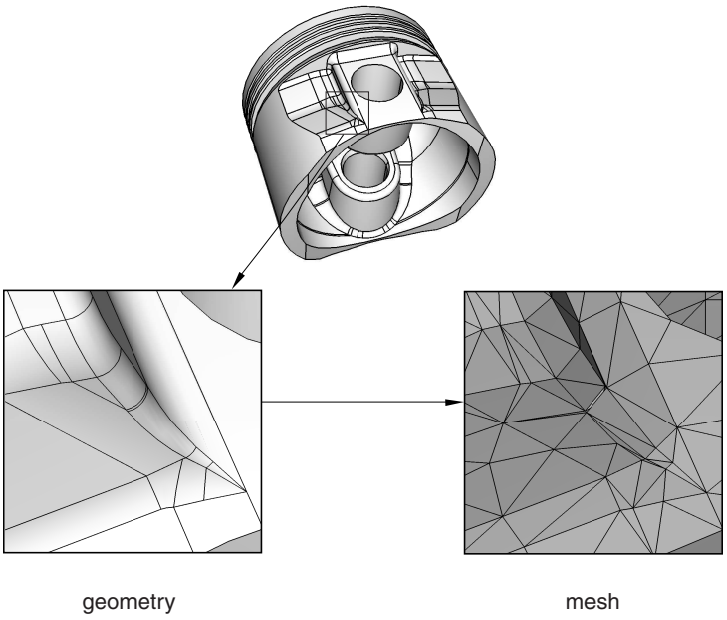


Figure 1: Small details result in a dense mesh of sliver-like elements.

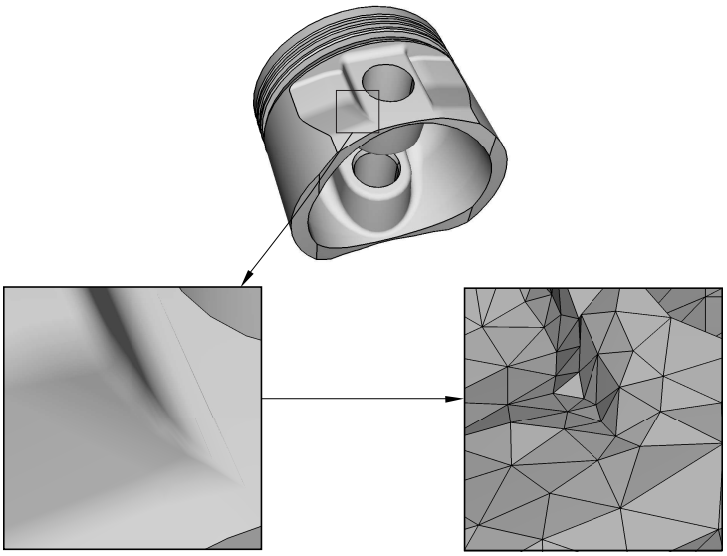


Figure 2: Using virtual topology to ignore small details results in a more uniform mesh of better quality.

Abaqus/CAE treats virtual topology in the same way that it treats standard geometry. For example, you can do the following:

- Partition virtual topology.
- Use the Geometry Edit toolset on virtual topology.
- Use Part module tools on virtual topology, such as extrude, sweep, and blend.

For example, *Figure 3* shows the piston containing virtual topology from *Figure 2* after it has been cut using the extruded cut tool in the Part module.

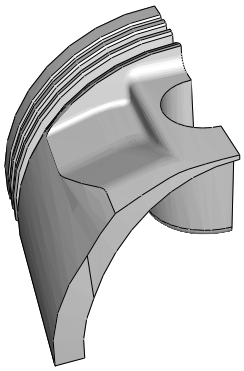


Figure 3: A part containing virtual topology is cut.



Note:

When you export a part containing virtual topology to an ACIS file, Abaqus/CAE removes the virtual topology from the part that is exported.

What can I do with the Virtual Topology toolset?

The Virtual Topology toolset is available only in the Mesh module. You can apply virtual topology to parts, or you can apply it to independent instances in the assembly. The Virtual Topology toolset allows you to do the following:

Combine faces or edges

You can select two or more faces or edges that Abaqus/CAE will combine into a single virtual face or virtual edge. If you combine two faces, any edges between the faces are ignored when Abaqus generates the mesh. Similarly, if you combine two edges, any vertices between the edges are ignored when the mesh is generated.

Ignore an edge or a vertex

You can select edges or vertices that Abaqus/CAE will ignore. Ignoring an edge between two faces is the equivalent of combining the faces. Similarly, ignoring a vertex between two edges is the equivalent of combining the edges; however, you will find the combine tools useful when you have a large number of faces and edges to combine. Using ignore to create virtual topology has the same effect as using combine to create virtual topology—ignored edges or vertices are not considered when Abaqus generates the mesh.

Automatically combine or ignore entities

You can specify a set of geometric parameters that Abaqus/CAE will use to create virtual topology. You can also select the area—faces, parts, part instances, or an entire assembly—to which Abaqus/CAE will apply the selected parameters. The virtual topology parameters are located in the **Create Virtual Topology** dialog box, which also includes tools that you can use to measure and highlight entities and preview the virtual topology. When you are finished, Abaqus/CAE creates a single virtual topology feature that contains all of the changes.

Restore entities

Rather than deleting or suppressing virtual topology features that may contain entities that you want to ignore along with some that you need to use, you can use the **Restore Entities** tool to highlight all entities that have been ignored and select the ones that you want to reactivate.

To preserve mesh quality, edges and faces to be combined should have shallow included angles that are close to 180°. The recommended angle is between 120° and 240°. You can use the angle method to choose only adjacent faces with shallow included angles. For more information, see *Using the angle and feature edge method to select multiple objects*. If the included angle at some edges or vertices lies outside this range, Abaqus/CAE displays a warning and allows you to remove those edges or vertices from your selection. If you choose to retain the sharp edges, the resulting mesh may not lie on a smooth surface. *Figure 1* illustrates a mesh on a virtual surface that is not smooth.

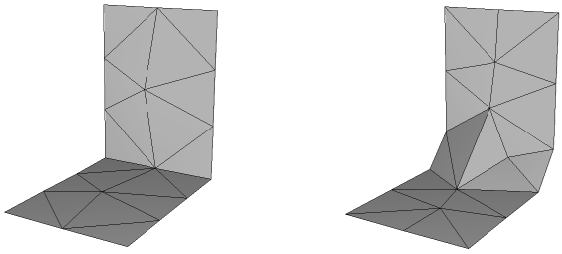


Figure 1: When you apply virtual topology to sharp edges and vertices, the resulting mesh can deviate substantially from the original surface.

You can ignore an edge only if that edge is shared by two adjacent faces or if it is embedded in a face. For example, you cannot ignore the free edge of a shell or an edge shared by three faces, as shown in *Figure 2*.

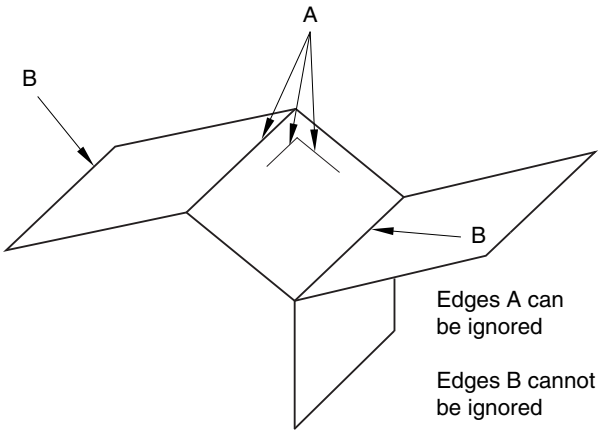


Figure 2: Some edges cannot be ignored.

Similarly, you can ignore a vertex only if that vertex is shared by two adjacent edges or embedded in a face. For example, you cannot ignore the vertex at the free end of a wire or where a wire intersects with a face, as shown in *Figure 3*.

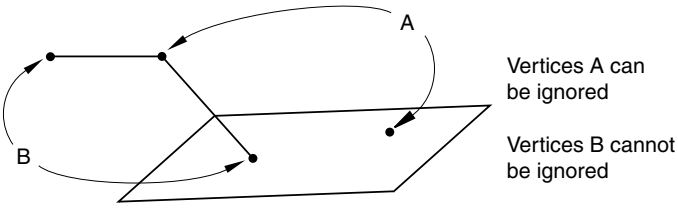


Figure 3: Some vertices cannot be ignored.

Abaqus/CAE does not support all of the meshing techniques on regions that contain virtual topology (combined faces or combined edges). Specifically, Abaqus/CAE does not support the following:

- Two-dimensional free meshing with quadrilateral or quadrilateral-dominated elements using the medial axis algorithm.
- Three-dimensional swept meshing using the medial axis algorithm.
- Two-dimensional structured meshing if the region to be meshed is not bounded by four corners.
- Three-dimensional structured meshing if the region to be meshed is not bounded by six sides.

If you need to apply virtual topology to a dependent instance, you can create a copy of the original part and then create an independent instance of the copy. You can then replace the dependent instance with the new independent instance and apply virtual topology to the replacement. For more information, see *What is the difference between a dependent and an independent part instance?*.

You may be able to make a part swept meshable by combining multiple faces on the target side into a single face; for more information, see *Swept meshing of three-dimensional solids*. However, if you combine many faces into a single face, the meshing procedure may be slower and the resulting mesh may not be acceptable. If you try to combine a large number of faces into a single face, Abaqus/CAE displays a warning and allows you to change your selection. Abaqus/CAE calculates the number of faces being combined from the cumulative number of faces in your selection. For example, if you try to combine two faces each of which already contains five combined faces, the cumulative number of faces will be ten.

What can I do with a part or a part instance containing virtual topology?

Abaqus/CAE stores virtual topology as features of the corresponding part or assembly. As a result, you can use the Model Tree to rename, suppress, resume, and delete virtual topology features. You cannot edit a virtual topology features. You can restore entities that have been ignored or combined using virtual topology. Restoring entities creates more new virtual topology features in the Model Tree—the original virtual topology features are unchanged.

You can create sets and surfaces that contain virtual faces and edges. In addition, if an existing set or surface contains an edge or vertex that you ignored, the edge or vertex is removed from the set or surface. Similarly, if an existing set or surface contains faces or edges that you combined, Abaqus/CAE replaces the original faces and edges with the new combined faces and edges. This is true only if all of the faces and edges that you combined are members of the same set.

Virtual topology is not exported; for example, if you export an assembly containing virtual topology to an ACIS file, Abaqus/CAE removes the virtual topology from the exported model. When you are creating the assembly in the Assembly module, you cannot merge or cut part instances that contain virtual topology.

Why repair a part if I can use virtual topology?

Introducing virtual topology is a convenient method for creating a clean, well-formed mesh. The emphasis is to ignore small geometric details that result in distorted elements in areas where the analysis results are not of significant interest. Virtual topology relaxes the need for the mesh to conform to every edge or vertex and allows you to obtain a simpler model with less detail. However, virtual topology does not change the underlying geometry of the part instance.

When you combine two faces to form a virtual face, Abaqus/CAE updates the display to reflect the ignored edge. However, to form a basis for the new virtual face, Abaqus/CAE maintains the underlying geometry of the faces that were combined. The nodes in the mesh will be placed on the underlying geometry. Therefore, a coarse mesh will interpolate across the virtual face, and a fine mesh will tend to converge to the geometry of the underlying combined faces. Virtual topology allows the mesh definition to be controlled by meshing parameters, such as seed size, and it helps to free meshing from constraints imposed by geometry, such as small faces.

In contrast, you use the Geometry Edit toolset to correct minor flaws that are typically introduced when a part is imported from a third-party application. You use the Geometry Edit toolset to correct the underlying geometry; for example, to stitch small gaps and to delete self-intersecting faces and recreate a valid replacement face. You also use the Geometry Edit toolset to remove redundant topology, such as superfluous vertices and edges. The Geometry Edit toolset modifies the actual geometry, and geometric restrictions limit how you can apply the repair tools; for example, you can remove redundant vertices only if the vertex lies on a curve that is common to both edges.

When you export a virtual part instance, the virtual topology is lost because the virtual face, edge, or vertex is not true geometry. For example, a virtual face is an abstracted reference to the underlying collection of combined faces. Therefore, you should use the Geometry Edit toolset if you need to remove small details from a part and want to export the part with the changes retained.

Creating virtual topology based on geometric parameters

You can use the controls in the **Create Virtual Topology** dialog box to automatically ignore unnecessary model details based on a set of geometric parameters. You can control the following parameters to determine which features Abaqus/CAE considers merging with surrounding geometry:

- Edge length
- Face size
- Face aspect ratio
- Face corner angle
- Thickness of faces representing small stair features
- Corner angle for combining adjacent edges or faces
- Angle and radius controls for combining blends (chamfered or rounded corners) with surrounding faces
- Redundant entities

Abaqus/CAE attempts to simplify the model by merging neighboring edges and faces when the selected parameters are exceeded.

The redundant entities parameter can only be toggled on or off—there are no other settings for this parameter. Redundant entities do not add to the geometry of the model. Redundant entities include:

- Edges that are interior to an existing face
- Edges that separate an otherwise continuous planar or curved face
- Vertices that are interior to a face
- Vertices that separate two straight edges or smooth curves without connecting a new edge

For example, *Figure 1* contains four redundant edges and one redundant vertex. Virtual topology will merge the edges with the surrounding faces and merge the vertex with the surrounding edges, leaving only the underlying box.

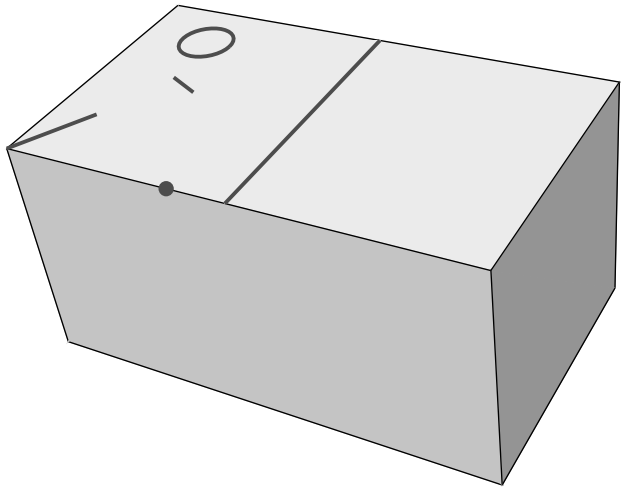


Figure 1: Redundant edges and vertices.

Keep in mind that redundant entities may define areas used by other tools within Abaqus/CAE. For example, the circular edge in *Figure 1* may define a face for the application of a pressure load. The controls in the **Create Virtual Topology** dialog box are based entirely on geometric parameters; you should carefully review the changes made using

virtual topology to make certain that none of the changed entities are required for other parts of the Abaqus/CAE model definition.

By default, all the virtual topology parameters in the dialog box are active, as shown in *Figure 2*. The default values for the edge length, face size, stair feature, and blend radius are calculated based on the size of the selected part or part instance; the other criteria are dimensionless and suitable default values are provided. You can toggle off the controls for entities that you want to be ignored by virtual topology. For example, if you have small faces that are a critical component of the model, toggle off **Faces smaller than**.

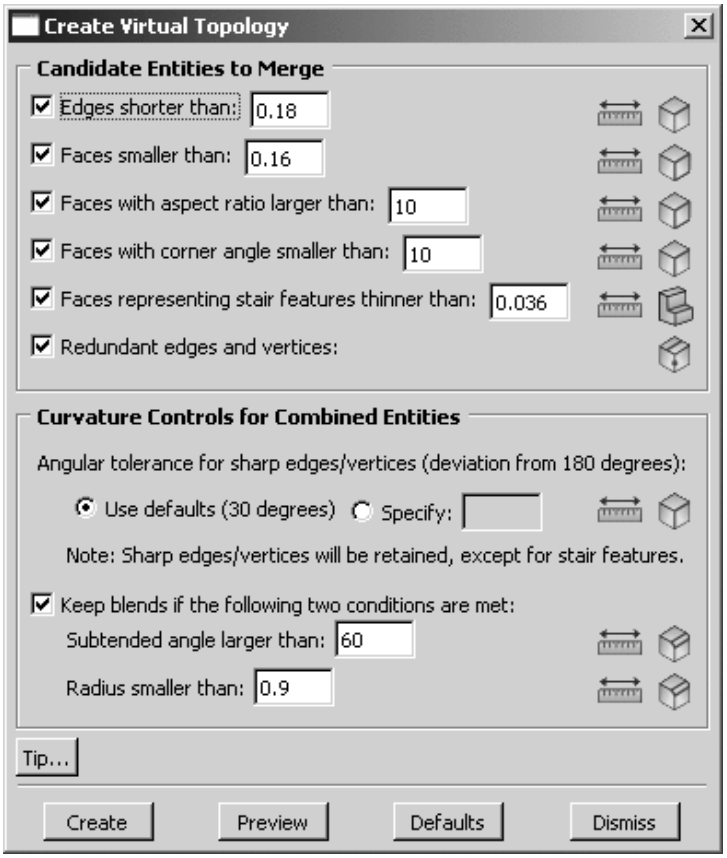


Figure 2: The Create Virtual Topology dialog box.

To help you review potential changes, each parameter in the dialog box contains tools that you can use to measure and highlight entities in the viewport that satisfy the corresponding geometric criteria. Items that are highlighted may not be changed if they conflict with other criteria or model requirements. To use the tools, click on the icons located to the right of each parameter in the **Create Virtual Topology** dialog box. The measure entities tools allow you to measure entities in the viewport. Abaqus/CAE displays the measured values in the message area; you can use the measurements to revise the parameter limits for your model. The highlight tools, such as **Highlight Short Edges**, indicate entities that satisfy the current parameter settings. For the **Candidate Entities to Merge** area of the **Create Virtual Topology** dialog box, the highlight tools indicate entities that may be merged with the surrounding geometry.

The highlight tools do not account for the combination of parameters or any other factors in the model that may prevent the highlighted features from being merged with surrounding geometry. For example, small stair features must satisfy two parameters for replacement using automatic virtual topology:

- The stair height or thickness must be less than the specified value.

- The opposing angles that separate the stair face from the surrounding faces must deviate from 180° by an angle larger than the specified angular tolerance for sharp edges/vertices (explained below with the curvature controls).

Figure 3 shows the angles and the face dimension that make up a stair feature. If you use the highlight tool to view potential faces, Abaqus/CAE does not consider the angle measurements when selecting the faces to highlight.

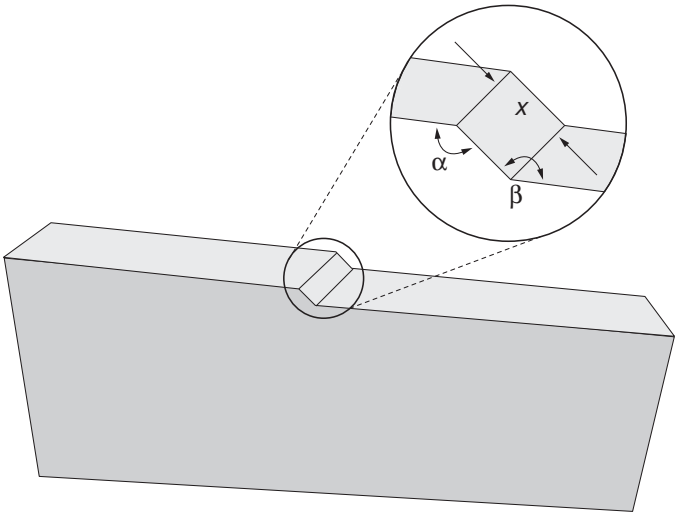


Figure 3: The face thickness (stair height), x, and the angles α and β create a stair feature.

The **Curvature Controls for Combined Entities** area of the **Create Virtual Topology** dialog box is used for combining corner angle and blend entities. The curvature control parameters and highlight tools indicate edges, vertices, or blends that will be retained—not replaced by virtual topology—using the current settings. The following examples explain how the curvature control parameters for corner angles and blends are interpreted by Abaqus/CAE.

Angular tolerance for sharp edges/vertices (deviation from 180 degrees)

Corner angles are the angles between any two adjacent faces or edges that create an edge or vertex, respectively, in the model. If a corner angle is close enough to 180° to be considered flat, Abaqus/CAE considers using virtual topology to ignore the associated edge or vertex. The default parameter setting considers angles flat if they deviate from 180° by less than 30°. *Figure 4* shows how merging faces effects the resulting mesh. The two upper images use a shallow angle so the resulting mesh is a close approximation of the geometry. Increasing the angle between the merged faces as shown in the lower images yields a progressively worse approximation. The corner angle parameter also applies to face corner angles between the edges of two-dimensional parts and to the angles that create small stair features.

Tip: To keep all corner angles, specify a deviation of 0°.

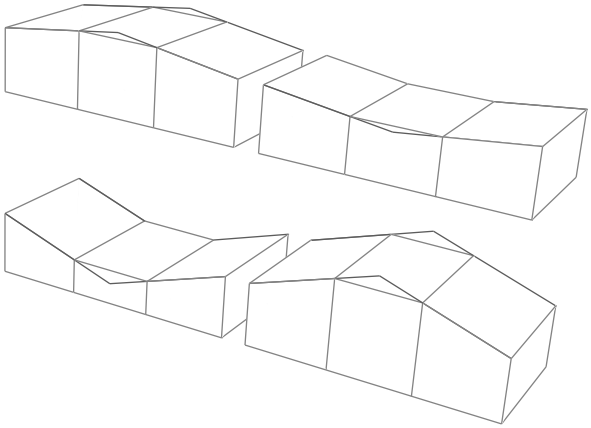


Figure 4: The correspondence between the mesh and the geometry decreases as the corner angles replaced by virtual topology (the peak and valley) deviate further from 180°.

Keep blends with a large subtended angle and a small radius

Blends—rounded or filleted corners—must satisfy two parameters to avoid consideration for virtual topology. This is due to the fact that either the subtended angle or the blend radius alone can create a feature that can be approximated well by the mesh, but combining a large angle with a small radius leads to a poor approximation. *Figure 5* illustrates how two different blend angles are approximated by the mesh if they are merged with the surrounding geometry. The blend radius is the same for both parts, but the larger subtended angle in the upper image results in a bumpy mesh at the corner.

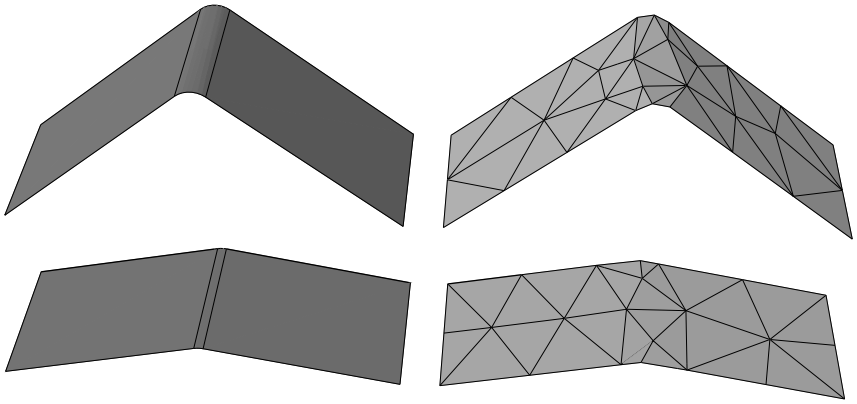


Figure 5: A large blend angle (upper image) may yield a bumpy mesh when it is replaced by virtual topology.

In *Figure 6* the subtended angle is the same for both parts, but the small blend radius in the upper image results in a poor quality mesh.

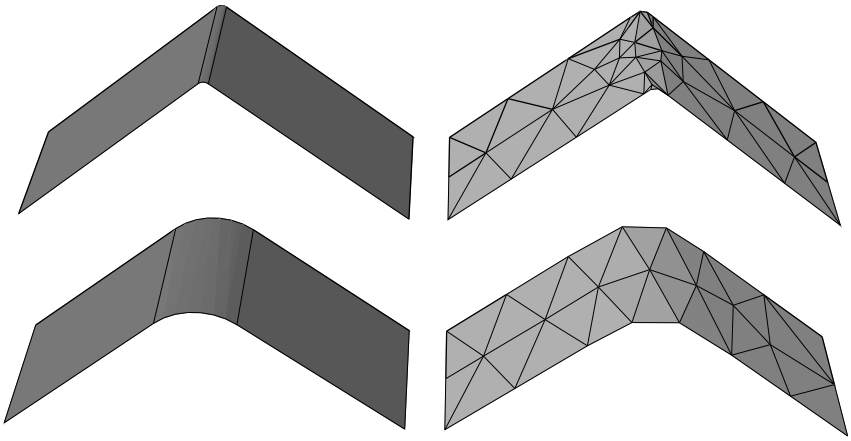


Figure 6: A small blend radius (upper image) may yield a poor quality mesh when it is replaced by virtual topology.

For detailed instructions on how to use the **Create Virtual Topology** dialog box, see *Creating virtual topology automatically*.

Using the Virtual Topology toolset

This section explains how to use the Virtual Topology toolset to remove small details from a part or a part instance prior to meshing and how to restore small details without deleting the virtual topology.

In this section:

- *Creating virtual topology automatically*
- *Combining faces*
- *Combining edges*
- *Ignoring edges and vertices*
- *Restoring entities that were replaced by virtual topology*

Creating virtual topology automatically

You can select **Tools->Virtual Topology->Automatic Create** from the main menu bar to open the **Create Virtual Topology** dialog box. Use the controls in the dialog box to combine faces, combine edges, or ignore entities in the model.

1. From the **Object** field in the context bar, select a part or the assembly.
2. From the main menu bar, select **Tools->Virtual Topology->Automatic Create**.

   Tip: You can also automatically create virtual topology using the  tool, located in the Mesh module toolbox.

3. Select the target regions to be considered for virtual topology.

   The default selection method is parts or part instances, depending on whether you selected a part or the assembly in Step 1. If desired, you can change the selection method in the prompt area to select faces.

   You can use a combination of drag-select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one part instance in an assembly or more than one face in a part or an assembly. You can also use the angle method or the face curvature method to select more than one face. For more information, see *Selecting objects within the current viewport*.

   Abaqus/CAE opens the **Create Virtual Topology** dialog box.

4. Toggle off any criteria that you do not want Abaqus/CAE to consider for virtual topology.
5. If desired, click the  tools, located next to each control that includes a size parameter, to measure entities in the viewport.

   Follow the instructions in the prompt area to make your selection. You can measure the length of multiple edges or the area of multiple faces in a single operation. When you are finished, click **Done** in the prompt area (if necessary).

   Abaqus/CAE displays the measured value in the message area. If you selected multiple entities, Abaqus/CAE displays the average, minimum, and maximum values. Use the measured values to verify or adjust the virtual topology controls in the dialog box.

6. Click on the highlight tools, located on the right side of the dialog box, to view entities that may be replaced by virtual topology according to the corresponding parameter. For example, the  tool highlights edges within your target region that are smaller than the current short edge length setting. Highlighted entities will not be replaced if they are an integral part of the model or if merging them violates the curvature control parameters.

   Note:

   The highlight tools for the curvature controls (corners and blends) indicate entities that will be *retained* according to the current settings.

7. When you are satisfied with the control settings, click **Preview** to highlight all the entities that will be replaced by virtual topology.



Note:

For large parts or assemblies, or any selection set that contains numerous items for replacement, the preview function may take some time. Instead, you can use the highlight tools as mentioned in Step 6 to quickly view entities that meet the parameter settings.

8. Click **Create** to create the virtual topology according to the selected parameter settings.

   Abaqus/CAE highlights any edges or vertices in the viewport that will become redundant using your current selections and prompts you to choose whether they should also be combined. Abaqus/CAE creates an **Auto virtual topology** feature in the Model Tree to combine or ignore the selected entities.

9. If the target regions selected in Step 3 are still valid, you can repeat Steps 4 through 8 to create more virtual topology on the same regions or click **Dismiss** to close the **Create Virtual Topology** dialog box.

   If the selected target regions are invalidated by the virtual topology, Abaqus/CAE closes the **Create Virtual Topology** dialog box and returns you to Step 3 to select new target regions.

You can use the Model Tree to restore the selected entities by deleting or suppressing the **Auto virtual topology** objects; you can use the **Restore Entities** tool to restore some entities while retaining the rest of the virtual topology feature. For more information on restoring individual entities, see *Restoring entities that were replaced by virtual topology*.

Combining faces

You can select **Tools->Virtual Topology->Combine Faces** from the main menu bar to combine selected faces. During mesh generation, Abaqus/CAE treats the combined faces as a single face and ignores any edges and vertices that were removed.

1. From the **Object** field in the context bar, select a part or the assembly.
2. From the main menu bar, select **Tools->Virtual Topology->Combine Faces**.

   Tip: You can also combine faces using the  tool, located in the Mesh module toolbox.

3. Select the faces that you want to combine.

   You can use a combination of drag-select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one face to combine. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired face, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. If the vertices at the end of any edges of the selected faces can also be ignored, Abaqus/CAE highlights the vertices and asks if you want to include them in your selection. Click **Yes** to ignore the highlighted vertices. Click **No** to keep the highlighted vertices.

   Abaqus/CAE combines the selected faces and vertices. You can use the Model Tree to restore the selected faces and vertices by deleting or suppressing the virtual topology feature; you can use the **Restore Entities** tool to choose faces and vertices to restore while retaining the rest of the virtual topology feature. For more information on restoring individual entities, see *Restoring entities that were replaced by virtual topology*.

Combining edges

You can select **Tools->Virtual Topology->Combine Edges** from the main menu bar to combine selected edges. During mesh generation, Abaqus/CAE treats the combined edges as a single edge and ignores any vertices that were removed.

1. From the **Object** field in the context bar, select a part or the assembly.
2. From the main menu bar, select **Tools->Virtual Topology->Combine Edges**.

   Tip: You can also combine edges using the  tool, located in the Mesh module toolbox.

3. Select the edges that you want to combine.

   You can use a combination of drag-select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one edge to combine. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edge, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

   Abaqus/CAE combines the selected edges. You can use the Model Tree or the **Restore Entities** tool to restore the selected edges. For more information on restoring individual entities, see *Restoring entities that were replaced by virtual topology*.

Ignoring edges and vertices

You can select **Tools->Virtual Topology->Ignore Entities** from the main menu bar to ignore selected edges and vertices. Abaqus/CAE removes the selected entities from the assembly and ignores the entities when the mesh is generated.

1. From the **Object** field in the context bar, select a part or the assembly.
2. From the main menu bar, select **Tools->Virtual Topology->Ignore Entities**.

   Tip: You can also ignore edges and vertices using the  tool, located in the Mesh module toolbox.

3. Select the combination of edges and vertices that you want to ignore.

   You can use a combination of drag-select, **[Shift] + Click**, **[Ctrl] + Click**, and the angle method to select more than one edge or vertex to remove. For more information, see *Selecting objects within the current viewport*.

   Tip: If you are unable to select the desired edge or vertex, you can use the **Selection** toolbar to change the selection behavior. For more information, see *Using the selection options*.

4. If the vertices at the end of a selected edge can also be ignored, Abaqus/CAE highlights the vertices and asks if you want to include them in your selection. Click **Yes** to ignore the highlighted vertices. Click **No** to keep the highlighted vertices.

   Abaqus/CAE ignores the selected edges and vertices. You can use the Model Tree or the **Restore Entities** tool to restore the selected entities. For more information on restoring individual entities, see *Restoring entities that were replaced by virtual topology*.

Restoring entities that were replaced by virtual topology

You can select **Tools->Virtual Topology->Restore Entities** from the main menu bar to restore edges and vertices that were previously replaced by virtual topology. Using the restore tool allows you to select individual entities to restore rather than deleting or suppressing the virtual topology feature, which would bring back all of the entities.

1. From the **Object** field in the context bar, select a part or the assembly.
2. From the main menu bar, select **Tools->Virtual Topology->Restore Entities**.

   Tip: You can also restore entities using the tool, located in the Mesh module toolbox.

   Abaqus/CAE highlights in blue all entities that have been replaced by virtual topology.
3. Select the combination of entities that you want to restore.

   You can use a combination of drag-select, **[Shift] + Click**, and **[Ctrl] + Click** to select more than one entity to restore. For more information, see *Selecting objects within the current viewport*.
4. Click mouse button 2, or click **Done** in the prompt area.

   Abaqus/CAE creates a **Restore ignored geometry** feature in the Model Tree to restore the selected entities—the virtual topology features are not changed. If there are more entities that can be restored, Abaqus/CAE continues the selection procedure.
5. When you have finished restoring entities, click mouse button 2 or the cancel button in the prompt area.

Customizing model display

This part explains how you can customize your model display.

In this section:

- *Customizing geometry and mesh display*
- *Color coding geometry and mesh elements*
- *Using display groups to display subsets of your model*
- *Overlaying multiple plots*
- *Cutting through a model*

Customizing geometry and mesh display

This chapter explains how you can customize your geometry and mesh display.
The Visualization module has its own set of options to customize model appearance; for more information, see *Customizing plot display* and *Customizing viewport annotations*.

In this section:

- *Using geometry and mesh display options*
- *Choosing a render style*
- *Controlling edge visibility*
- *Controlling curve refinement*
- *Defining mesh feature edges*
- *Controlling translucency for substructure parts*
- *Controlling beam profile display*
- *Controlling shell thickness display*
- *Controlling datum display*
- *Controlling the display of individual coordinate systems*
- *Controlling reference point display*
- *Customizing mesh display*
- *Controlling model lighting*
- *Controlling instance visibility*
- *Controlling the display of attributes*
- *Saving your display options settings*

Using geometry and mesh display options

You can use the **Part Display Options**, **Assembly Display Options**, and **ODB Display Options** menus to customize geometry and mesh display.

Depending on the module you are using, Abaqus/CAE presents you with one of the following menu selections to customize geometry and mesh display:

View->Part Display Options

In geometry-related modules you can use this menu item to control your geometry render style, edge visibility, curve refinement, element and node labels, and the visibility of the various types of datum geometry.

View->Assembly Display Options

In assembly-related modules you can use this menu item to control your assembly render style, edge visibility, and the visibility of the various types of datum geometry. You can also use it to control the display of the mesh, element and node labels, part instances, loads, boundary conditions, and predefined fields in those same assembly-related modules.

View->ODB Display Options

This menu item is applicable only to the Visualization module. For more information, see *Customizing plot display*.

You can save the current display options and apply them to future Abaqus/CAE sessions by selecting **File->Save Options** from the main menu bar. For more information, see *Saving your display options settings*.

Choosing a render style

Render style is the style in which Abaqus/CAE displays your model.

You can use the **View->Part Display Options** and **View->Assembly Display Options** menu items to display your model in one of three render styles: wireframe, hidden, or shaded; these styles are shown in *Figure 1*. An explanation of these choices follows.

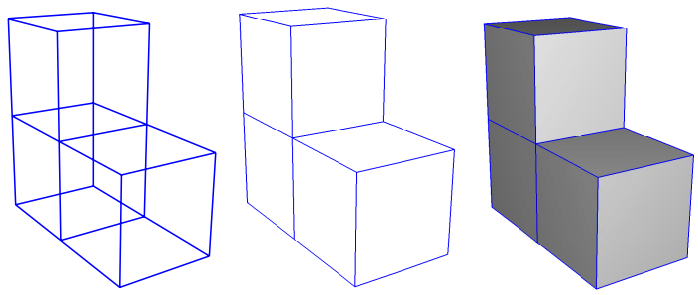


Figure 1: Model showing render style options. From left to right: the wireframe, hidden, and lightsource-shaded render styles.

Wireframe

Displays model edges; both interior and exterior edges are potentially visible. Wireframe plots produce a frame-like visual effect in which model faces are not displayed. Wireframe is the most rapidly drawn render style.

Hidden

Displays a wireframe plot in which edges obscured by the model are either not shown or are shown as dotted lines, depending on which option you select. (For more information on this option, see *Controlling edge visibility*.) Hidden plots produce a solid rather than frame-like appearance.

Shaded

Displays a filled plot in which a light source appears to be directed at the model. Shaded plots produce a highly three-dimensional visual effect. Edges attached to faces in shaded plots are always drawn in black.

1. Locate the **Render Style** options.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.

2. From the top of the dialog box, click **Wireframe**, **Hidden**, or **Shaded** to select the style that you want.

   Tip: You can also select the render style using the wireframe , hidden , and shaded icons located in the Render Style toolbar.

3. + Click **OK** to implement your changes and to close the dialog box.

Abaqus/CAE renders the display in the selected style, and your changes are saved for the duration of the session.

Additional information

- *Customizing geometry and mesh display*
- *Choosing a render style*

Controlling edge visibility

You can control the visibility of geometry edges, reference representations, highlighted faces, and mesh edges.

Using the **View->Part Display Options** or **View->Assembly Display Options** menu items, you can control the visibility of the following:

Geometry edges

If a part or part instance is displayed with the hidden render style, Abaqus/CAE suppresses obscured geometry edges by default. Alternatively, if you toggle on the **Show dotted lines in hidden render style** option, Abaqus/CAE displays the obscured edges using a dotted line style.

If a part or part instance is displayed with the shaded render style, Abaqus/CAE displays the edges by default. Non-wire edges (edges attached to faces) are displayed in black. Alternatively, if you toggle off the **Show edges in shaded render style** option, Abaqus/CAE suppresses edge display.

If a three-dimensional part or part instance contains faces with curved edges, by default Abaqus/CAE displays gray "silhouette" edges originating from the faces, as shown in the hidden-line plot in *Figure 1*.

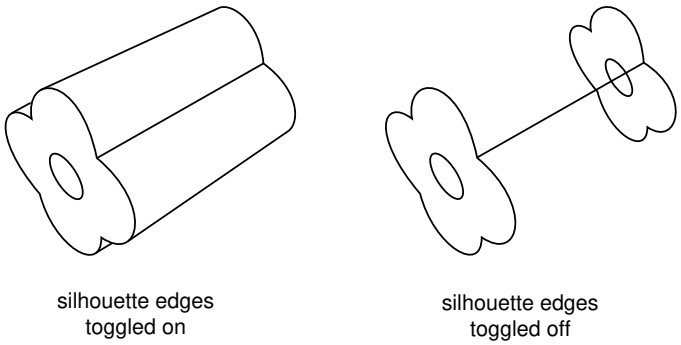


Figure 1: Hidden-line plot showing silhouette edges.

Unlike true edges, silhouette edges serve only as a visual aid; for example, you cannot select or partition a silhouette edge. Alternatively, if you toggle off the **Show silhouette edges** option, Abaqus/CAE displays only true edges.

Abaqus/CAE displays a curved part using a faceted representation of the part, and you use the **Curve refinement** option to specify the degree of faceting. For more information, see *Controlling curve refinement*.

Reference representation

If you are creating a midsurface model and have assigned midsurface regions to cells in the solid model, the geometry of the selected cells is contained in the reference representation. By default, Abaqus/CAE displays the reference representation in the Part module. However, you can use the **Show reference representation** option to toggle display of the reference representation in all modules where the part or assembly is displayed. Toggle off **Apply translucency** to display the reference representation as opaque instead of the default translucent appearance. For more information on the reference representation, see *Understanding the reference representation*.

Highlighted faces

You can control the style with which Abaqus/CAE displays the highlighted geometry faces of parts and assemblies. *Figure 2* shows three views of a sample part with its front face selected: the left figure uses stippling as the selection method, the middle figure shows an example of isolines, and the right figure displays facet selection.

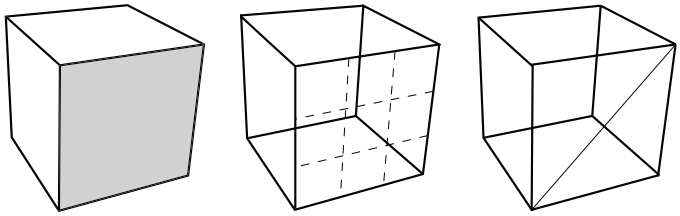


Figure 2: Highlighting faces with Stippling, Isolines, and Facets.

The stippling method offers a performance advantage, particularly for large, complex parts and assemblies. Using isolines can allow you to see a part or assembly more easily in wireframe mode than the stippling method. Finally, displaying all of a part or assembly's facets can help you debug a mesh more effectively, because meshing depends on the orientation of facets in a part or assembly.

Mesh edges

For mesh edges within a meshed part or a part imported from an output database, the visibility options are:

All edges

Displays all element edges. To see element edges on the interior of the model, you must also set the *render style* to wireframe.

Exterior edges

Displays only edges on the exterior of the model.

Feature edges

Displays only edges on the exterior of the model that are calculated to be feature edges. Feature edges lie between elements that have normals that differ by more than the “feature angle.” For more information on controlling the feature angle, see *Defining mesh feature edges*.

Free edges

Displays only edges that belong to a single element. Free edge display is particularly useful for locating potential holes or cracks in your mesh.

These options are shown in *Figure 3*.

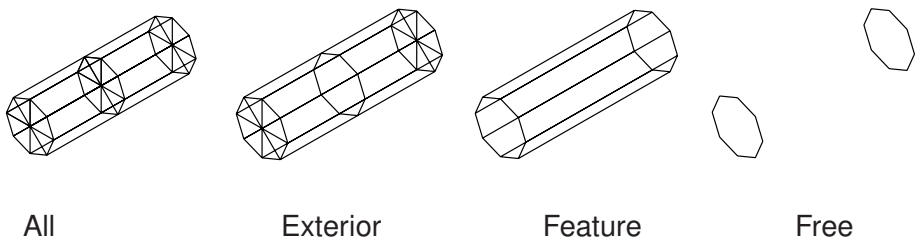


Figure 3: Model showing mesh edge display options.

If a mesh is displayed with the shaded render style, Abaqus/CAE displays the edges by default. Alternatively, if you toggle off the **Show edges in shaded render style** option, Abaqus/CAE suppresses edge display.

With the exception of showing hidden geometry edges as dotted lines, you cannot control the line style, color, or thickness of edges.

1. Locate the edge visibility options.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.

2. Select the desired **Geometry** edge settings.
3. Select the desired **Mesh Edges** settings.
4. + Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Defining mesh feature edges*
- *Choosing a render style*
- *Customizing geometry and mesh display*

Controlling curve refinement

Abaqus/CAE uses a faceted representation of a curved face or a curved edge when displaying a part or part instance. When you are working in the Part module, you can use the **Curve refinement** option from the **Part Display Options** dialog box to specify the degree of this faceting applied to the current part. You can select one of five faceting levels between extra coarse and extra fine. Set the refinement to **Extra Coarse** to speed up display of a large model. Set the refinement to **Extra Fine** if you want to create a very accurate display for printing. Abaqus/CAE applies the curve refinement setting only to the part in the current viewport.

In addition, Abaqus/CAE uses the faceted representation of a part instance in the Assembly module when determining contact between part instances and when determining the position of attachment lines. You use the **Curve refinement** option to control the accuracy of the contact detection tool and to help display the part geometry more accurately.

1. Locate the **Curve refinement** options.

   From the main menu bar, select **View->Part Display Options**. In the dialog box that appears, click the **General** tab.
2. Select the desired curve refinement setting.
3. + Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE applies the curve refinement setting only to the part in the current viewport.

Additional information

• *Customizing geometry and mesh display*

Defining mesh feature edges

You can specify that only the feature edges of a meshed part are visible.

You use feature edges to mask the detail provided by a mesh; feature edges are typically the physical edges of the part being meshed and do not include all the additional element edges. *Figure 1* shows a meshed part displayed at three different feature angles—0°, 5°, and 20°.

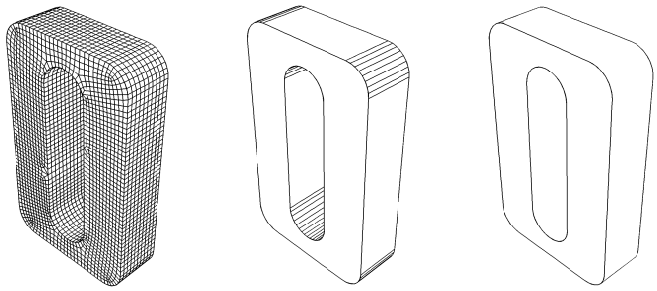


Figure 1: Plots showing feature angles of 0°, 5°, and 20°.

Feature edges are defined as adjacent edges with normals that differ by more than the "feature angle." You can customize the feature angle when you select **Feature** mesh edge visibility. Larger angles will reduce the number of feature edges; conversely, smaller angles will cause more edges to be visible. The default mesh feature angle is 20°.

1. Locate the feature angle option.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.
2. From the bottom of the dialog box, select **Feature edges** from the list of mesh edges to show.

   Abaqus/CAE displays an **Angle** data field to the right of **Feature**.
3. Click the **Angle** data entry field, and enter the desired feature angle.
4. + Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

- *Controlling edge visibility*
- *Customizing geometry and mesh display*

Controlling translucency for substructure parts

You can specify that the substructure parts and part instances in your model be displayed with or without translucency. If you want to control the level of translucency for part and assembly display, see *Changing the translucency*.

1. Locate the substructure translucency control option.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.

2. From the middle of the dialog box, toggle on **Always show substructure with translucency** from the set of mesh-related options.

3. + Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

• *Customizing geometry and mesh display*

Controlling beam profile display

If you use wire parts to model beams, you must create a beam section that refers to a beam profile, and you must assign the beam section to the wire part. In addition, you must assign a beam orientation to the wire part. You can then use the **View->Part Display Options** and the **View->Assembly Display Options** menu items to view a realistic display of the beam profile in parts and the assembly in the current viewport.

When beam profiles are displayed, Abaqus/CAE disables both view cuts and scaling and shrinking of the model. Displaying beam profiles is useful for checking that the correct profile has been assigned to a particular region and that the assigned beam orientation results in the expected orientation of the profile. For example, *Figure 1* shows the box beam profiles displayed on the light-service crane described in *Example: cargo crane*.

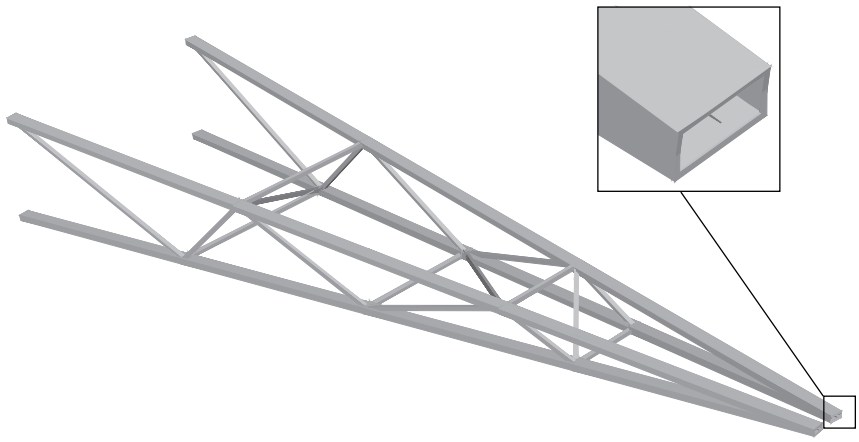


Figure 1: The cargo crane example with beam profiles displayed.

If you assign a general beam section to a wire, Abaqus/CAE displays the beam profile as an ellipse with a cross-sectional area and moments of inertia (I_{11} and I_{22}) that match the values you specified. If you assign a truss section to a wire, Abaqus/CAE displays the truss profile as a circle with a cross-sectional area that matches the value you specified.

Abaqus/CAE does not render the tapering of beam profiles along their length. If your model includes tapered beam sections, Abaqus/CAE renders these beams using the beam's starting profile along its entire length. For more information about tapered beams, see *Creating beam sections*.

Abaqus/CAE renders beam profiles according to the current settings for color coding and translucency. When these settings change, the color and translucency of the beam profiles change as well.

1. Locate the **Render beam profiles** option.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.
2. From the bottom of the dialog box, toggle on **Render beam profiles** to display beam profiles.
3. If desired, apply a **Scale factor** to the beam profiles to increase or decrease their size. The default value is 1.
4. + Click **OK** to implement your changes and to close the dialog box.

   Abaqus/CAE displays the profile of the beam section with the appropriate dimensions and in the correct orientation. Your changes are saved for the duration of the session.

Additional information

• *Defining profiles*

- *Customizing geometry and mesh display*
- *Controlling beam profile display for postprocessing*

Controlling shell thickness display

Displaying shell thickness enables you to examine the thickness of shell geometry relative to the rest of the model. You can apply a scale factor to reduce or increase the display of shell thickness for your session.

If you use shell elements to model relatively thin components in your analysis, you can use the **View->Part Display Options** and the **View->Assembly Display Options** menu options to view the actual thickness of these shell elements in your model. *Figure 1* shows the effect of scale factor changes displayed on the stiffened plate model described in *Example: blast loading on a stiffened plate*.

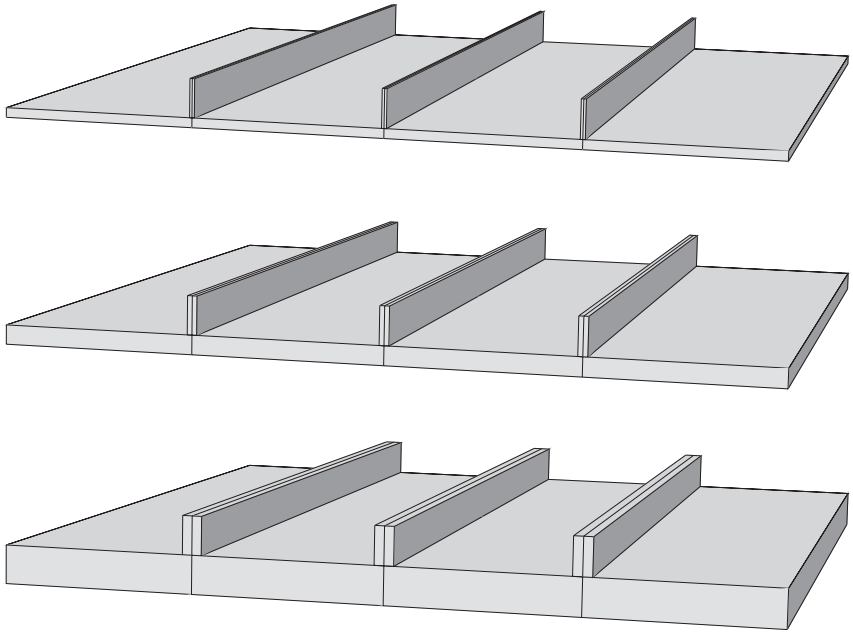


Figure 1: From top to bottom: shell thickness scale factor settings of 1 (default), 2, and 4.

Abaqus/CAE renders shell thickness for three-dimensional shell elements only; thickness is not displayed for axisymmetric shell elements, such as SAX1 elements. When shell thickness is displayed, Abaqus/CAE also renders the edges of shell geometry unless a view cut is displayed in the viewport. Abaqus/CAE renders shell thickness according to the current settings for color coding and translucency. When these settings change, the color and translucency of the shell thickness change as well.

If a discrete field has been used to define shell thicknesses or the shell offset, the **Render shell thickness** option has no effect. The shells are always displayed with zero thickness and no offset.

1. Locate the **Render shell thickness** option.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **General** tab.

2. From the bottom of the dialog box, toggle on **Render shell thickness** to display shell thicknesses for the shell sections in your model.
3. If desired, apply a **Scale factor** to the shell thicknesses to increase or decrease their thickness. The default value is 1, which produces a realistic rendering of the shell thickness settings.
4. + Click **OK** to implement your changes and to close the dialog box.

Abaqus/CAE displays the shell sections in the selected part or assembly with the appropriate thickness. Your changes are saved for the duration of the session.

Additional information

- *Defining sections*
- *Customizing geometry and mesh display*
- *Controlling shell thickness display for postprocessing*

Controlling datum display

You can use the **View->Part Display Options** and the **View->Assembly Display Options** menu items to control the display of datum geometry in parts and the assembly in the current viewport. You can control the display of each of the datum types—points, axes, planes, and coordinate systems—and you can toggle their display individually or all at once. Datum geometry that you choose not to display, although invisible, is still a feature of the part or assembly. For more information on datum geometry, see *The Datum toolset*. You can also control the display of reference points; for more information, see *The reference point*.

Datum geometry created on parts can be distracting when you are assembling instances of the part; turning off the display of datum geometry can result in a clearer display of the assembly. Similarly, turning off the display of datum geometry is useful for generating a clean printed image of a part or assembly.

1. Locate the **Datum** display options.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. Click the **Datum** tab in the dialog box that appears.

2. Toggle the appropriate buttons to control the display of:

   - Datum points
   - Datum axes
   - Datum planes
   - Datum coordinate systems

   Alternatively, click **Show all datums** to display all datum geometry in the viewport, or click **Show no datums** to hide all datum geometry in the viewport.

3. Click **OK** to implement your changes and to close the dialog box.

   Your changes apply only to the current viewport and are saved for the duration of the session.

Additional information

- *Customizing geometry and mesh display*

Controlling the display of individual coordinate systems

You can highlight, display, and hide individual coordinate systems in the viewport. Abaqus/CAE provides the following display options during modeling and postprocessing:

Controlling display of datum coordinate systems during modeling

All datum geometry you create, including datum coordinate systems, are considered features of the part or assembly to which they apply. Abaqus/CAE provides shortcuts for datum coordinate systems and other features in the Model Tree under the **Features** container for the part or assembly. You can highlight an individual datum coordinate system by clicking its shortcut in the Model Tree; when highlighted, a coordinate system is rendered in red in the viewport and its title is displayed. To hide or display the coordinate system in the viewport, click mouse button 3 on the shortcut and select **Suppress** or **Resume**.

You can also control the display of individual datum coordinate systems by adding them to display groups. See *Managing display groups* for more information.

Controlling display of any coordinate system during postprocessing

In the Visualization module the available coordinate systems are divided into two groups: ODB coordinate systems, which are part of the output database file; and session coordinate systems, which are created during postprocessing. Session coordinate systems apply to one output database only and persist only for your Abaqus/CAE session, unless you move the session coordinate system to the output database. See *Saving a coordinate system to an output database file*.

You can highlight, display, or hide individual coordinate systems using either of the following techniques:

- All available coordinate systems have shortcuts in the Results Tree under the **ODB Coordinate Systems** and **Session Coordinate Systems** containers. You can click any shortcut to highlight that coordinate system in the viewport; when highlighted, a coordinate system is rendered in red in the viewport and its title is displayed. You can also display or hide any coordinate system by clicking mouse button 3 on the shortcut and selecting a Boolean operator from the list that appears.
- You can control the display of ODB coordinate systems and session coordinate systems by adding them to display groups, which can be displayed or hidden using the Boolean display options. See *Managing display groups* for more information.

The **Coordinate System Manager** also provides information about the ODB coordinate systems and session coordinate systems for the selected output database. You cannot change the display of coordinate systems from this manager, but you can rename or delete them. See *Creating coordinate systems during postprocessing*.

Additional information

- *Creating coordinate systems during postprocessing*
- *Creating datum coordinate systems*
- *Customizing geometry and mesh display*

Controlling reference point display

You can use the **View->Part Display Options** and the **View->Assembly Display Options** menu items to control the display of reference points in the current viewport on a part or on the assembly. Turning off the display of reference points is useful for generating a clean printed image of a part or the assembly. Even though you choose not to display reference points, they are still a feature of the part or assembly. For more information on reference points, see *The reference point*.

1. Locate the **Datum** display options.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. Click the **Datum** tab in the dialog box that appears.

2. Toggle the appropriate buttons to control the display of:

   - Reference point labels
   - Reference point symbols

3. Click **OK** to implement your changes and to close the dialog box.

   Your changes apply only to the current viewport and are saved for the duration of the session.

Additional information

- *Customizing geometry and mesh display*

Customizing mesh display

You can use the **View->Part Display Options** and the **View->Assembly Display Options** menu items to specify whether or not node and element labels are displayed on a meshed part or assembly. You can choose to display or suppress your native mesh and, if displayed, to do so only in the Mesh module or in all part-related or all assembly-related modules. If you display the native mesh, you can choose to also display the geometry of bottom-up meshed sections, non-bottom-up sections, or both along with the mesh. Displaying the geometry provides a visual indication of how well the mesh conforms to the geometry.

1. Locate the mesh display options.

   From the main menu bar, select **View->Part Display Options** or **View->Assembly Display Options**. In the dialog box that appears, click the **Mesh** tab.

2. Toggle **Show native mesh** to display or suppress the native mesh.

   When **Show native mesh** is on, you can also control the following options:

   a. Select one or both of the following options to display geometry with the native mesh:

      - Toggle on **Show bottom-up geometry** to display the geometry of regions that have been assigned the bottom-up meshing technique. This option is on by default.
      - Toggle on **Show non-bottom-up geometry** to display the geometry of regions assigned all other meshing techniques. This option is off by default.

   b. Select one of the following options to control the modules in which you can display the native mesh:

      - Choose **In the Mesh module only** to display your native mesh only in the Mesh module.
      - Choose **In all part-related modules** from the **Part Display Options** dialog box or **In all assembly-related modules** from the **Assembly Display Options** dialog box to display your native mesh in all modules that support the display of the part or assembly, respectively.

3. Toggle **Show node labels** and **Show element labels** to affect the display of these items.
4. + Click **OK** to implement your changes and to close the dialog box.

   Your changes are saved for the duration of the session.

Additional information

• *Customizing geometry and mesh display*

Controlling model lighting

You can use the **View->Light Options** menu item to control the lighting of your model. Lights affect the appearance of your model in the current viewport when the **Shaded** render style is used. You can control the intensity of the **Ambient** light and the **Shininess** of the model surface. You can also control the locations and intensities of up to eight additional positional lights. The combined effects of the light settings can be used to simulate different surface finishes and light conditions on your model. The default settings provide good contrast for viewing all features in most models.

The default settings provide good contrast for viewing all features in most models. Using the default settings is particularly important for tessellated, banded contour plots, for which intense light can fade the colors in the facets of the contour plot and display misleading results that do not match the colors in the plot legend. These changes to contour colors can also appear in printouts of banded contour plots if you print to file in EPS, PostScript, or SVG formats, because these three formats use tessellated contours by default. To print a banded contour plot to any of these formats without creating misleading contour colors, turn off shading before printing.

Global settings

Ambient light is applied evenly to the entire scene and brightens a model from all directions. Low intensity ambient light allows the positional lighting to create shading that distinguish small features and surface contours from the rest of the model. High intensity ambient light removes shading and may make some model features difficult to see.

If your computer's graphics card supports OpenGL shading language (GLSL), Abaqus/CAE also reveals the global **Shading** options, from which you can enable Phong shading for your session. Phong shading renders more realistic shading on three-dimensional surfaces than the default option, Gouraud shading, but it can cause a noticeable performance impact on some systems. This performance impact occurs only when the mesh is hidden; if the mesh is displayed (either during modeling or postprocessing), Abaqus/CAE renders Phong lighting with no noticeable performance impact.

Shininess is the reflectivity of the model surface and is used to control the size of specular highlights from the positional lights. A very shiny surface reflects light like a mirror—the light reflects in one direction based on the incident angle of the light source. Therefore, a surface must be positioned correctly with respect to the light source to reflect light to your viewpoint. Surfaces that are less shiny reflect light more randomly, so a wider range of surface angles can reflect light to your viewpoint. Like high intensity ambient lighting, low shininess can obscure small features and contours of your model from view.

The **Viewpoint** option controls how specular lighting effects are calculated. An **Infinite** viewpoint assumes a constant vector for the direction from the camera to each point when calculating reflections and specular highlights at a point. A **Local** viewpoint calculates a separate direction vector for each point based on its position in the viewport. A **Local** viewpoint creates more realistic lighting effects but can decrease overall performance.

Positional light settings

Positional lights provide a light bulb effect. A positional light is projected onto the model from the specified location, and its effects change as you rotate your model view. Positional lights work in conjunction with shininess to determine how your model reflects light.

The distance of the light from the model is equal to the distance from the camera to the model. You can position the lights by specifying their **Latitude** and **Longitude** on a hemisphere around the model. You can also specify the **Type** of light source being used. A **Directional** light is a planar light source; the angle of incident light on the model will be equivalent for all parallel surfaces. A **Point** light is a point light source; the angle of incident light depends on the location of the surface or point relative to the light. A **Point** light creates more realistic lighting effects but can decrease overall performance.

You can control two different qualities for positional lights: **Diffuse** intensity and **Specular** intensity. The **Diffuse** setting controls the intensity of a positional light. Unlike ambient light, surfaces that face toward the light position

will brighten more readily than other surfaces in the model when increasing the diffuse intensity of a positional light. The **Specular** setting controls the brightness of those surfaces that are reflecting from the light toward the viewpoint. You should first set the position and diffuse intensity of the positional light to achieve the shading you desire. Then you can adjust the brightness of reflected light with the **Specular** slider.

1. Locate the lighting options.

   From the main menu bar, select **View->Light Options**.
2. If the **Shading** options are displayed, select **Gouraud** or **Phong** shading.
3. From the **Viewpoint** field, select **Infinite** or **Local** to define the viewpoint type.
4. Use the slider to set the desired **Ambient** light intensity.
5. Use the slider to set the desired **Shininess** for the model surface. A higher number corresponds to a shinier surface.
6. Toggle on a number in the **Lights** field to add a positional light to the scene.
7. To change the settings for a positional light, click the associated number tab in the **Lights** field.
   - From the **Type** field, select **Directional** or **Point** to define the light type.
   - Use the sliders to set the desired **Latitude** and **Longitude** for the light's position.
   - Use the slider to set the desired **Diffuse** intensity.
   - Use the slider to set the desired **Specular** intensity.
8. Click **Defaults** to revert all lighting to the default settings.
9. Click **Dismiss** to close the dialog box.

   Your changes are saved for the duration of the session. To save the settings for future sessions, select **File->Save Options** from the main menu bar (see *Saving your display options settings*).

Additional information

• *Customizing geometry and mesh display*

Controlling instance visibility

By default, Abaqus/CAE displays all part instances included in the assembly. You can turn the display of all instances on or off, or you can toggle the display of individual instances. Part instances that you have suppressed or that do not belong to the current display group cannot be made visible using this dialog box; you must use the Feature Manipulation toolset or the Display Group toolset instead. For more information, see *The Assembly module* or *Using display groups to display subsets of your model*.

This section describes how to control instance visibility from the **Assembly Display Options** dialog box. You can also change instance visibility from the Model Tree or the viewport: from the Model Tree, highlight the part instances that you want to display or hide, click mouse button 3, and select **Hide** or **Show**; from the viewport, highlight the instances you want to hide, click mouse button 3, and select **Hide Instance** from the menu that appears.

1. Locate the **Instance** display options.

   From the main menu bar, select **View->Assembly Display Options**. Click the **Instance** tab in the dialog box that appears. The **Instance** options become available; each part instance in the assembly is listed.

2. To control instance visibility, do any of the following:

   - Click **Set All On** to make all (except suppressed) instances visible.
   - Click **Set All Off** to turn off the display of all instances.
   - Click individual instance names to toggle their appearance.

3. Click **OK** to implement your changes and to close the dialog box.

   Your changes apply only to the current viewport and are saved for the duration of the session.

Additional information

• *Customizing geometry and mesh display*

Controlling the display of attributes

The **Attribute** display options in the **Assembly Display Options** dialog box allow you to control the display of symbols representing

- interactions, constraints, and connectors that you create in the Interaction module,
- loads, boundary conditions, and predefined fields that you create in the Load module,
- engineering features that you create in the Property module and Interaction module, and
- optimization attributes that you create in the Optimization module.

You can control when and how these attributes are displayed, and you can click **Set all on** or **Set all off** to display or hide all attributes, respectively. For information on the symbols representing each attribute, see *Special graphical symbols*.

For information on controlling the display of boundary conditions, coupling constraints, connectors, and point elements in the Visualization module, see *Controlling the display of model entities*.

1. Locate the **Attribute** display options.

   From the main menu bar, select **View->Assembly Display Options**. Click the **Attribute** tab in the dialog box that appears. The **Attribute** options become available.

2. Click the **Main** tab to specify which attributes you want to display and in which modules you want them to appear.

   a. Select the **Show attribute in** option.

      - Select **Module in which it was created** to display attributes only in the modules in which they are created. For example, loads would appear only in the Load module and interactions only in the Interaction module.
      - Select **All assembly-related modules** to display attributes in all modules that support the display of the assembly.

   b. From the **Show** list, select the attributes that you want to display. Only those attributes that you select will appear in the viewport; for example, if you toggle on **Loads** and **BCs**, only load and boundary condition symbols will appear in the viewport. You can also select all categories and all types within each category by clicking **Set all on**, or you can deselect all categories and all types within each category by clicking **Set all off**.

3. Click the **Symbol** tab to control the size and density of attribute symbols. You can also reduce the number of attribute symbols displayed on orphan mesh regions to a fraction of the maximum allowable number.

   a. Specify your **Size** preferences. The higher the size setting, the larger the symbols appear in the viewport.

      - Drag the **Arrows** slider to specify the size of arrow symbols.
      - Drag the **Other symbols** slider to specify the size of all symbols other than arrows.
      - Toggle off **Scale symbols based on analytical field value** to remove the symbol scaling for attributes that specify analytical fields. For more information, see *Displaying symbols for interactions and prescribed conditions that use analytical fields*.

   b. If you are working with geometry, specify the desired density of the attribute symbols. The higher the density setting, the more symbols Abaqus uses to represent each attribute.

      - Drag the **Face density** slider to control the density of symbols that appear on faces.

- Drag the **Edge density** slider to control the density of symbols that appear on edges.

The effect of changing a symbol density setting may vary depending on the size of the region in the viewport.

c. If you want to reduce the density of attribute symbols displayed on orphan mesh regions, enter a value between 0 and 1 in the **Fraction of symbols displayed on orphan mesh regions** field. The higher the fraction value, the more symbols Abaqus/CAE uses to represent each attribute. Choosing the default density of 1 prompts Abaqus/CAE to draw symbols on every mesh entity within the region.

4. Click the tab of the attribute of interest to specify which particular attribute categories and types you want to appear in the viewport.

For example, if you click the **Load** tab, a list of load categories appears. If you click the arrow next to one of the categories, a list of all the load types within that category appears.

Use the following techniques to specify the attribute categories and types that you want to display:

- Click the arrow next to the category of interest. From the list of types that appear, select those types that you want to display.
- Toggle the desired category. This action selects or deselects all types within that category.
- Click **Set All On** to select all categories and all types within each category.
- Click **Set All Off** to deselect all categories and all types within each category.

The check box next to a category label becomes a black check mark on a white background when all types in that category are selected. The check box becomes light gray and the check mark becomes dark gray if only some of the types within that category are selected.



Note:

The attribute categories and types that you specify only appear in the viewport if you toggled on that attribute in Step 2.

5. Repeat the previous step as necessary to display specific categories and types of other attributes of interest.

6. At the bottom of the **Assembly Display Options** dialog box, click **OK** to implement your display settings in the current viewport and to close the dialog box.

Additional information

- *Customizing geometry and mesh display*
- *Special graphical symbols*

Saving your display options settings

If you change your display options settings (for example, if you change the render style or turn off the display of datum planes), you can store the new settings for future sessions. Abaqus/CAE saves your settings in a file called `abaqus_2025.gpr`. For more information, see *Working with abaqus_2025.gpr files*.

From the main menu bar, select **File->Save Display Options** to save all the current display options settings. The **Save Display Options** dialog box appears and allows you to save the options in the current directory or in your home directory.

You can edit the `abaqus_2025.gpr` file using API commands in the Abaqus Scripting Interface; for more information, see *Editing display preferences and GUI settings*. You can also delete the file to restore the default GUI and display options settings, and you can copy the file to other directories on your computer or transfer the file to a different computer. When you save your settings from `abaqus_2025.gpr`, Abaqus/CAE always saves all the current settings and always overwrites all the settings that were previously saved. You cannot save only selected settings. You can use the **noSavedOptions** command line option to start Abaqus/CAE without loading the settings in `abaqus_2025.gpr`. For more information, see *Starting Abaqus/CAE (or Abaqus/Viewer)*.

Abaqus/CAE saves the following display options settings in `abaqus_2025.gpr`:

- Part and assembly display options; for example, render style, the visibility of the various types of datum geometry and simulation attributes, and the display of mesh, element, and node labels.

  Note:

  The settings in the **Main** tab of the **Assembly Display Options** dialog box are not saved in `abaqus_2025.gpr`; however, the settings for the simulation attribute categories and types specified on other tabs (such as **Interaction** and **Load**) are saved in `abaqus_2025.gpr`. For more information, see *Controlling the display of attributes*.

- Graphics options and viewport annotation options. Abaqus/CAE also saves the perspective setting.
- Print options.
- Display options in the Visualization module; for example, the contour type for contour plots and the render style and fill color in the common plot options.
- Animation options in the Visualization module.
- Other options in the Visualization module, such as probe, field report, *X*–*Y* plot, and *X*–*Y* report options.
- Options selected in the **Linked Viewports Manager**.

Color coding geometry and mesh elements

This section explains how you can apply color coding to viewable geometry and mesh elements.

In this section:

- *Understanding color coding*
- *Changing the initial color*
- *Changing the translucency*
- *Coloring geometry and mesh elements*
- *Coloring all geometry in the Visualization module*
- *Coloring nodes or elements in the Visualization module*
- *Coloring constraints in the Visualization module*
- *Customizing the display color of individual objects*
- *Displaying multiple color mappings*
- *Editing the colors in the Auto-Color List*
- *Saving and restoring custom color mappings*

Understanding color coding

This section discusses how to use color coding to distinguish between components in your model or output database.

In this section:

- *Color coding concepts*
- *Color coding in the Visualization module*

Color coding concepts

You can distinguish between components in your model or output database by color coding the viewable geometry and mesh elements in the current viewport. Abaqus/CAE applies color coding according to **Color Mappings** that specify the colors assigned to each item for a particular type of object, such as parts, section assignments, boundary conditions, or display groups. In the example shown in *Figure 1* the color mapping is by parts, and each row describes the color assigned to one of the three part definitions in this model. The **Color Code** dialog box thus provides a legend that describes all of the color coding currently displayed in the current viewport.

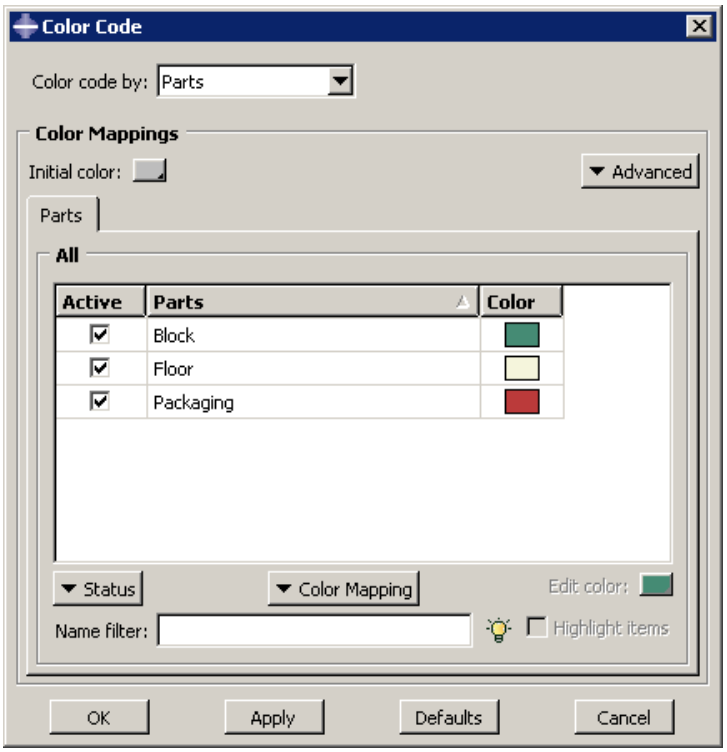


Figure 1: The Color Code dialog box.

Color coding is applied in two layers. All geometry and mesh elements are first colored with the **Initial color**, a customizable setting that is grey by default. (See *Changing the initial color*, for more information.) Abaqus/CAE then applies color coding on top of the initial color according to the colors and objects in the color mapping that you select. Areas will remain visible in the initial color if you apply a color mapping such as Boundary conditions, Loads, or Sets, for which Abaqus/CAE usually color codes only distinct points or surfaces in the model.

Abaqus/CAE automatically creates color mappings for all items in your model by associating the name of each item with a color in the **Auto-Color List**. The colors are selected by matching the **Auto-Color List** with an alphabetical listing of item names; thus, in the **Parts** example in *Figure 1***Block** is color coded with the first color in the **Auto-Color List**, **Floor** is color coded with the second color, and so on. Color assignments depend on the item name only; therefore, resorting these items in the **Color Code** dialog box does not change the color assignments.

Note:

If a region is shared by two or more items in the model, such as when a skin section is assigned to the surface and a solid section is assigned to the entire block, the common region will be colored with the color definition for the item whose name appears later in the alphabetical list. Abaqus/CAE overwrites the first color mapping color definition as each color assignment is displayed in the viewport.

Because each color mapping is a set of links between item names and color definitions, color mappings persist between modules in Abaqus/CAE and between a model database and its output databases. However, because Abaqus/CAE relies on object names for color coding, you cannot retain the color coding associated with an object when you rename that object. By contrast, Abaqus/CAE usually deletes an object's color definition when you delete the object definition from your model; the two exceptions are material and section definitions, whose color coding persists in the viewport even after you delete the definition. To refresh the color coding in the current viewport after you delete a material or section definition, you must either apply a color mapping or switch to a different module.

Abaqus/CAE also provides a default color mapping for each module. For example, when you display the **Mesh defaults** color mapping in the Mesh module, Abaqus/CAE color codes items in the viewport according to their meshability. Each module's default color mapping is available only in that module; you cannot color code objects in the Property module using the assignments in the **Mesh defaults** color mapping. Module default mappings cannot be edited, and the module default mappings correspond to the default colors that Abaqus/CAE uses if no color coding is applied.

Color mappings are viewport-specific and, in some cases, they persist between modules. Persistence of color coding between modules depends on whether you have the default color mapping displayed for your current module:

- If the default color mapping for your module is selected, Abaqus/CAE automatically changes the color mapping as you switch modules. For example, if you select the **Assembly defaults** color mapping in the Assembly module and then switch to the Mesh module, Abaqus/CAE automatically applies the **Mesh defaults** color mapping.
- If a non-default color mapping is selected, Abaqus/CAE retains the color mapping as you switch modules. For example, if you select the **Part instances** color mapping in the Assembly module and then switch to the Mesh module, Abaqus/CAE continues to color code by part instance name rather than by meshability.

When you change the color mapping, Abaqus/CAE refreshes color coding only in the current viewport while retaining any color coding in other inactive, visible viewports. When you add a new viewport to your session, the new viewport inherits the color mappings of the previously current viewport.

Once color mappings are created, you can customize any color mapping (except the module defaults) by changing the colors assigned to any of its individual objects. You can also toggle the active status of individual objects in the color mapping, which controls whether an individual object is color coded in the current viewport. Objects whose color coding is inactive are rendered with the initial color. The **Color Code** dialog box also provides sorting and filtering tools that you can use to display a subset of the objects in a color mapping. These tools can help you focus your display when a color mapping has many objects.

Color coding in the Visualization module

The color coding process is slightly different in the Visualization module than in other Abaqus/CAE modules. You control the overall edge and fill colors of your model using the **Color & Style** options in the **Common Plot Options** or **Superimpose Plot Options** dialog box. Using these options, you select a single color for all element and surface edges and a separate color for all element faces and surfaces. The colors you select apply uniformly to the entire model.

You can also control the colors of individual elements and part instances using the **Color Code** options. The **Color Code** dialog box allows you to select separate colors for individual items. You must use the **Color Code** options to execute any complex, nonuniform color scheme.

By default, individual item colors override the overall common or superimposed *edge color* and *fill color*. You can change this behavior by using the **Color & Style** options in the **Common Plot Options** or **Superimpose Plot Options** dialog box to specify whether individual or overall item colors should take precedence. (Individual item colors do not apply to contour plots.)

The Visualization module offers a smaller subset of color mappings than are available in the other modules. When an output database is displayed in the current viewport, the available color mappings for color coding are part instances, elements, nodes, constraints, materials, sections, display groups, averaging regions, internal sets, layups, and plies; when a model in the current model database is displayed, the only available color mapping is by part instance. However, you can control both the edge and fill colors when you customize colors for the individual objects in the current viewport, and you can choose to apply color coding to the entire model or to its nodes or elements only. In addition, settings in the Visualization module depend upon other options; when you specify an individual item color in the **Color Code** dialog box, Abaqus/CAE applies the color based on two characteristics of the current plot:

Color precedence setting

Abaqus/CAE applies the color if, on the **Color & Style** page of the **Common Plot Options** or **Superimpose Plot Options** dialog box, **Allow color code selections to override options in this dialog** is toggled on.

Render style

In the wireframe and hidden render styles, Abaqus/CAE displays only element edges. If the current plot uses the wireframe or hidden render style, Abaqus/CAE applies the individual item color to those edges.

In the filled and shaded render styles, Abaqus/CAE displays element faces as well as element edges. If the current plot uses the filled or shaded render style, Abaqus/CAE applies the individual item fill color to the element faces and the individual item edge color to the element edges.

In the filled and shaded render styles, line-type elements (such as beams) are treated as if their lines are faces. In the filled and shaded render styles, Abaqus/CAE applies the individual item fill color to the lines representing such an element.

Changing the initial color

Abaqus/CAE begins the color coding process by applying an **Initial color** to the viewable geometry and mesh elements in the current viewport. By default, this initial color is grey, but you can customize the initial color by selecting one of the following from the **Initial color** field:

- You can select the current color, which prompts Abaqus/CAE to retain the current colors displayed in the viewport. Selecting this option does not change any colors; therefore, this option is most useful when you do not want to change a geometry or mesh element color that you have already applied in the viewport.
- You can set the initial color to display the default color mappings for your selected module. This option is useful for resetting individual selections to their default colors without removing all color changes from the viewport. For example, when you choose this option in the Assembly module, Abaqus/CAE color codes components of the assembly according to the assembly module default settings: **Display body**, **Geometry**, **Native mesh**, and **Orphan mesh**.
- You can select a custom color as the initial color.

If you are working in any module other than the Visualization module, you can use the translucency tool in the **Color Code** toolbar to make the initial color translucent and to select a level of translucency. For more information, see *Changing the translucency*.



Note:

In the Visualization module, you control translucency from the **Other** page of the **Common plot options** or **Superimpose plot options** dialog box. See *Customizing render style, translucency, and fill color*, for more information.

1. Click the  tool in the Color Code toolbar.
   Abaqus/CAE displays the **Color Code** dialog box.
2. Click and hold the **Initial color** sample, and then select one of the following choices from the list that appears:
   - Select the equal sign (=) to select the current color.
   - Select the asterisk (*) to select the default color mapping for the module.
   - Select the color sample () to select a custom color. Choosing the color sample symbol opens the **Select Color** dialog box for you to choose a new color. (For more information, see *Customizing colors*.)
3. Click **Apply**.
   Abaqus/CAE displays the new initial color selection in the current viewport.

Changing the translucency

By default, Abaqus/CAE displays geometry and mesh elements in the shaded render style using opaque colors. Interior features and features that are “behind” other objects in the viewport are not visible.

In some cases, Abaqus/CAE applies translucency to the model to help you select interior or hidden entities during a procedure. You can also use the  tool in the **Color Code** toolbar to toggle translucency on and off when it is not required by a procedure.

To set the percentage of translucency that Abaqus/CAE uses, click the arrow to the right of the  tool. Abaqus/CAE displays a vertical slider. Drag the slider up to make the display colors more opaque or down to make them more transparent.



Note:

In the Visualization module, you control translucency from the **Other** page of the **Common plot options** or **Superimpose plot options** dialog box. See *Customizing render style, translucency, and fill color* for more information.

Coloring geometry and mesh elements

You can apply color coding to geometry and mesh elements from any module other than the Visualization module. If you want to apply color coding in the Visualization module, see either *Coloring all geometry in the Visualization module*, or *Coloring nodes or elements in the Visualization module*.

Abaqus/CAE applies color coding to the current viewport according to color mappings for specific types of objects, such as part instances, materials, sections, or display groups. This section describes how to examine the assignments in a particular color mapping and how to apply the assignments in a color mapping to the current viewport.

Abaqus/CAE provides two methods that you can use to apply color coding to predefined object types in the current viewport. You can quickly select a color mapping by choosing its name from the list immediately to the right of the  tool in the Color Code toolbar. Abaqus/CAE refreshes the current viewport with the color coding specified in that color mapping. Alternatively, you can select the color mapping from the **Color Code** dialog box. You must use the dialog box if you want to customize the color mapping. See *Customizing the display color of individual objects*, for a description of the options provided for changing the colors assigned to individual objects.

Additional information

- *Changing the initial color*
- *Coloring nodes or elements in the Visualization module*
- *Customizing the display color of individual objects*

Apply a color mapping using the toolbar list

1. Locate the list of color mappings.

   The list resides immediately to the right of the  tool in the Color Code toolbar.

2. Select a color mapping from the list.

   Abaqus/CAE color codes the geometry and mesh settings according to the colors displayed in the **Color Code** dialog box.

Apply a color mapping using the Color Code dialog box

1. Click the  tool in the Color Code toolbar.

   Abaqus/CAE displays the **Color Code** dialog box, which displays the default color mappings for the current module.

2. If you are color coding geometry or mesh elements, select a color mapping from the **Color Code by** list. (If you are performing a different color coding operation, refer to either *Coloring all geometry in the Visualization module*, or *Coloring nodes or elements in the Visualization module*.)

   Abaqus/CAE displays the selected color mapping in the **Color Mapping** portion of the dialog box.

3. Click **Apply**.

   Abaqus/CAE color codes the current viewport according to the colors displayed in the **Color Code** dialog box.

Coloring all geometry in the Visualization module

This section describes how to apply color coding to all viewable geometry in the current viewport when you are working in the Visualization module. To apply color coding to geometry and mesh elements from any of the other Abaqus/CAE modules, see *Coloring geometry and mesh elements*. To color code by selected nodes or elements in the Visualization module instead of the entire geometry, see *Coloring nodes or elements in the Visualization module*.

When an output database is selected in the Visualization module, you can select any of the following color mappings when you apply color coding to all of the geometry:

- **Part instances**
- **Element sets**
- **Materials**
- **Sections**
- **Element types**
- **Averaging regions**
- **Internal sets**
- **Composite layups**
- **Composite plies**

When a model in the current model database is selected in the Visualization module, only the **Part instances** color coding option is available.

Abaqus/CAE lists all the items for the current selection method in the **Color Mappings** table. Once you select a color mapping in the **Color Code** dialog box, you can also customize the display color of its individual items. See *Selecting multiple items from lists and tables*, for more information. You make your selections directly from the table, and you can select multiple cells; for more information, see *Selecting multiple items from lists and tables*.

1. Click the  tool in the Color Code toolbar.

   Abaqus/CAE displays the **Color Code** dialog box, which displays the default color mappings for the current module.
2. From the **Color Code** list, select **All**.
3. From the **By** list, select the name of the color mapping that you want to apply.
4. Click **Apply**.

   Abaqus/CAE color codes the current viewport according to the colors displayed in the **Color Code** dialog box.

Coloring nodes or elements in the Visualization module

When an output database is selected in the current viewport, you can color code selected **Nodes** and **Elements** to customize the display of your results in a viewport. For step-by-step instructions on using the **Color Code** dialog box, see *Coloring geometry and mesh elements*. To color code by selected item attributes instead of nodes or elements, see *Customizing the display color of individual objects*.

For the **Element** and **Node** item types, your item choices vary with the method that you select from the **By** list at the top of the **Color Code** dialog box. Some selection methods require you to complete the information in the **Color Mappings** table. Regardless of whether the **Color Mappings** table is completed by Abaqus/CAE or by you, once it is complete, you can select multiple cells to change node and element colors, node symbol shapes, and node symbol sizes (see *Selecting multiple items from lists and tables* for more information).

When you change node and element colors, you must select the colors you want from the **Select Color** dialog box. Abaqus/CAE does not support automatic color coding for nodes and elements. In addition, the columns in the **Color Mappings** table are not sortable when you examine color mappings for **Nodes** and **Elements**; consider using the filter if you need to find node or element names from a large list.

The **Color Code** dialog box does not retain node and element color selections when you close the dialog box or switch viewports. You should consider saving color macros frequently when you change node and element colors; see *Saving and restoring custom color mappings*, for more information.

Choose from the following selection methods to color elements and nodes in your model:

Pick from viewport

Select **Pick from viewport** to specify elements or nodes by picking them directly from the viewport. Click **Edit Selection** or **Add Selection**, respectively, to edit an existing row or to add a new row to the **Color Mappings** table. Abaqus/CAE automatically enters pick mode, and **Select *items* for color coding** appears in the prompt area. See *Selecting objects within the viewport* for more information on picking items in the viewport. Click **Delete Selection** to delete a highlighted row from the table.

Element labels (Node labels)

Select **Element labels** or **Node labels** to specify elements or nodes by number. For each row in the **Color Mappings** table, select the name of the part instance to which the nodes or elements belong from the list in the **Part instance** column and type a list of element or node numbers separated by commas or a range of numbers such as `1:4` into the **Labels** field. If desired, you can specify a range using a number other than `1` as the operator; for example, `1:21:5` selects items labeled `1`, `6`, `11`, `16`, and `21`.

Result value

Select **Result value** to specify elements or nodes containing results within a given range of values.

The output variable to be considered is displayed at the bottom of the **Color Mappings** table, to the right of the **Field Output** button. To select a new result variable, click **Field Output**; see *Selecting the field output to display* for more information on the **Field Output** dialog box. Choose from the filtering methods in the **Type** cell; the symbols represent results below , inside , outside , or above  the selected value or range. Enter the required value or values in the **Min value** and **Max value** cells to specify the range for the filter type that you selected. You can add rows to the table and select different filtering methods and result ranges, but all rows refer to the same field output variable. Every element (or node) in the model is used in the filtering process, regardless of the current active display group in the model.



Note:

The bounds for filtering based on element or nodal output variables are always based on the values of a variable at the nodes. Therefore, element-based output quantities are extrapolated and averaged at the nodes before comparing them against the user-defined bounds. The averaging settings in the **Result Options** dialog box determine how element-based variables are calculated at the nodes. For example, consider a case where elements are filtered based on Mises stress using the default averaging threshold of 75%. After extrapolation to the nodes, the values are averaged according to this threshold. This conditional averaging may result in several distinct values of Mises stress at the node based on contributions from the various elements to which the node belongs. Any element whose Mises stress contribution falls within the user-defined bounds is included in the color coding selection.

All elements (All nodes)

Select **All elements** or **All nodes** to select all items of the specified type in your model. No further item specification is necessary.

Node sets

Select **Node sets** to specify a new color for node sets saved in your model. The **Color Mappings** table lists all the available set names. You can select element sets by using the **All** item type.

Internal sets

Select **Internal sets** to specify internal (created by Abaqus/CAE) node or element sets. The **Color Mappings** table lists all the available set names.

Display groups

Select **Display groups** to specify a saved display group. The **Color Mappings** table lists all the available display group names.

Part instances

Select **Part instances** to specify a new color for all nodes in the selected part instance. The **Color Mappings** table lists all part instances in the model.

To select all the elements in a part instance, use the **Part instances** method for the **All** item type.

Additional information

- *Coloring geometry and mesh elements*
- *Customizing the display color of individual objects*

Coloring constraints in the Visualization module

You can apply color coding to constraints displayed in the current viewport when you are working in the Visualization module and when an output database is selected.
To color code all geometry in the Visualization module, see *Coloring all geometry in the Visualization module*.

1. Click the  tool in the Color Code toolbar.

   Abaqus/CAE displays the **Color Code** dialog box, which displays the default color mappings for the current module.
2. From the **Color Code** list, select **Constraints**.
3. In the **Constraint Types** list, edit any of the assigned colors.
4. Click **Apply**.

   Abaqus/CAE color codes the current viewport according to the selections in the **Color Code** dialog box.

Customizing the display color of individual objects

This section describes how to customize the display of individual objects; these examples apply throughout Abaqus/CAE, including in the Visualization module. Once Abaqus/CAE creates the automatic color mappings for your viewport, you can customize the color display by changing any of the colors assigned to individual objects in the color mapping. The **Color Code** dialog box displays each object and its assigned color on the same row in the **Color Mappings** table. For each object, you can select a different color, activate or deactivate its display, and set (or revert to) the default colors for the selected objects. The **Color Code** dialog box also provides options that support the management of colors for each object: you can filter by object name, sort table columns, and highlight the selected object.

In the Visualization module, you can apply color coding to the entire viewport or to the nodes or the elements only. You cannot color code nodes or elements in a model database in the other Abaqus/CAE modules.

The following options are available for customizing the display color of individual objects:

Changing the fill color for an individual item

The **Color** column displays the fill color assigned to each object. To choose a different color, either double-click the color sample  in that row or highlight the color sample and click the **Edit color** color sample. Abaqus/CAE opens the **Select Color** dialog box, from which you can select a new fill color. (For more information about selecting custom colors, see *Customizing colors*.) When you click **Apply**, Abaqus/CAE refreshes the current viewport with your new fill color.

Changing the edge color for an individual item (Visualization module only)

When you open the **Color Code** dialog box from the Visualization module, the dialog box includes the **Edge** column, which displays the edge color assigned to each object. You cannot customize the edge color in any other module.

To change the edge color for an individual item, either double-click the color sample  in that row or highlight the color sample and click and hold the **Edit color** symbol (=, *, or ). Abaqus/CAE opens the **Select Color** dialog box, from which you can select a new edge color. (For more information about selecting custom colors, see *Customizing colors*.) When you click **Apply**, Abaqus/CAE refreshes the current viewport with your new edge color.

Activating and deactivating table rows

Abaqus/CAE displays color coding for active objects only; inactive objects are rendered with the initial color. Deactivating color coding for several objects in a color mapping can simplify your display and help you examine or debug a model.



Note:

You cannot deactivate color coding for objects in the module default color mappings.

To toggle the activation status of color coding for an object, click **Status** and choose an activation or deactivation option from the list that appears. If you select **Activate all** or **Deactivate all**, Abaqus/CAE toggles the status of the **Active** column for all rows in the **Color Mappings** table. If you select **Activate selected** or **Deactivate selected**, Abaqus/CAE toggles the status of the **Active** column for the highlighted rows only. You can also select and deselect the checkboxes in the **Active** column to toggle the status of color coding for the objects in those rows. When you click **Apply** in the **Color Code** dialog box, Abaqus/CAE applies color coding to the current viewport for active rows only.

Highlighting selected objects

Select the **Highlight items** checkbox to display a highlighted border around any items whose rows are selected in the **Color Mappings** table. Abaqus/CAE enables highlighting only for the following color mappings: **Part instances**, **Sets**, **Surfaces**, **Internal sets**, and **Internal surfaces**. You cannot highlight objects when you apply color coding to nodes or elements.

Reverting to default colors and setting new defaults

After you make changes to the display color of individual objects, you can revert to the default color settings for the current color mapping by clicking **Color Mapping**, and then selecting **Restore Defaults** from the list that appears. Alternatively, if you want your changes to become the new default color mappings for your session, select **Set As Defaults** from the same list.

Auto-coloring individual objects

You can apply auto-coloring to selected objects in a color mapping. Select the individual objects that you want to change, then select **Auto-color Selected** from the **Color Mapping** list. Abaqus/CAE applies color coding from the auto-color list according to the alphabetical order of the names in the selected rows.

Sorting by columns

Click the column headings in the **Color Mappings** table to sort the table by the contents of the selected column. Click the same heading a second time to reverse the sorting order.



Note:

You cannot sort by columns when you color code nodes or elements.

Changing the sort order of the objects in the **Color Mappings** table is for navigation only; their order in the dialog box has no effect on the colors that are selected for color coding. Abaqus/CAE assigns colors from the auto-color list to model definitions in alphabetical order, based on the names of the items in the color mapping.

Filtering rows by name

If your selected color mapping contains many rows, you can use the **Name filter** to reduce the number of rows displayed. Click  next to the **Name filter** field to see examples of valid filtering syntax.

Displaying multiple color mappings

You can apply color coding for two or three different color mappings to the viewport at the same time. Displaying multiple color mappings can reveal interactions between aspects of the model that might not be apparent when you display the color mappings separately. For example, you might want to display color mappings for both **Boundary conditions** and **Loads** in the same viewport.

To add another color mapping to the **Color Mappings** portion of the dialog box, click **Advanced** and select **Add Mapping** from the list that appears. Abaqus/CAE adds another tabbed color mapping to the dialog box, in which you can display any color mapping by selecting the mapping from the **Color Code by** list. *Figure 1* shows the **Color Code** dialog box with the **Boundary conditions** and **Loads** color mappings displayed.

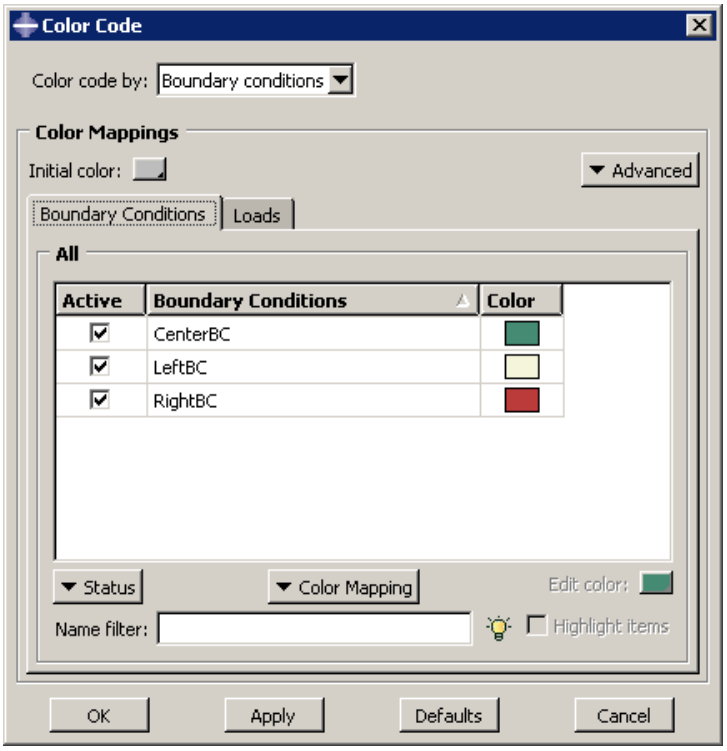


Figure 1: The Color Code dialog box with multiple color mappings.

You can also remove a color mapping from the dialog box by clicking its tab and selecting **Remove Current Mapping** from the **Advanced** list.

When multiple color mappings are open, Abaqus/CAE applies color mappings for the leftmost color mapping first, before proceeding to the mapping or mappings to its right. You can display up to three color mappings at a time, but Abaqus/CAE does not permit you to reorder the tables. If you want to color code attributes in a different order, you must close each color mapping and then add the color mappings in the order in which you want Abaqus/CAE to display them in the viewport.

Editing the colors in the Auto-Color List

When you create color mappings, Abaqus/CAE assigns colors to objects by referring to the color definitions in the **Auto-Color List**. You can modify the contents of this list by adding, removing, rearranging, and changing the colors.

Unlike color mappings, the Auto-Color List is common for all viewports.

1. Click the  tool in the Color Code toolbar.

   Abaqus/CAE displays the **Color Code** dialog box.
2. Click **Advanced**, and select **Edit Auto-colors** from the list that appears.

   Abaqus/CAE displays the **Edit Auto-Colors** dialog box.
3. To insert a new color into the Auto-Color list:

   1. Highlight the color in the Auto-Color list that you want to precede or follow the color that you add.
   2. Click **Insert Before** or **Insert After** to add a new color in the indicated location in the Auto-Color list.

      The **Select Color** dialog box opens.
   3. Choose a color, and click **OK** to close the **Select Color** dialog box.

   Abaqus/CAE adds the new color in the selected location.
4. To change one of the colors in the Auto-Color list, double-click the color and select a new color from the **Select Color** dialog box that appears.
5. To move a color within the Auto-Color list, highlight the color and click either **Move Up** or **Move Down** to move the color one step in the selected direction.
6. To delete a color from the Auto-Color list, highlight the color and click **Delete**.
7. Continue adding, changing, moving, and deleting colors until the Auto-Color list contains the colors you want in the desired order.
8. Click **OK**.

   Abaqus/CAE uses the revised Auto-Color list for any subsequent color coding.

Saving and restoring custom color mappings

Color mappings are session-specific settings that are not saved to the model database or output database by default. Abaqus/CAE provides two methods you can use to save your custom color coding definitions.

- You can save your color mapping definitions as session options. Session options can be saved to the model database, to an output database, or to an XML file. When you save color mappings using this process, Abaqus/CAE records only the color mappings for the item currently selected in the **Color Code** dialog box. For example, when color mappings for part instances are displayed, those color mapping definitions are the only ones written to the selected file.
- You can create a color macro or write the selected color mapping to an ASCII file. A color macro records all of the color mappings and your initial color selection, and an ASCII file records the current color mapping only. You can run a color macro as you would any other macro; see *Managing macros*. Color macros record all of the color mappings you select, rather than just the ones currently displayed in the **Color Code** dialog box.

This section describes the macro-based method for saving color mapping definitions. For more information about saving color mappings as a session option, see *Managing session objects and session options*.

Additional information

• *Managing macros*

Save a color macro

1. Click the  tool in the Color Code toolbar.
   Abaqus/CAE displays the **Color Code** dialog box.
2. Click **Advanced**, and select **Save Color Macro** from the list that appears.
   Abaqus/CAE displays the **Create Macro** dialog box, which indicates the location where the macro will be saved.
3. Enter a **Name** for this macro, and click **OK**.
   Abaqus/CAE saves the macro, making it available for you to run from the **Macro Manager** dialog box.

Read or write an ASCII file for a color mapping

1. Click the  tool in the Color Code toolbar.
   Abaqus/CAE displays the **Color Code** dialog box.
2. Click mouse button 3 with the cursor positioned over the **Color Mappings** table, then select one of the following choices from the list that appears:
   - Select **Write to File** to select a file name and to save the current color mapping.
   - Select **Read from File** to select a file name and to read the contents of a saved color mapping.

Using display groups to display subsets of your model

By default, Abaqus/CAE displays your entire model; however, you can choose to display subsets of your model by creating display groups.
These subsets can contain combinations of part instances, geometry (cells, faces, or edges), datum geometry (points, axes, planes, or coordinate systems), elements, nodes, and surfaces from the current model or output database. This chapter explains the concept of display groups and how you can manage them.

In this section:

- *Understanding display groups*
- *Managing display groups*

Understanding display groups

A display group is a collection of selected model components and can contain the entire model or combinations of part instances, geometry (cells, faces, or edges), datum geometry (points, axes, planes, or coordinate systems), elements, nodes, surfaces, constraints, and output database coordinate systems.
Display groups allow you to reduce clutter on your screen and focus on areas of interest within your model, to access "hidden" components in complex models, and to decrease the amount of time needed to refresh the display in the current viewport. For example, you can use display groups to show contact surfaces but suppress elements or to produce a contour plot showing elements in the interior of your model that would otherwise be obscured. You can plot, save, edit, copy, rename, and delete display groups.

In the Visualization module you can plot more than one display group in the same viewport, and you can customize the plot options for each display group independently.

In this section:

- *Understanding how to create display groups*
- *Understanding display group Boolean operations*

Understanding how to create display groups

A display group can contain combinations of model components: part instances, geometry (cells, faces, or edges), datum geometry (points, axes, planes, or coordinate systems), elements, nodes, surfaces, constraints, output database coordinate systems, or the entire model. In addition, you can create a display group by operating on previously saved display groups. However, while creating a display group, you can perform operations on only one type of model component at a time. Creating a display group containing selected components of more than one type is an incremental process. The model components that can be used to create a display group depend on the module in which you are working, as shown in *Table 1* and *Table 2*.

Table 1: Model components that can be used to create display groups in the Part- and Assembly-related modules and in the Visualization module.

Modules	Available Model Components
Part-related (Part, Property)	Geometry (cells, faces, or edges)
	Datum geometry (points, axes, planes, or coordinate systems)
	Elements
	Nodes
	Reference points
	Sets (geometry, element, or node)
	Display groups
Assembly-related (Assembly, Step, Interaction, Load, Mesh)	Part instances
	Geometry (cells, faces, or edges)
	Datum geometry (points, axes, planes, or coordinate systems)
	Assembly wires (connector wires)
	Elements
	Nodes
	Reference points
	Sets (geometry, element, or node)
	Surfaces
	Display groups
Visualization module	Part instances
	Elements
	Nodes
	Surfaces
	Display groups

Table 2: Model components that can be used to create display groups for output databases in the Visualization module.

Module	Available Model Components
Visualization	Part instances
	Elements

Module	Available Model Components
	Nodes
	Surfaces
	Display groups
	Coordinate systems
	Tie constraints
	Shell-to-solid coupling constraints
	Distributing coupling constraints
	Kinematic coupling constraints
	Rigid body constraints

To create a display group, you first select the particular items of interest. You can then perform a Boolean operation on your selection and the contents of the current viewport. This sequence can be repeated as necessary to create the desired group. In addition, you can create a new display group by editing (for example, performing additional Boolean operations on) the contents of a previously saved display group.

You can save either the selection in the dialog box or the contents of the current viewport as a display group. By default, a display group persists only for the duration of the session. If you want to retain a display group for use in subsequent sessions, save it to an XML file, to the model database, or to an output database; for more information, see *Managing session objects and session options*. You can access a display group only in modules of the same type as the one in which it was created (see *Table 1* and *Table 2*). For example, if you create and save a display group in the Part module, you will be able to access this display group only in the Part and Property modules.

A display group named `All` that contains all objects in the current part, assembly, or output database is created automatically by Abaqus/CAE. This display group appears in the display group manager for the current module and cannot be edited, copied, renamed, or deleted. It is not associated with a particular part, assembly, or output database. You can use this display group to quickly return to a plot of the entire part or model after performing Boolean operations.

Additional information

- *Creating or editing a display group*
- *Controlling the display of constraints in the Visualization module*

Understanding display group Boolean operations

To create or edit a display group, you can perform Boolean operations on selected model components and the contents of the current viewport. Abaqus/CAE offers the following Boolean operations in the Display Group toolset: **Replace**, **Add**, **Remove**, **Intersect**, and **Either**.

As an example of a simple Boolean operation, assume the current viewport shows the entire model. If you select a single element set and then apply the **Remove** operation, that element set is eliminated from the display in the current viewport.

For each Boolean operation you perform to create or edit a display group, you can select only one type of model component: part instances, geometry (cells, faces, or edges), datum geometry (points, axes, planes, or coordinate systems), elements, nodes, surfaces, previously saved display groups, or the entire model. For a given display group, Abaqus/CAE initially assumes that you want to include all nodes connected to all elements in the group. However, if you select particular nodes, all subsequent operations on that display group include only the nodes you have selected.

An explanation of each of the Boolean operations follows. In the icons below, the circle on the left represents the items in the current viewport; the circle on the right represents your selection. The shaded portion represents the resulting display group. All of these Boolean operations except the last one, **Replace All**, are available in the **Create Display Group** dialog box; the Display Group toolbar provides access to the **Replace**, **Remove**, **Either**, and **Replace All** Boolean operations, as well as **Undo** and **Redo** operations.

 Replace

Use the **Replace** operator to replace the current viewport contents with your selection.

 Add

Use the **Add** operator to add your selection to the current viewport contents.

 Remove

Use the **Remove** operator to remove your selection from the current viewport contents. If your selection includes one or more elements or surfaces, the nodes connected to those items are also removed, provided that you have not specifically operated to include those nodes in the display group.

 Intersect

Use the **Intersect** operator to display only those items common to your selection and the display in the current viewport.

 Either

Use the **Either** operator to display only model components that are either in your selection or in the current viewport but not in both.



Note: From the Display Group toolbar, click **Invert Display**  to hide all the model components that are displayed in the current display group or to display all the model components that are hidden. This

operation is equivalent to selecting all the objects in the **Create Display Group** or **Edit Display Group** dialog box and clicking the **Either** button.

Replace All

Use the **Replace All** operator to replace your display group and display all entities in the current viewport. This operator is available in the toolbar only.

When you select the  (**Replace Selected**) or  (**Remove Selected**) operators from the toolbar, Abaqus/CAE displays a list of model components in the prompt area. Select the type of model component that you want to replace or remove from the selected display group, then click one or more model components in the viewport to change the contents of the display group.



Note:

The **Replace Selected** and **Remove Selected** operators perform the same actions as the **Replace** and **Remove** operators in the display group dialog boxes. The toolbar operators have different names to indicate that you must select model components from the viewport before using these operators to change the current display.

Additional information

• *Creating or editing a display group*

Managing display groups

The display group managers allow you to create a display group and to plot, edit, copy, rename, or delete a previously saved display group.
The display group manager for the current module lists only the display groups that you have saved in modules of the same type during the current session. The **Part Display Group Manager** in the part-related modules lists the model and part to which each display group belongs; the **Assembly Display Group Manager** in the assembly-related modules lists the model to which each display group belongs.

In the Visualization module you can plot multiple display groups in the same viewport. The **ODB Display Group Manager** in the Visualization module allows you to lock and unlock plotted display groups to selectively customize plot options. In addition, you can synchronize all plotted display groups to use the same plot options.

In this section:

- *Creating or editing a display group*
- *Selection methods for creating or editing a display group*
- *Copying, renaming, and deleting a display group*
- *Plotting display groups*
- *Customizing display groups in the Visualization module*
- *Applying display groups to multiple viewports and models*

Creating or editing a display group

To create a display group, you first select the particular items of interest. You can then perform Boolean operations on your selection and the contents of the current viewport. When you create a display group, you can choose at any time to save either the contents of the current viewport or the items selected in the dialog box (if applicable). You must save a display group to do any of the following:

- Plot the display group later in the session.
- Apply the display group to a different model or to a different viewport.
- Edit, copy, or rename the display group.
- Plot multiple display groups in the same viewport in the Visualization module.

Saved display groups are available only during the current session and only when you are in modules of the same type as the module in which you created the display group.



Note:

Saved display groups are not updated to reflect changes made to the model, such as adding, deleting, or suppressing features or partitioning a part; such changes may invalidate display groups created with respect to the original model.

You can edit the combination of model components in a previously saved display group. Editing a display group is similar to creating a display group: in both cases you can select model components and then apply Boolean operations on your selection and the contents of the current viewport. When you select a display group to edit, the contents of the current viewport update to show the selected display group. You use the display in the current viewport to determine the model components in the group you are editing; you cannot list the model components. When you are finished editing, all viewports that use the display group will be updated.

1. Locate the options for creating or editing a display group.

   - To create a display group, select **Tools->Display Group->Create** from the main menu bar.

     Tip: You can also create a display group by clicking the  tool in the Display Group toolbar.

     The **Create Display Group** dialog box appears.

   - To edit a previously saved display group, select **Tools->Display Group->Edit** from the main menu bar. From the menu that appears, select the display group you want to edit.

     The **Edit Display Group** dialog box appears.

   Tip: You can also use the display group manager to create or edit a display group. From the main menu bar, select **Tools->Display Group->Manager** to display the manager for the current module. To create a display group, click **Create**. To edit a display group, select it from the list and double-click on it or click **Edit**.

2. From the **Item** list at the top left of the dialog box, select the type of model component to use for the display group (as available for the current module): **Cells**, **Faces**, **Edges**, **Datums**, **Assembly wires**, **Reference points**, **Elements**, **Nodes**, **Sets**, **Surfaces**, **Part instances**, **Display groups**, **Internal Sets**, **Internal Surfaces**, **Coordinate systems**, **Ties**, **Shell-to-solid couplings**, **Distributing couplings**, **Kinematic couplings**, or **Rigid bodies**. You can select **All** to specify all objects in the current part, assembly, or output database when you are creating a new display group.

Abaqus/CAE refreshes the **Method** list and the right side of the dialog box. If you select **All**, these fields are empty and no further item specification is necessary. If the item that you select has only one selection method, Abaqus/CAE immediately enters the required selection mode. For example, if you select **Cells**, Abaqus/CAE immediately enters the viewport selection mode.

3. Choose a selection method from the **Method** list; and/or select the specific items for the display group by picking from the viewport, selecting items from the list that appears on the right side of the dialog box, or entering data in the right side of the dialog box. See *Selection methods for creating or editing a display group*, for more detailed information.

   Tip: If your model contains many items of a particular type, you can use the filter to reduce the number of item names displayed on the right side of the **Create Display Group** dialog box. Click the **Tip** button next to the **Filter** field to see examples of valid filtering syntax.

   Certain items can be highlighted in the viewport to verify your selection. Toggle **Highlight items in viewport** if available.

   Toggle **Show common list between linked viewports** to display items common to all linked viewports in the **Method** list. For **Part instances**, viewports containing model databases or output databases are considered; for other model components, only viewports containing output databases are considered.

4. From the icons in the bottom portion of the dialog box, click the desired Boolean operation (see *Understanding display group Boolean operations*, for more information).

   Abaqus/CAE carries out the selected Boolean operation on the model components you have selected in the dialog box and the contents of the current viewport.

5. Repeat steps beginning with Step 2 as needed to produce the desired display group.
6. If you are creating a new display group, you can choose to save it at any time.
   a. To save the contents of the current viewport as a display group, click **Save As**; then enter a name for the display group in the dialog box that appears, and click **OK**.

      Abaqus/CAE saves the contents of the current viewport (regardless of what is selected in the dialog box) as a display group.

   b. If model components are selected in the **Create Display Group** dialog box, you can save the selected components as a display group. Click **Save Selection As**; then enter a name for the display group in the dialog box that appears, and click **OK**.

      Abaqus/CAE saves only the components selected in the dialog box (regardless of what is displayed in the current viewport) as a display group.

7. Click **Dismiss** to close the **Create Display Group** dialog box or **OK** to close the **Edit Display Group** dialog box.

Additional information

- *Understanding how to create display groups*
- *Understanding display group Boolean operations*
- *Selection methods for creating or editing a display group*
- *Understanding result value averaging*
- *Controlling edge visibility*
- *Choosing background colors*
- *Controlling the display of constraints in the Visualization module*

Selection methods for creating or editing a display group

You can use the following selection methods to specify the items that will be contained in a display group:

- To specify geometry (cells, faces, or edges), assembly wires, or reference points by picking them directly from the viewport, select them from the **Item** list. (For datums, elements, and nodes, you must also select **Pick from viewport** from the **Method** list.) Abaqus/CAE automatically enters the pick mode, and **Select *items* for the display group** appears in the prompt area. See *Selecting objects within the viewport* for more information on picking items in the viewport.

  Click **Done** in the prompt area when you have finished picking items in the viewport.

  Click **Edit selection**, **Add selection**, or **Delete selection** in the dialog box to further modify your viewport selections.
- To specify elements or nodes by number, select **Element labels** or **Node labels** from the **Method** list. Select the name of the part instance to which the nodes or elements belong from the list in the **Part instance** field on the right side of the **Create Display Group** dialog box. Type into the **Labels** field a list of element or node numbers separated by commas or a range of numbers optionally followed by an increment; for example, 1:10 or 1:10:2.
- To specify elements in the Visualization module by type, select **Element type** from the **Method** list. A list of the available element types in your model appears on the right side of the **Create Display Group** dialog box. Select one or more element types from this list (see *Selecting multiple items from lists and tables* for more information).
- To specify elements in the Visualization module by material or section, select **Material assignment** or **Section assignment** from the **Method** list. A list of the available materials or sections in your model appears on the right side of the **Create Display Group** dialog box. Select one or more materials or sections from this list (see *Selecting multiple items from lists and tables* for more information).
- To specify elements in the Visualization module by layups or plies, select **Composite Layups** or **Composite Plies** from the **Method** list. A list of the available composite layups or plies in your model appears on the right side of the **Create Display Group** dialog box. Select one or more layups or plies from this list (see *Selecting multiple items from lists and tables* for more information).
- To specify geometry, element, or node sets in part- and assembly-related modules or surfaces in assembly-related modules, select **Set names** or **Surface names** from the **Method** list. A list of the available set or surface names in your model appears on the right side of the **Create Display Group** dialog box. Select one or more set names from this list (see *Selecting multiple items from lists and tables* for more information).
- To specify internal (created by Abaqus/CAE) sets in part- and assembly-related modules or internal surfaces in assembly-related modules, select **Internal sets** or **Internal surfaces** from the **Item** list. A list of the available set or surface names in your model appears on the right side of the **Create Display Group** dialog box. Select one or more set names from this list (see *Selecting multiple items from lists and tables* for more information).
- To specify element, node, surface, or internal (created by Abaqus/CAE) sets in the Visualization module, select **Element sets**, **Node sets**, **Surface sets**, or **Internal sets** from the **Method** list. A list of the available set names in your model appears on the right side of the **Create Display Group** dialog box. Select one or more set names from this list (see *Selecting multiple items from lists and tables* for more information).
- To select all items of the specified type in your model, select **All elements**, **All nodes**, or **All surfaces** from the **Method** list. No further item specification is necessary.
- If you selected **Part instances** or **Display groups** from the **Item** list, a list of the available part instances or display groups in your model appears on the right side of the **Create Display Group** dialog box. Select one or more part instance or display group names from this list (see *Selecting multiple items from lists and tables* for more information).

  Note:

  In assembly-related modules you can also display a subset of the part instances in your model using the visibility options in the **Instance** tabbed page of the **Assembly Display Options** dialog box (see *Controlling*

instance visibility). Part instances that have been toggled off in the **Assembly Display Options** dialog box cannot be made visible using display groups.

- To specify elements, nodes, or surfaces containing results within a given range of values in the Visualization module, select **Result value**.

  The output variable to be considered is displayed at the top right side of the dialog box. To select a new result variable, click **Field Output**; see *Selecting the field output to display* for more information on the **Field Output** dialog box. Choose from the filtering methods in the **Type** field.

  - If you select  (inside) or  (outside), enter values for the upper and lower bounds of the range of results in the **Min value** and **Max value** fields, respectively.
  - If you select , enter a value above which the results should fall in the **Min value** field.
  - If you select , enter a value below which the results should fall in the **Max value** field.

  Every element (or node or surface) in the model is used in the filtering process, regardless of the current active display group in the model. If you choose a range of values for which no elements (or nodes or surfaces) exist, you will create an empty display group.

  You can achieve better performance in time history animations using the *status field output variable* rather than the result-based display group functionality. The status field output variable allows you to remove elements that meet a specified result-based failure criteria from your model plots; see *Selecting the status field output variable* for more information.

  Note:

  The bounds for filtering based on element, nodal, or surface output variables are always based on the values of a variable at the nodes. Therefore, element- and surface-based output quantities are extrapolated and averaged at the nodes before comparing them against the user-defined bounds. The averaging settings in the **Result Options** dialog box determine how element- and surface-based variables are calculated at the nodes. For example, consider a case where elements are filtered based on Mises stress using the default averaging threshold of 75%. After extrapolation to the nodes, the values are averaged according to this threshold. This conditional averaging may result in several distinct values of Mises stress at the node based on contributions from the various elements to which the node belongs. Any element whose Mises stress contribution falls within the user-defined bounds is included in the display group.

- To specify coordinate systems in the Visualization module, select **Coordinate systems** from the **Item** list, then select one of the following:

  - To specify coordinate systems that are part of the output database, select **Odb systems** from the **Method** list. A list of the output database coordinate systems appears in the right side of the **Create Display Group** dialog box.
  - To specify coordinate systems that have been created during your session, select **User systems** from the **Method** list. A list of the user-defined coordinate systems appears in the right side of the **Create Display Group** dialog box.
  - To specify coordinate systems from the output database and the session, select **All** from the **Method** list. A list of the output database coordinate systems and user-defined coordinate systems appears in the right side of the **Create Display Group** dialog box.

  Select one or more coordinate systems from the list that appears (see *Selecting multiple items from lists and tables* for more information).

- To specify analysis constraints in the Visualization module, select one of the following constraint types in the **Item** list: **Ties**, **Shell-to-solid couplings**, **Distributing couplings**, **Kinematic couplings**, or **Rigid bodies**.

Additional information

- *Creating or editing a display group*

Copying, renaming, and deleting a display group

To copy, rename, or delete a display group, use one of the following:

- The **Copy**, **Rename**, and **Delete** items listed under the **Tools->Display Group** menu on the main menu bar.

  The **Copy**, **Rename**, and **Delete** items contain submenus listing all the display groups you have saved in modules of the same type as the current module during the session.
- The display group manager dialog box for the current module. The display group manager dialog boxes contain functions identical to those listed under the **Tools->Display Group** menu on the main menu bar but with a convenient browser that lists all the display groups for the current module and session. To display the display group manager dialog box for the current module, select **Tools->Display Group->Manager** from the main menu bar.



Note:

If you try to delete a display group that is shown in multiple viewports, Abaqus/CAE warns that the display groups in those viewports will be reset to the entire model.

Additional information

• *Managing display groups*

Plotting display groups

You can plot a display group for the model in the current viewport. In the Visualization module you can plot more than one display group in the same viewport.
Abaqus/CAE plots only the components in each display group that are valid for the current model. For example, if a display group references an element set called `Fixture`, the model in the current viewport should also contain an element set called `Fixture`. Abaqus/CAE ignores model components that are not valid.

Additional information

• *Managing display groups*

Plot a single display group in any module

1. From the main menu bar, select **Tools->Display Group->Plot**.
2. From the menu that appears, select the display group that you want to plot.

   Tip: You can also use the display group manager for the current module to plot a display group. From the main menu bar, select **Tools->Display Group->Manager**. Select the display group you want to plot, and click **Plot** from the buttons on the right side of the manager.

   The model plot in the current viewport changes to show only the selected display group.

Plot one or more display groups in the Visualization module

1. From the main menu bar, select **Tools->Display Group->Manager**.

   The **ODB Display Group Manager** appears with a list at the top of all display groups saved in the current session.
2. Select one or more display groups from the list (see *Selecting multiple items from lists and tables*, for more information), and click **Plot** from the buttons on the right side of the manager.

   The model plot in the current viewport changes to show only the selected display groups, and the plotted display groups appear in the list of display group instances at the bottom of the **ODB Display Group Manager**.
3. To add one or more display groups to the plot in the current viewport, select them from the list of saved display groups at the top of the **ODB Display Group Manager**, and click **Add** from the buttons on the right side of the manager.

   The selected display groups are added to the model plot in the current viewport and to the list of display group instances at the bottom of the **ODB Display Group Manager**.
4. To remove a display group from the plot in the current viewport, select the instance from the list at the bottom of the **ODB Display Group Manager**, and click **Remove** from the buttons at the bottom of the manager.

   The selected display group is removed from the model plot in the current viewport and from the list of display group instances.
5. You can customize plot state–dependent and independent options separately for each display group in the current viewport. For more information, see *Customizing display groups in the Visualization module*.

Customizing display groups in the Visualization module

The **ODB Display Group Manager** in the Visualization module allows you to lock and unlock plotted display groups to selectively customize plot options. In addition, you can synchronize all plotted display groups to use the same plot options. For example, you can display a particular group of elements in the shaded render style with translucency and display the remainder of the model in the wireframe render style. *Figure 1* shows another example of two display groups plotted in the same viewport using different plot states and plot options.

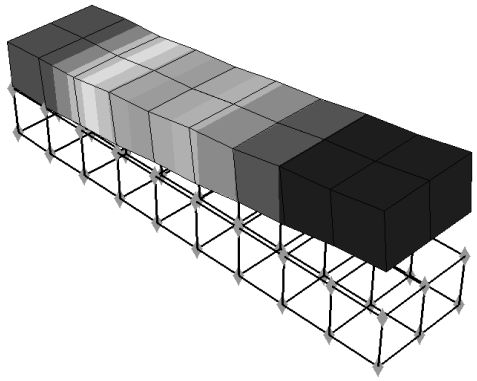


Figure 1: Multiple customized display groups in one viewport.

When you plot a display group in the Visualization module, the display group appears (along with its current plot states) in the list of display group instances at the bottom of the **ODB Display Group Manager**. Plot state–dependent and independent options that you specify apply to all unlocked display group instances. You can lock a display group instance that you do not want to be affected by plot options.

Changing the plot state will not affect locked display group instances in the viewport. However, view manipulations and changes to the color coding will affect all display group instances, both locked and unlocked. In addition, the contents of a locked display group can be edited.

Display groups are plotted in the viewport in the order in which they appear in the list of display group instances; the first instance in the list is the topmost instance in the viewport. Therefore, if two display group instances contain the same components and those components overlap in the viewport, the plot options for the first instance in the list take precedence. You can rearrange the order of the instances in the list to control which plot options will be displayed.

1. Follow the procedure described in *Plotting display groups*, to plot multiple display groups in the same viewport.
2. In the list of display group instances in the **ODB Display Group Manager**, click in the **Lock** column next to the display group name to lock the instances that you do not want to customize.

   Locked display groups are indicated by a check mark in the **Lock** column.
3. Customize the plot options for the unlocked display group instance or instances as desired. (See *Customizing plot display*," for more information.)
4. When you have finished customizing the plot options for a display group instance, click in the **Lock** column next to the display group name to prevent your settings from being modified.
5. Repeat Steps 3 and 4 as needed until you obtain the desired display in the viewport.
6. If you want all display group instances to use the same plot state and plot options, select the instance that reflects the desired plot state and plot options, and click **Sync Options** from the buttons at the bottom of the manager.

   All other unlocked display group instances will be synchronized to the selected display group's plot state and plot options.

7. To rearrange the order in which the display groups are plotted in the viewport, select an instance from the list at the bottom of the **ODB Display Group Manager** and click **Move Up** to move the instance up in the list or **Move Down** to move the instance down in the list.

   The display in the current viewport is updated to reflect your change.

Additional information

• *Managing display groups*

Applying display groups to multiple viewports and models

You can use a single display group to view the same subset of different models, and you can view a display group in multiple viewports.

A single display group can apply to different models as long as the display group is valid for each model. For example, *Figure 1* shows a single display group applied to two different models shown in two different viewports.

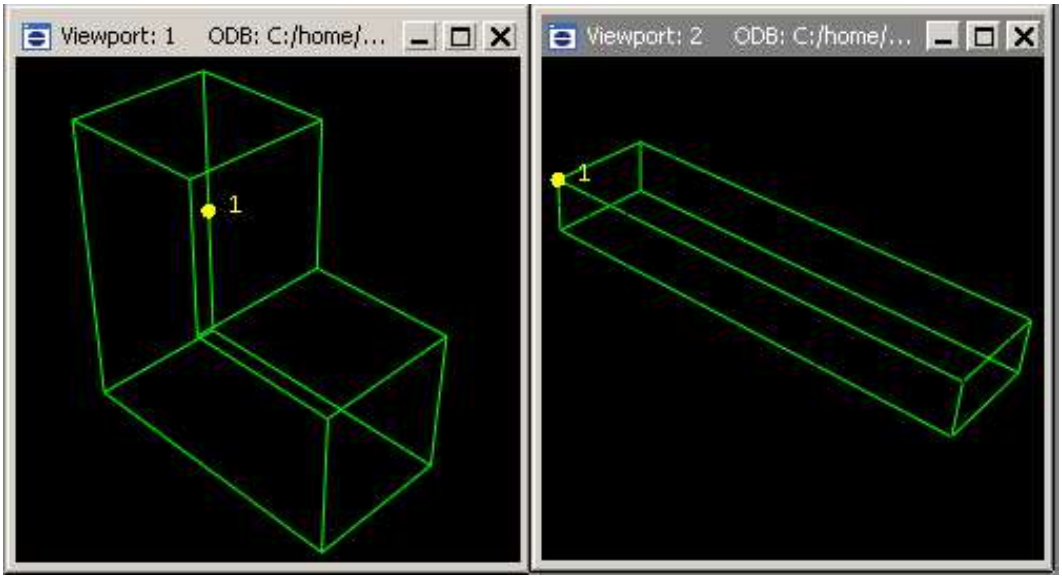


Figure 1: A display group applied to two different models.

Additional information

• *Managing display groups*

Plot a previously saved display group in any viewport

1. Click on the viewport border to make the viewport current.
2. From the main menu bar, select **Tools->Display Group->Plot**. Select the display group to plot from the menu that appears.

   Abaqus plots the display group in the current viewport. Abaqus plots only the components in the display group that are valid for the current model.

Edit a previously saved display group in any viewport where it is displayed

1. Click on the viewport border to make the viewport current.
2. From the main menu bar, select **Tools->Display Group->Edit**. Select the display group to edit from the menu that appears.

   The **Edit Display Group** dialog box appears.
3. Select the desired model components, and apply the desired Boolean operations.

   During the editing process, the results of any Boolean operations appear only in the current viewport.
4. Click **OK** to close the **Edit Display Group** dialog box.

   Your editing changes are applied to all viewports that reference the selected display group. If the modified display group becomes invalid for one of the models (for example, by the inclusion of a node not contained in that model), Abaqus warns that part of the display group is now invalid.

Overlaying multiple plots

This chapter explains the concept of overlaying plots and how you can create and manage such plots.
By default, Abaqus/CAE displays only one plot at a time in the current viewport. A plot may display multiple plot states, such as the contours and material orientations of the same model. However, a single plot cannot display data from more than one output database nor can it display both the model and an *X*–*Y* plot in the same viewport. If you want to display data in this fashion, you must overlay individual plots in the same viewport.

In this section:

- *Understanding how to overlay plots*
- *Producing and modifying overlay plots*

Understanding how to overlay plots

You can create a display that contains multiple plots in one viewport. For example, you may want to do any of the following:

- combine a contour plot and an *X*–*Y* plot
- compare the deformed plot shapes from two different output database files in the same viewport
- combine a time history animation with an animated *X*–*Y* plot displaying the change in several variables in the model over time

Overlay plots are useful, for example, for displaying data from both output databases in a co-simulation in the same viewport.

Overlay plots are composed of layers; each layer contains one plot, and the layers are stacked on top of each other to create the combined plot. *Figure 1* shows a plot containing the deformed shape plots at four different increments of an analysis as well as an *X*–*Y* plot of the strain energy in the model versus time.

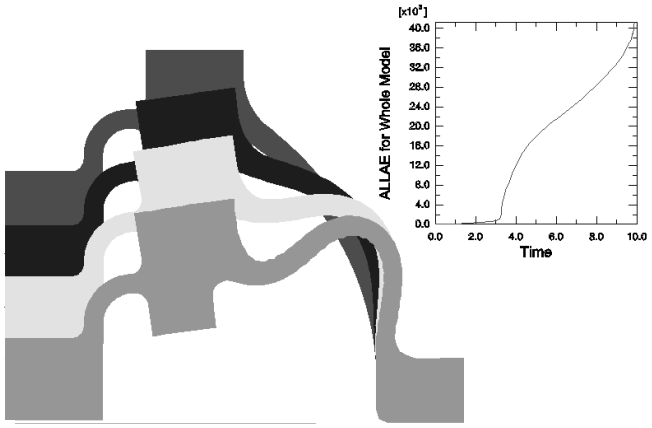


Figure 1: Overlay plot.

By default, the viewport does not contain any layers; only one plot at a time is displayed. To overlay multiple plots, you create a layer for each individual plot as you interact with Abaqus/CAE. You then choose the layers to display in the current viewport. You can create as many layers as you would like; any number of layers can be displayed in the same viewport. In addition, you can open multiple output databases and automatically display the combined contents in an overlay plot in a single viewport.

Use the **Overlay Plot Layer Manager** to create, display, position, and delete layers. To access the manager, select **View->Overlay Plot** from the main menu bar or click the **Overlay Plot Layer Manager** tool in the toolbox.

When you create a layer, it contains everything that is visible in the current viewport. You can change the contents of a layer, manipulate its view, reorder the layer with respect to the other layers in the overlay plot, and change the various display options that are applied to the contents. By default, layers are plotted directly on top of one another. Sometimes lines that appear directly on top of one another can create undesirable visual effects. You can offset the layers with respect to each other to avoid such display anomalies.

The settings in the **Overlay Plot Layer Manager** are applied to the contents of the current viewport only when you click **Plot Overlay**. Abaqus/CAE then enters the overlay plot state; when you click **Plot Single** in the **Overlay Plot Layer Manager**, the overlay plot disappears from the viewport and the display reverts back to the previous plot state.

You can also click the tool in the toolbox to switch between the single plot and overlay plot state at any time.

When you are in the overlay plot state, Abaqus/CAE displays your plots relative to an overlay plot coordinate system. As you create a layer, Abaqus/CAE assigns the view in which the layer was created to the overlay plot front (1–2) view. You can modify what is displayed in the overlay plot front view by manipulating the view for each individual layer, as described in *Manipulating the view for an overlay plot*. User-specified views defined in the overlay plot state are relative to the overlay plot coordinate system.

Columns in the **Overlay Plot Layer Manager** display the following information about each layer:

Visible

A check mark in this column indicates that the layer is visible in the viewport when you are in the overlay plot state.

Current

A check mark in this column indicates that the layer is current. Only one layer can be current at a time, although each layer can contain multiple plot states (for more information, see *Displaying multiple plot states*). Plot options are applied to only the current layer when you are in the overlay plot state; you can choose whether view manipulation options are applied to all existing layers or to only the current layer. The current layer is not necessarily the foremost layer in the viewport.

Name

The name of the layer.

Object

The name of the object contained in the layer; for example, an output database or an *X*–*Y* plot.

Producing and modifying overlay plots

This section explains how to overlay multiple plots by creating layers and plotting them in the same viewport and how to modify an overlay plot once you have created it.
Each layer in an overlay plot is completely independent. You can change the output database, the plot state (or plot states), reorder the layers with respect to each other, manipulate the view for each individual layer, and change the various display options that are applied to the contents.

In this section:

- *Producing an overlay plot*
- *Reordering the layers in an overlay plot*
- *Manipulating the view for an overlay plot*
- *Editing the layers in an overlay plot*

Producing an overlay plot

An overlay plot is a display that contains multiple plots in one viewport.
Overlay plots are composed of layers; you can open multiple output databases and create the layers automatically, or you can use the **Overlay Plot Layer Manager** to create layers manually.

You also use the **Overlay Plot Layer Manager** to configure your overlay plots.

Tip: You can also plot any combination of deformed and undeformed models shapes with or without contours, symbols, or materials orientations for a single output database by using the **Allow Multiple Plot States** tool. Since all the plot states are created on a single layer, the **Allow Multiple Plot States** tool is limited to displaying data from a single output database, step, and frame. For more information, see *Displaying multiple plot states*.

1. Determine the method to use for producing the overlay plot:
   - Create layers automatically by opening multiple output databases.
   - Create layers manually.

2. To create layers automatically, do the following:
   a. Select **File->Open** from the main menu bar.
   b. From the **Open Database** dialog box that appears, select **Output Database (*.odb*)** as the **File Filter**.
   c. Select the output databases to open, toggle on **Append to layers**, and click **OK**. For more details, see *Opening a model database or an output database*.

      An overlay plot containing the combined contents of the output databases is automatically created in a viewport, and each output database is assigned to a separate layer. You can select **View->Overlay Plot** from the main menu bar to display the **Overlay Plot Layer Manager** and view the layers.

3. To create layers manually, do the following:
   a. Create the first plot that you want to include in the display.
   b. Open the **Overlay Plot Layer Manager** by selecting **View->Overlay Plot** from the main menu bar.

      Tip: You can also open the manager by clicking the **Overlay Plot Layer Manager** tool in the toolbox.

   c. Click **Create** to create a layer that contains the plot shown in the current viewport.

      The **Create Viewport Layer** dialog box appears.

   d. Enter a name for the layer, and click **OK**.

      The new layer appears in the **Overlay Plot Layer Manager**, with check marks in both the **Visible** and **Current** columns.

   e. Create a new plot in the current viewport.
   f. Repeat Step c through Step e to create as many layers as you would like to include in the plot. In the **Create Viewport Layer** dialog box for all layers after the first one, you can choose to copy the view from an existing layer by toggling on **Copy view from** and selecting a layer from the list.

      The new layers appear in the **Overlay Plot Layer Manager** as you create them; the most recently created layer will have a check mark in the **Current** column.

g. Check that the layers you want to include in the plot have a check mark in the **Visible** column, and click **Plot Overlay**.

   The layers appear stacked on top of each other in the current viewport in the order in which they are listed in the **Overlay Plot Layer Manager**.

4. From the **Overlay Plot Layer Manager**, you can delete a layer by clicking in the **Name** or **Object** column to select the layer and clicking **Delete**.
5. With the overlay plot displayed in the viewport, you can click **Plot Single** in the **Overlay Plot Layer Manager** to exit the overlay plot state.

   The overlay plot disappears from the viewport, and the display returns to the previous plot state; but the **Overlay Plot Layer Manager** remains open. Click **Dismiss** to close the **Overlay Plot Layer Manager**.



Tip: You can also click the  tool in the toolbox to switch between the single plot and overlay plot state at any time.

Additional information

- *Understanding how to overlay plots*
- *Producing and modifying overlay plots*

Reordering the layers in an overlay plot

Layers are plotted in the viewport in the order in which they are listed in the **Overlay Plot Layer Manager**. By default, the layers are plotted directly on top of one another, which can sometimes result in undesirable visual effects where the plots appear to overlap.

You can rearrange the order in which the layers are plotted, and you can apply a small offset to all the layers in the screen *Z*-direction (perpendicular to the plane of the screen) to separate them in the display and eliminate any undesirable visual effects. If you apply a positive offset, the layers are drawn in the viewport in the order shown in the **Overlay Plot Layer Manager** (the last layer in the manager is the topmost layer in the viewport). If you apply a negative offset, the drawing order is reversed. *Figure 1* shows an example of an overlay plot with different offset values applied.

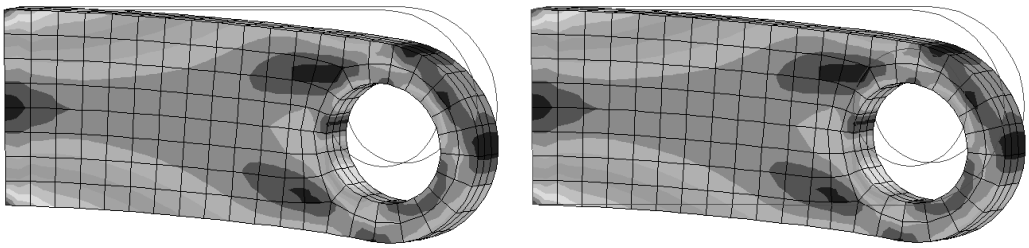


Figure 1: Overlay of undeformed and contour plots: positive offset on left, negative offset on right.

1. Create an overlay plot using the procedure described in *Producing an overlay plot*.
2. Reorder the layers using one of the following methods:
   - Select the layer you wish to reorder by clicking in the **Name** or **Object** column in the **Overlay Plot Layer Manager**. Click **Move Up** to move the layer up in the manager; click **Move Down** to move the layer down in the manager.
   - Drag the **Layer offset** slider to a positive or negative value to shift the layers in the positive or negative screen *Z*-direction. You may have to experiment with the offset values to achieve your desired display.

   Abaqus/CAE applies your settings to the overlay plot in the current viewport.
3. With the overlay plot displayed in the viewport, you can click **Plot Single** in the **Overlay Plot Layer Manager** to exit the overlay plot state.

   The overlay plot disappears from the viewport, and the display returns to the previous plot state; but the **Overlay Plot Layer Manager** remains open. Click **Dismiss** to close the **Overlay Plot Layer Manager**.



Tip: You can also click the  tool in the toolbox to switch between the single plot and overlay plot state at any time.

Additional information

- *Understanding how to overlay plots*
- *Producing and modifying overlay plots*

Manipulating the view for an overlay plot

While you are in the overlay plot state, you can choose whether view manipulations are applied to all existing layers or to only the current layer.

1. Create an overlay plot using the procedure described in *Producing an overlay plot*.
2. In the **Overlay Plot Layer Manager**, toggle **All** or **Current** next to **View manipulation layer** to specify whether you want the view manipulations that you apply to affect all the layers or only the current layer.
3. Use the view manipulation tools to position, orient, and magnify the objects in the viewport. (See *Manipulating the view and controlling perspective*," for information on the view manipulation tools.)

   The view manipulations will be applied either to all the visible layers or to only the current layer, depending on your selection in Step 2. The 3D compass can be used only to manipulate all of the visible layers; it is not available when you are manipulating only the current layer.
4. With the overlay plot displayed in the viewport, you can click **Plot Single** in the **Overlay Plot Layer Manager** to exit the overlay plot state.

   The overlay plot disappears from the viewport, and the display returns to the previous plot state; but the **Overlay Plot Layer Manager** remains open. Click **Dismiss** to close the **Overlay Plot Layer Manager**.



Tip: You can also click the  tool in the toolbox to switch between the single plot and overlay plot state at any time.

Additional information

- *Understanding how to overlay plots*
- *Producing and modifying overlay plots*

Editing the layers in an overlay plot

You can change the model, plot state, plot options, animation controls, or field output variable options for each layer in an overlay plot.

1. Create an overlay plot using the procedure described in *Producing an overlay plot*.
2. Click in the **Current** column next to the layer that you want to modify.
3. To change the model, open a new output database. (For more information, see *Opening a model database or an output database*.)
4. To change the plot type, select a new plot type from the **Plot** menu or from the plot tools in the toolbox.
5. To change the plot state-independent or plot state-dependent customization options, select new options from any of the following dialog boxes:

   - **Viewport Annotation Options**
   - **ODB Display Options**
   - **Result Options**
   - **Common Plot Options**
   - **Superimpose Plot Options**
   - **Contour Plot Options**
   - **Symbol Plot Options**
   - **Material Orientation Plot Options**

   For more information, see *Using viewport annotation options*, *Selecting result options*, or *Customizing plot display*."

   Click **OK** or **Apply** in each dialog box to implement your changes in the current layer of the overlay plot.

6. You can synchronize the visible layers by clicking any of the  icons under **Layer Options** in the **Overlay Plot Layer Manager**: **View manipulation layer**, **Plot state layer**, or **Plot options layer**.
7. Change any of the animation options desired. For more information, see *Object-based animation*, and *Controlling animation playback*.
8. Change the field output variable options desired. For more information, see *Selecting field output variables*.
9. You can synchronize the field output variable options of the visible layers by clicking the  icon next to **Field Output layer**.
10. Repeat Step 2 through Step 8 for each layer that you wish to modify.
11. With the overlay plot displayed in the viewport, you can click **Plot Single** in the **Overlay Plot Layer Manager** to exit the overlay plot state.

    The overlay plot disappears from the viewport, and the display returns to the previous plot state; but the **Overlay Plot Layer Manager** remains open. Click **Dismiss** to close the **Overlay Plot Layer Manager**.



Tip: You can also click the  tool in the toolbox to switch between the single plot and overlay plot state at any time.

Additional information

- *Understanding how to overlay plots*
- *Producing and modifying overlay plots*

Cutting through a model

View cuts allow you to cut through a model so that you can visualize the interior or selected sections of the model and, for results data in the Visualization module, display its resultant forces and moments.

In this section:

- *Understanding view cuts*
- *Managing view cuts*

Understanding view cuts

View cuts allow you to cut planar or deformable sections through a model to see the interior of the model. You can define and use view cuts during both modeling and postprocessing activities, though some aspects of view cut functionality are available only for one of these activities. This section describes the view cut functionality; unless otherwise noted, the view cut functionality described is available for both modeling and postprocessing.

Figure 1 shows how a planar view cut can be used to cut through a contour plot of a gearbox model in the Visualization module.

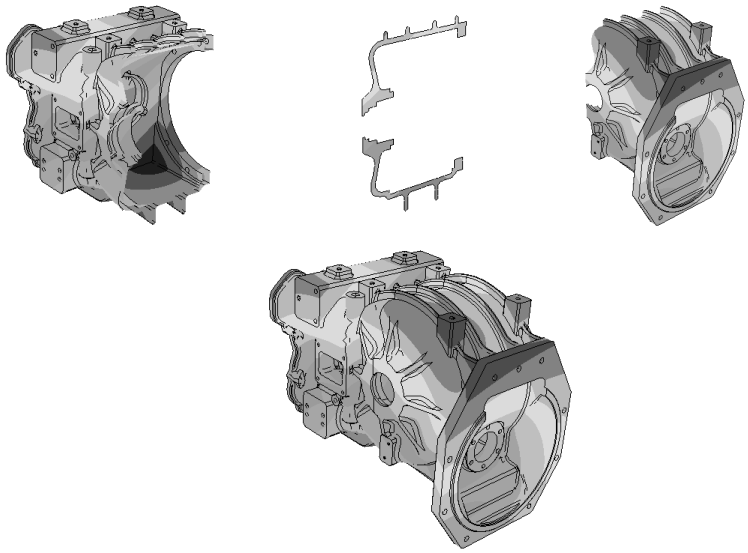


Figure 1: Planar view cut through a contour plot of a gearbox model. Top, left to right: the model below the cut, on the cut, and above the cut; bottom: the entire model.

View cut shapes

You can create view cuts based on a plane. In the Visualization module you can also create view cuts based on the following shapes:

- a cylinder,
- a sphere, or
- an isosurface corresponding to a constant value of a field variable or model attribute.

The types of shapes for view cut creation are illustrated in *Figure 2*.

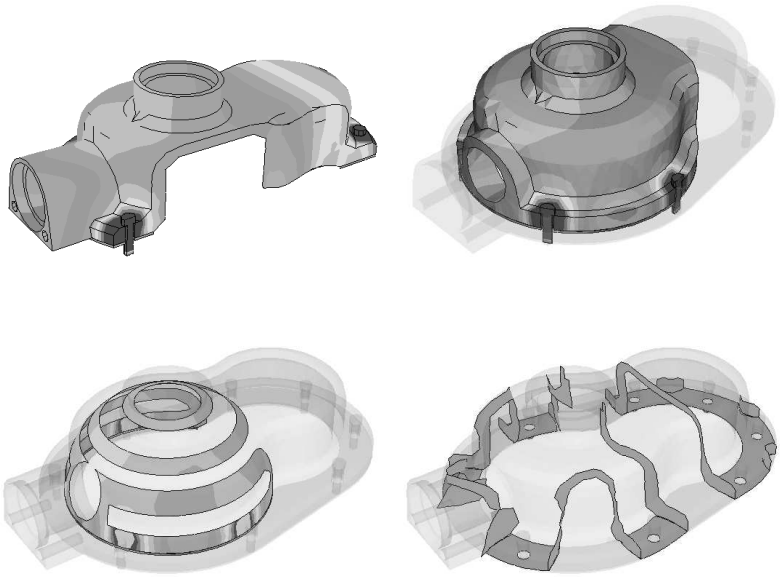


Figure 2: View cuts based on planar, cylindrical, spherical, and isosurface shapes.

For isosurface cuts the result values are computed as described in *Understanding how contour values are computed*, for line- and banded-type contours. By default, Abaqus/CAE applies an averaging threshold of 100% for isosurface cuts to insure the continuous display of results at the cut location. You can apply an averaging threshold less than 100% by toggling off **Override primary variable averaging** when you create the isosurface cut. Abaqus/CAE then applies the averaging threshold specified in the **Averaging** options in the **Result Options** dialog box; for more information, see *Controlling result averaging*. Cuts along the *X*-, *Y*-, and *Z*-planes are created by default.

Note:

Isosurface view cuts offer slightly different functionality and behavior than isosurface-type contour plots. An isosurface view cut always reflects the results of the primary variable that was active when the view cut was created; you cannot change the variable for which an isosurface view cut displays results. By contrast, isosurface-type contour plots always display data from the currently selected primary variable for your session. Because each isosurface view cut is tied to a single variable, you can investigate the locations where two isosurface view cuts intersect if you display multiple isosurface view cuts that are based on different output variables.

Displaying the cut section

To display the cut section of your model, you activate a cut and choose whether to display the model on the cut, above the cut, and/or below the cut, as illustrated in *Figure 1*. The cut itself is never visible. For planar cuts the portion of the model below the cut is defined as that part located on the negative side of the plane (relative to the orientation of the normal to the plane). For cylindrical and spherical cuts the portion of the model below the cut is defined as that part located at radii less than the radius of the cut shape. For an isosurface cut the portion of the model below the cut is defined as that part located at isovalues less than the specified isovalue. By default, Abaqus/CAE displays the model on and below the cut. In the Visualization module, you can display the model above the cut and the model below the cut at the same time; in all other modules, only one of these display options is available at a time.

Plot options

You can select different plot options for the portions of the model below the cut, on the cut, and above the cut; for example, in *Figure 2* some portions of the model are displayed with translucency activated, while others are displayed with no translucency.

Displaying multiple view cuts (Visualization module only)

Only one cut can be active (used to display the model in the current viewport) at a time in modules other than the Visualization module; in the Visualization module, however, you can display multiple view cuts at once. In addition, in the Visualization module you can activate view cuts on undeformed, deformed, contour (texture-mapped only), symbol, or material orientation plots. For symbol and material orientation plots, Abaqus/CAE displays symbols and material orientation triads at all integration points for each element included in the view, even if the element is partially cut.

Animation of view cuts (results data only)

Plots with an active view cut can be animated for output database data in the Visualization module; the view cut will be updated for each animation frame. View cuts cannot be activated on tessellated contour plots; swept, shrunk, or extruded plots; or highly refined plots (medium, fine, or extra fine).

Results caching

By default, Abaqus/CAE caches the result values used to generate images of a cut model in memory to improve performance. However, you can disable results caching for cut fields to decrease memory usage if necessary; see *Controlling results caching*, for more information.

Following the deformation

For plots of the deformed model, you can choose to have the cut follow the deformation; i.e., the cut surface will be calculated relative to a reference frame, and the cut deformation will match the deformation of the model.

Repositioning view cuts

You can reposition cuts on your model: planar cuts can be rotated or translated, while the other cut shapes can only be translated.

View Cut Manager

You can select a cut (either active or inactive) in the **View Cut Manager** to position, edit, copy, rename, or delete it. Selecting a cut in the manager is not the same as activating the cut. A cut can be both selected and active, unselected and active, or selected and inactive; and you can display or hide a free body cut for the currently displayed view cut if it is active. You can specify free body options that are common to all view cuts or that are specific to a particular view cut. *Figure 3* shows an example of the information displayed in the view cut manager in the Visualization module; in other modules, the free body display and cut-specific options are not included in this dialog box.

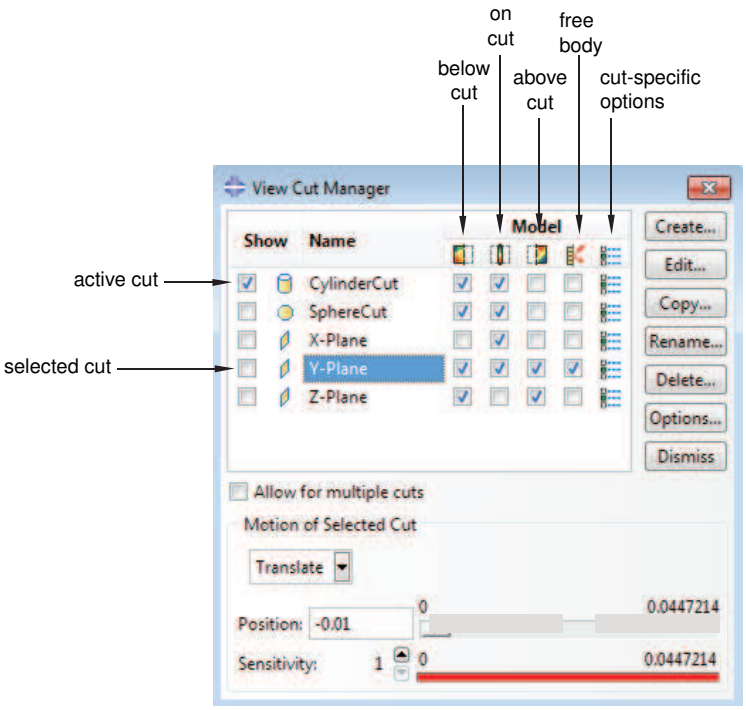


Figure 3: The View Cut Manager dialog box in the Visualization module.

Resultant forces and moments on view cuts (results data only)

For view cuts of output database data in the Visualization module you can also display the resultant forces and moments, and you can compute these values based on a display group, an element set, or the entire model. You can display the resultant force and moment on a view cut only for solid geometry, composite solid sections, shell sections, or beam sections. For resultant forces and moments to be displayed, the output database must include element force nodal output (NFORC) for composite solid sections and section force (SF) and section moment (SM) output for shell sections and beam sections. Abaqus/CAE does not support display of resultant forces and moments on view cuts for axisymmetric models. You can display the resultant force and moment on a view cut for any of the displayed view cuts in your session. Abaqus/CAE updates the resultant force and moment vectors and summation point as you reposition the view cut or animate the model.

You may observe different values for the resultant forces and moments if you compare the values in the deformed and undeformed shape plots. These differences can occur because view cut–based free body force computations integrate the stress field over the area, and, if you toggle the plot state, the elements that are part of the view cut can change or the elemental deformations can change the area of the cross section.

By default, Abaqus/CAE displays vectors on the view cut only, showing the resultant force and moment across the view cut. However, you can also display a series of vectors that show resultant force and moment data at regular intervals in your model. This series of resultant force and moment vectors can run through the entirety of your model or within a user-specified region.

XFEM-based view cut components (results data only)

Abaqus/CAE automatically creates and displays a view cut when you open an output database containing a crack calculated by the extended finite element method (XFEM). The view cut displays the model at the isosurface where the value of the signed distance function is zero, which corresponds with the surface of the XFEM crack. Boundary conditions are not displayed while an XFEM crack view cut is active.

The tools in the **View Cut Manager** operate as follows for XFEM cracks:

- The below cut  check box displays the entire model, except for the XFEM crack.
- The on cut  check box displays the isosurface where the value of the signed distance function is zero, which corresponds with the surface of an XFEM crack.
- The above cut  check box is not used by an XFEM crack.

Optimization-based view cut components (results data only)

Abaqus/CAE automatically creates and displays a view cut when you open an output database created by an optimization process, for both topology and shape optimizations. By default, the view cut displays the model at the isosurface where the value of the material property is zero, which corresponds with elements that have a density and stiffness close to zero and consequently play an insignificant role in the strength of the model. Boundary conditions are not displayed while an optimization view cut is active.

The tools in the **View Cut Manager** operate as follows for optimizations:

- The below cut  check box displays the material that has a property less than the value of the slider. By default, this is the material that is not contributing to the strength of the model.
- The on cut  check box displays the isosurface where the material has a property equal to the value of the slider.
- The above cut  check box displays the material that has a property greater than the value of the slider. By default, this is the material that continues to contribute to the strength of the model.

Consideration for small faces

A view cut may display inconsistent results if the area of one or more faces of the part is less than 1E–6 because faceting of such small faces is not possible.

Managing view cuts

The view cut manager allows you to create a view cut and to activate, reposition, edit, copy, rename, delete, or customize a previously created view cut.
When you display view cuts for output database data in the Visualization module, the manager also enables you to activate and deactivate display of the resultant forces and moments for the displayed view cut. This section describes how to create and manage view cuts and, for results data in the Visualization module, the display of their associated resultant force and moment vectors.

In this section:

- *Creating or editing a view cut*
- *Displaying a cut section and its resultant force and moment vectors*
- *Allowing for multiple free body cuts*
- *Repositioning a view cut*
- *Copying, renaming, and deleting a view cut*
- *Customizing the cap color for a view cut*
- *Customizing the cut model display in the Visualization module*
- *Customizing display and calculation of resultant force and moment on the active view cuts*
- *Customizing slicing options*

Creating or editing a view cut

You can create view cuts based on a plane. In addition, in the Visualization module, you can create view cuts based on a cylinder, a sphere, or an isosurface corresponding to a constant value of any contour variable. Cuts along the *X*-, *Y*-, and *Z*-planes are created by default.

When you edit an isosurface view cut, Abaqus/CAE updates its definition so that it reflects the current result options and variable selection.

1. Locate the options for creating or editing a view cut.

   - To create a view cut, select **Tools->View Cut->Create** from the main menu bar.

     The **Create Cut** dialog box appears.

   - To edit a previously created view cut, select **Tools->View Cut->Edit** from the main menu bar. From the menu that appears, select the view cut you want to edit.

     The **Edit Cut** dialog box appears.

   Tip: You can also use the view cut manager to create or edit a view cut. From the main menu bar, select **Tools->View Cut->Manager**. To create a view cut, click **Create**. To edit a view cut, select it from the list and double-click on it or click **Edit**.

2. Type a name for the view cut. For more information on naming objects, see *Using basic dialog box components*. You cannot change the name in the **Edit Cut** dialog box, but you can rename a view cut (see *Copying, renaming, and deleting a view cut*).
3. If the **Shape** text field appears in the editor, select the cut shape. Click the arrow next to the **Shape** text field, and select a shape from the list that appears: **Plane**, **Cylinder**, **Sphere**, or **Isosurface**. The shape cannot be changed once the cut has been created.

   The **Shape** text field can be edited only for view cuts created in the Visualization module.
4. If you select **Isosurface** as the cut shape, the cut will be created on a constant value of the current primary field output variable. Toggle on **Override primary variable averaging** to apply a variable averaging threshold of 100% for the selected view cut, or toggle off this option to use the variable averaging threshold specified in the **Averaging** options in the **Result Options** dialog box; for more information, see *Controlling result averaging*

   Isosurface cuts cannot be created for variables stored on surfaces, since no information is available for the interior of the model. See *Selecting the primary field output variable*, for information on changing the primary field output variable.
5. By default, the cut is defined on the displayed frame, and the structure moves through the cut as it deforms. Toggle on **Follow model deformation** to define a Lagrangian cut; i.e., the cut is defined on a reference frame and deforms with the model when you display a different frame. Choose the **Reference frame** to be the **First**, **Last**, or **Current** frame of the analysis.

   The **Follow model deformation** option is available only for view cuts in the Visualization module.
6. For planar, cylindrical, or spherical view cuts in the Visualization module, toggle on the method you wish to use to define the view cut values: **Key-in** or **From CSYS** (the latter option is available only if the output database contains a coordinate system or if you have created a coordinate system in the current session).

   a. If you select the **Key-in** method, enter the coordinates for the following points, depending on the cut shape:

- For **Plane** cuts, enter the coordinates for the origin; the normal axis; and, optionally, axis 2. Axis 2 is an axis lying in the plane; the view cut can be rotated about this axis or about Axis 3, defined as the axis orthogonal to the normal axis and to Axis 2. See *Repositioning a view cut*.
- For **Cylinder** cuts, enter the coordinates for the origin and the cylinder axis.
- For **Sphere** cuts, enter the coordinates for the origin.

b. If you select the **From CSYS** method, choose a coordinate system from the list that appears (systems designated with an asterisk have been saved to the current output database), and select the following, depending on the cut shape:

- For **Plane** cuts, select **Axis 1**, **Axis 2**, or **Axis 3** of the coordinate system to be the normal axis. The cut can be rotated about either of the other two coordinate system axes; see *Repositioning a view cut*.
- For **Cylinder** cuts, select **Axis 1**, **Axis 2**, or **Axis 3** of the coordinate system to be the cylinder axis.
- For **Sphere** cuts, the origin of the coordinate system will be the origin of the cut; you do not need to select anything else.

The **From CSYS** option is available only for view cuts in the Visualization module.

7. Click **OK** to close the **Create Cut** or **Edit Cut** dialog box.

When you create a new view cut, it is activated automatically (i.e., used to display the model in the current viewport). By default, the cut is positioned halfway through the model.

By default, view cuts persist only for the duration of your session. If you want to retain a view cut that you have defined to make it available in subsequent sessions, you can save it to an XML file, to the model database, or to an output database. For more information, see *Managing session objects and session options*.

Additional information

- *Creating coordinate systems during postprocessing*
- *Understanding view cuts*

Displaying a cut section and its resultant force and moment vectors

To display the cut section of your model, you activate a view cut and choose whether to display the model on the cut, below the cut, and/or above the cut. The cut itself is not visible. In the Visualization module you can display the model both below the cut and above the cut; in all other modules, you can display one view or the other.

In the Visualization module you can activate multiple view cuts; in all other modules, only one view cut can be active at a time. For each view cut in the Visualization module, you can also display the resultant force and moment based on a display group, an element set, or the entire model. You can specify free body options that are common to all view cuts or that are specific to a particular view cut.

For view cuts displaying data from an output database, you can also display the resultant force and moment vectors at regular intervals through the model for the selected view cut. For more information, see *Customizing display and calculation of resultant force and moment on the active view cuts*.

The activation status of each view cut persists only for the duration of your session. If you want to retain this view cut activation status for subsequent sessions, you can save these data to an XML file, to the model database, or to an output database. For more information, see *Managing session objects and session options*.

1. From the main menu bar, select **Tools->View Cut->Manager**.

   The **View Cut Manager** appears with a list of all the view cuts that have been created during the current session. A check mark appears in the **Show** column to the left of the active view cut.

2. Click the check box in the **Show** column to the left of the view cut you wish to activate or deactivate.

   The active view cut is used to cut through the model in the current viewport.

   Tip: You can also activate or deactivate a view cut using one of the following methods:

   - In any module other than the Visualization module, click the tool in the View Cut toolbar.
   - In the Visualization module, click the tool in the toolbox.

   If no view cut is active, selecting either of these tools will activate the first view cut in the list in the **View Cut Manager** (usually the default *X*-plane cut). If a view cut is active, selecting this tool will deactivate the cut.

3. Click the check boxes in the **Model** columns to the right of the view cut to display the model below , on , and/or above the cut (these positions are not mutually exclusive).

   The selected portion of the model is displayed in the viewport.

4. If you are working in the Visualization module, click the check box under the icon in the **Model** columns for the active view cut to display the resultant force and moment along that view cut, acting on the visible component. For example, you will see equal and opposite resultant force and moment vectors when you switch from the “Show below cut” option to the “Show above cut” option.

   To specify free body options for a particular view cut, click the icon to the right of the view cut for which you want to specify options.

   Note:

   The **View Cut Options** (common to all view cuts or specific to a particular view cut) enable you to control three aspects of the resultant force and moment display on view cuts. You can

compute the resultant force and moment based on a display group, an element set, or the whole model; and you can adjust the summation point and component resolution. You can enable display of the resultant force and moment on a view cut for solid geometry, composite solid sections, shell sections, or beam sections. For resultant forces and moments to be displayed, the output database must include element force nodal output (NFORC) for composite solid sections and section force (SF) and section moment (SM) output for shell sections and beam sections. For more information, see *Customizing display and calculation of resultant force and moment on the active view cuts*.

5. In the Visualization module, toggle on **Allow for multiple cuts** to enable the display of multiple view cuts.

Additional information

- *Using the extended finite element method to model fracture mechanics*
- *Creating or editing a view cut*

Allowing for multiple free body cuts

In the Visualization module you can display multiple view cuts on your model at the same time by toggling on **Allow for multiple cuts**. When you select this option, Abaqus/CAE displays the cutting surface for all view cuts in your session for which the **Show** checkbox is selected. You can reposition only the active view cut. In practice, you might want to activate and reposition each view cut in turn to place them in the locations you want to investigate. In addition, the cuts you display do not interact; for example, you cannot display only the portion of your model that lies between two parallel planar view cuts.



Note:

Abaqus/CAE does not support display of multiple XFEM-based view cut components at the same time.

1. From the main menu bar, select **Tools->View Cut->Manager**.

   The **View Cut Manager** appears with a list of all the view cuts that have been created during the current session. A check mark appears in the **Show** column to the left of the active view cut.

2. Toggle on **Allow for multiple cuts**.

   Abaqus/CAE displays the on-cut position for every view cut for which the **Show** checkbox is selected, and Abaqus/CAE hides all free body cuts and disables the check boxes under the  icon in the **Model** columns. You can show or hide individual view cuts from the current display by toggling their **Show** checkboxes.

Additional information

- *Using the extended finite element method to model fracture mechanics*
- *Creating or editing a view cut*

Repositioning a view cut

By default, when a view cut is first activated, Abaqus/CAE positions it halfway through the model. You can use the view cut manager to reposition cuts on the model. You can translate plane cuts along the normal axis or rotate them about the other two axes. You can redefine the radius of cylindrical or spherical cuts and the contour value used for isosurface cuts to reposition where they cut through the model. The positioning options in the view cut manager apply only to the selected cut, which may be different from the currently active cut.

In all modules except the Visualization module, Abaqus/CAE repositions the cutting plane back to the halfway point if you modify the model.

1. From the main menu bar, select **Tools->View Cut->Manager**.

   The **View Cut Manager** appears with a list of all the view cuts that have been created during the current session.

2. Select a view cut in the manager by clicking on its name.

   The selected view cut is highlighted in the manager.

3. To translate a **Plane** view cut:

   1. Select **Translate** from the menu at the bottom of the view cut manager.
   2. Reposition the view cut using one of the following methods:

      - Enter a value for the **Position** along the normal axis.
      - Drag the **Position** slider to select a value. The slider is available only if the view cut is currently active.
      - Increase the **Sensitivity** of the slider for fine control over view cut repositioning. Each sensitivity increment focuses the slider's range of motion by a factor of 10, so a sensitivity setting of 100 enables you to reposition a view cut within 1/100 of the model.

        Alternatively, you can double-click a location in the **Sensitivity** bar to reposition the view cut to a general location, then drag the **Position** slider to select a value.

   The selected cut is translated by the specified value.

4. To rotate a **Plane** view cut:

   1. Select **Rotate** from the menu at the bottom of the view cut manager.
   2. Select the axis of rotation (the 2- or 3-axis for a cut defined by the **Key-in** method or either of the two non-normal axes for a cut defined by the **From CSYS** method; see *Creating or editing a view cut*); and enter a value for the **Angle** of rotation, or use the slider to select a value.

      The selected cut is rotated about the origin of the axis of rotation by the specified angle. Any translations applied to the view cut are ignored.

5. To reposition a **Cylinder**, **Sphere**, or **Isosurface** view cut, enter a value in the available text field, or use the slider to select a value. The slider is available only if the view cut is currently active; you can increase the slider's sensitivity for greater precision.

   - For **Cylinder** cuts enter a value for the **Radius** of the cylinder.
   - For **Sphere** cuts enter a value for the **Radius** of the sphere.
   - For **Isosurface** cuts enter a value for the contour **Value** to which the cut should correspond.

The selected cut is repositioned as specified.

Additional information

• *Creating or editing a view cut*

Copying, renaming, and deleting a view cut

You can copy, rename, or delete a view cut using commands from the main menu bar or the view cut manager.

To copy, rename, or delete a view cut, use one of the following:

- The **Copy**, **Rename**, and **Delete** items listed under the **Tools->View Cut** menu on the main menu bar.

  The **Copy**, **Rename**, and **Delete** items contain submenus listing all the view cuts you have created.
- The view cut manager dialog box. The view cut manager contains functions identical to those listed under the **Tools->View Cut** menu on the main menu bar but with a convenient browser that lists all the view cuts. To display the view cut manager dialog box, select **Tools->View Cut->Manager** from the main menu bar.



Note:

Abaqus/CAE creates a view cut when you open a model containing a crack calculated by the extended finite element method (XFEM) and when you open a model created by an optimization process. You cannot create, delete, or edit an XFEM view cut or an optimization view cut.

Additional information

• *Creating or editing a view cut*

Customizing the cap color for a view cut

In modules other than the Visualization module, you can customize the “cap” that appears when you display the portion of the cutting plane on the view cut.

The **Cap Color** options provide two choices for customizing the cap color:

- Select **Specify** and click  to display the entire cap with a fixed color. Abaqus/CAE displays that color for all components on the cutting plane and does not change the color as you move the plane or change the color coding selections.
- Select **Use body color** to display the current colors of each component in the model on the cutting plane. Abaqus/CAE changes the coloring of the model components on the cutting plane dynamically as you reposition the plane or change the color coding selections.



Note:

Abaqus/CAE displays different cap colors for different parts; however, Abaqus/CAE does not display a different cap color for different sections, materials, mesh defaults, etc., in the same part.

Abaqus/CAE hides the cap from the viewport if you toggle on translucency for your session; for more information, see *Changing the translucency*. In addition, Abaqus/CAE might hide the cap when you save or export the contents of the viewport, depending on the file format you select.

- If you save a viewport image to a file and select a vector-based file format, Abaqus/CAE hides the cap in the resulting file. Abaqus/CAE creates a vector-based file when you save the image in **SVG** format or when you save the image in **PS** or **EPS** format and specify **Vector** as the **Image format** from the **PS Options** dialog box or the **EPS Options** dialog box.
- If you export viewport data in either VRML or 3DXML format, Abaqus/CAE hides the cap in the exported file. The cap is included in the exported file for all other export formats.

1. From the main menu bar, select **Tools->View Cut->Manager**.

   The **View Cut Manager** dialog box appears.
2. Click **Options**.

   The **View Cut Options** dialog box appears.
3. From the list of **Cap Color** options, select one of the following:

   - Select **Specify** and click , then select a cap color from the **Select Color** dialog box that appears to display the selected color for the entire portion of the model displayed on the cut. For more information about selecting a new color, see *Customizing colors*.
   - Select **Use body color** to color the cap using the current color coding settings for each component that intersects the view cut.
4. Click **OK**.

Additional information

- *Controlling the destination and appearance of printed images*
- *Exporting geometry, model, and mesh data*

Customizing the cut model display in the Visualization module

You can use the view cut options to customize the appearance of a cut model in the Visualization module. You can select different options for the portions of the model below the cut, on the cut, and above the cut.

You can choose to apply the view cut options to the model or to use the current plot options. View cut options related to the cut model display are associated with the viewport rather than with an individual view cut; therefore, if the view cut options are applied to the model, any active cut will use them. *Figure 1* illustrates how the view cut options can be used to customize the model on the cut plane.

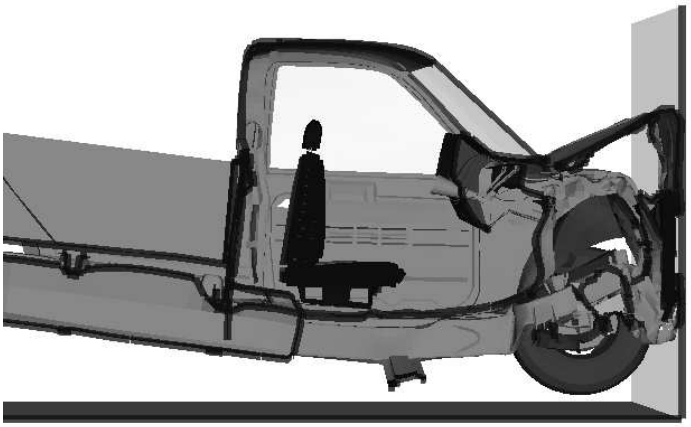


Figure 1: Longitudinal view cut through a truck model with customized plot options on the cut plane.

1. From the main menu bar in the Visualization module, select **Options->View Cut**.
   The **View Cut Options** dialog box appears.

   Tip: You can also use the view cut manager to customize the cut options. From the main menu bar, select **Tools->View Cut->Manager**, and click **Options** in the view cut manager.

2. Select the portion of the model that you wish to customize by clicking one of the following tabs: **Below Cut**, **On Cut**, **Above Cut**.
3. Toggle on **Use these options** to apply the view cut options to the selected portion of the model. Alternatively, toggle on **Use current plot options** to apply the common plot options or, if applicable, superimposed plot options to the selected portion of the model.
4. Click the following tabs to customize the appearance of view cuts in the current viewport:
   - **Basic**: Choose render style and edge visibility.
   - **Color & Style**: Control model edge color and style and model face color.
   - **Other**: The **Other** page contains the following tab:
     - **Translucency**: Control shaded render style translucency.

   To learn how to customize the render style and other display characteristics of your view cut, see *Customizing plot display*.

Customizing display and calculation of resultant force and moment on the active view cuts

You can use the free body options for view cuts in the Visualization module to customize how the resultant force and moment displayed on the active view cut are calculated and to customize the summation point and coordinate system transformation.

You can specify free body options that are common to all view cuts or that are specific to a particular view cut. Abaqus/CAE can calculate the resultant force and moment by cutting through a display group, an element set, or the whole model. You can also customize the content and appearance of all free body cuts in the current viewport; see *Customizing general display options for free body cuts*.

Display of resultant force and moment data is available for view cuts of output databases only. You cannot display this data for view cuts when a model from the current model database is displayed in the Visualization module.

You may observe different values for the resultant forces and moments if you compare the values in the deformed and undeformed shape plots. These differences can occur because view cut–based free body force computations integrate the stress field over the area, and, if you toggle the plot state, the elements that are part of the view cut can change or the elemental deformations can change the area of the cross section.

By default, Abaqus/CAE displays the resultant force and moment only once for each view cut for which you toggle on display of the resultant force and moment. If desired, you can also display a series of vectors that show the resultant force and moment data at regular intervals through the entire model or through a portion of the model.

1. From the main menu in the Visualization module, select **Options->View Cut**.

   The **View Cut Options** dialog box appears.

   Tip: You can also use the view cut manager to customize the cut options using one of the following methods:

   - To specify options common to all view cuts, select **Tools->View Cut->Manager** from the main menu bar, and click **Options**.
   - To specify options for a particular view cut, select **Tools->View Cut->Manager** from the main menu bar, and click the icon to the right of the view cut for which you want to specify options.

2. Click the **Free Body** tab.
3. If you are defining cut-specific options, select **Use these options**.
4. From the **Computation based on** options:

   - Select **Cutting through a display group** and select a display group to compute the resultant force and moment based on a view cut through the selected display group. By default, the current display group is selected.
   - Select **Cutting through an element set** and select an element set in the output database to compute the resultant force and moment based on a view cut through the selected set.
   - Select **Cutting through the whole model** to compute the resultant force and moment based on a view cut through the entire model.

5. If desired, customize the summation point or the coordinate system transformation options for the resultant force and moment vectors arising from view cuts.

   a. From the **Summation point** options, select the three-dimensional location from which the resultant force and moment vectors originate.

      - Select **Centroid of cut** to place the summation point automatically at the centroid of the surface of the view cut.

Note:

When the view cut cuts through a shell, the centroid is based on the length times the shell thickness.

When the view cut cuts through a beam, the centroid is at the nodal connectivity line.

- Select **User-defined**, and specify a custom three-dimensional location in space or click to select the summation point from the viewport. The summation point is highlighted in the viewport.

b. From the **Component resolution** options, you can specify the coordinate system transformation that takes place when vectors are displayed in component form. (See *Customizing general display options for free body cuts*, for more information about displaying force and moment vectors in component form.)

- Select **Normal and tangential** to orient the component vectors with the normal and the tangent of the surface you select.

  If desired, specify the *Y*-axis value of the tangent for the component vectors. This option enables you to enforce a uniform direction for the tangent vector when you display a series of free body cuts on view cut slices.

- Select **CSYS** and a coordinate system to transform the component vectors to a custom coordinate system. Alternatively, you can click **Create** to create a new datum coordinate system.

The **Component resolution** options affect the display of resultant forces and moments only when component vector display is selected in the **Free Body Plot Options** dialog box.

6. Toggle on **Show heat flow rate if available** to display the heat flow rate in terms of energy per time across the view cut.
7. If you want to customize the content and appearance of the free body cut displayed on the active view cut and those created using the Free Body toolset, click to access the **Free Body Plot Options**; this option is unavailable if you are defining cut-specific options. See *Customizing general display options for free body cuts*.
8. Click **OK**.

   By default, view cuts persist only for the duration of your session. If you want to retain a view cut that you have defined to make it available in subsequent sessions, you can save it to an XML file, to a model database, or to an output database. For more information, see *Managing session objects and session options*.
9. Click **Apply** to implement your changes.

   The view cuts change to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent view cuts in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Customizing slicing options

View cut slices are similar to the on-cut portion of a view cut, but they are displayed at a series of intervals along the range of the active view cut, rather than on just the cutting plane itself.
You can use the settings on the **Slicing** tabbed page in the Visualization module to display a series of view cut slices along the range of the active view cut. You can display view cut slices at regular intervals along the range of the active view cut, or you can display the slices along a predefined path in your session. When you enable slicing, you can also display the resultant force and moment at that location of the model.

You can specify slicing options that are common to all view cuts or that are specific to a particular view cut. View cut slicing can be enabled only when a single view cut is active and only in the Visualization module. However, if you define cut-specific slicing options and obtain reports for multiple active view cuts, the slicing options are considered in the free body computations. You can display slices for both translational and rotational view cut display. Slices are displayed along the entire range of the active view cut regardless of any repositioning of the view cut in the **Motion of Selected Cut** options in the **View Cut Manager**. For rotational view cuts, you can also rotate the slices and change their axis of rotation.

If you enable view cut slicing along a path in a model that has a great deal of folding or curvature, the cutting surfaces of the slices can sometimes cross different parts of your model at the same time. You can limit or eliminate this behavior by focusing on a subset of your model: either use a display group to hide a portion of your model or change the path definition so that slicing is displayed only along some of the curved parts.

1. From the main menu bar in the Visualization module, select **Options->View Cut**.

   The **View Cut Options** dialog box appears.

   Tip: You can also use the view cut manager to customize the cut options using one of the following methods:

   - To specify options common to all view cuts, select **Tools->View Cut->Manager** from the main menu bar, and click **Options**.
   - To specify options for a particular view cut, select **Tools->View Cut->Manager** from the main menu bar, and click the  icon to the right of the view cut for which you want to specify options.

2. Click the **Slicing** tab.
3. If you are defining cut-specific options, select **Use these options**.
4. Toggle on **Display slicing** to enable slicing for view cuts in your session.
5. If you want to display view cut slices along the entire range of the view cut, select **Step through the active view cut's range**. If you display view cut slices or free body cuts in multiple locations for the selected view cut, you can also customize the minimum and maximum values of the line segment within which Abaqus/CAE will display the resultant forces and moments. By default, the minimum and maximum values are set to allow for distribution of slices throughout the entire model.
6. If you want to display view cut slices along a predefined path, select **Step along a predefined path**, then do the following:
   a. Select the path.
   b. Specify the locations where you want to display view cut slices using the **Display slices at path nodes** option:
      - Toggle on this option to place the view cut slices at the nodes that comprise the path definition.
      - Toggle off this option to place the view cut slices at a number of regular intervals along the path. Specify the number of slices from the field and slider at the bottom of the dialog box.

c. Specify the location of the summation point for the resultant forces and moments using the **Set free body summation point on the path** option:

- Toggle on this option to place the summation point for the resultant forces and moments on the selected path.
- Toggle off this option to place the summation point for the resultant forces and moments at the location about which resultant moments are taken.

7. Click **Apply** to implement your changes.

   The view cuts change to reflect your specifications.

   By default, your changes are saved for the duration of the session and will affect all subsequent view cuts in that viewport. If you want to retain your changes for subsequent sessions, save them to a file. For more information, see *Saving customizations for use in subsequent sessions*.

Additional information

• *Creating a path through your model*

Using plug-ins

This part of the guide describes how you can use plug-ins and the Plug-in toolset to extend the capabilities of Abaqus/CAE.

In this section:

- *The Plug-in toolset*
- *Abaqus plug-ins*

The Plug-in toolset

The Plug-in toolset loads plug-in files into Abaqus/CAE.

In this section:

- *What is a plug-in?*
- *Where can I get plug-ins?*
- *How can I get information about a plug-in?*
- *An example of a Python module and a function*
- *What can I do with a GUI plug-in?*
- *How do I make a plug-in available from Abaqus/CAE?*
- *How are kernel plug-ins executed?*
- *Overwriting plug-ins*
- *How are GUI plug-ins executed?*
- *Hiding a plug-in's source code*
- *Displaying exceptions for imported plug-ins at startup*
- *Abaqus/CAE modules and plug-ins*
- *How can I provide information about a plug-in?*

What is a plug-in?

A plug-in is a piece of software that installs itself into another application to extend the capabilities of that application. Abaqus plug-ins execute Abaqus Scripting Interface and Abaqus GUI Toolkit commands, and they provide a way to customize Abaqus/CAE for your particular needs or preferences. For example, a simple plug-in could automatically print the contents of the current viewport according to some predefined options. A more complex plug-in could provide a graphical user interface to a specialized postprocessing routine that you have written.

There are two types of plug-ins: kernel and GUI. A kernel plug-in consists of functions written using the Abaqus Scripting Interface. In contrast to a kernel plug-in, a GUI plug-in is written using the Abaqus GUI Toolkit and contains commands that create graphical user interfaces, which in turn send commands to the kernel. Both kernel and GUI plug-ins are available from the **Plug-ins** menu on the main menu bar or from a plug-ins toolbox.

By default, several kernel and GUI example plug-ins are available from the **Plug-ins** menu on the main menu bar. You can use these examples to see how plug-ins are created and to learn how plug-ins interact with Abaqus/CAE. In addition, you can access two example plug-ins from a plug-ins toolbox. When you select **Plug-ins->Toolboxes->Examples** from the main menu bar, Abaqus/CAE displays the **Examples** toolbox. You can click an icon in the toolbox to start a plug-in. *Figure 1* shows the **Toolboxes** menu and the **Examples** toolbox.

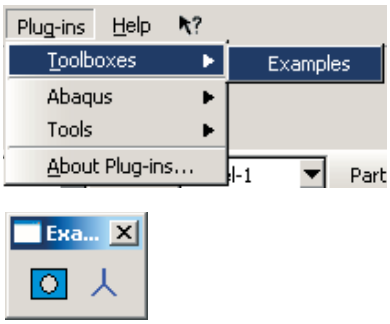


Figure 1: The Toolboxes menu and the Examples plug-in toolbox.

Where can I get plug-ins?

Several plug-ins are provided with the Abaqus installation.
You can view these plug-ins by selecting **Plug-ins->Abaqus** or **Tools** from the main menu bar.

For information on how to view a document that describes the plug-in and its usage, see *How can I get information about a plug-in?*.

Additional plug-ins are available in the SIMULIA Community (*SIMUILA Community > Learning Resources > 3DEXPERIENCE and Traditional Products > Abaqus > Plug-ins/Scripts*), which provides example plug-ins as well as access to a community of users that fosters the advance of the Abaqus Scripting Interface and the Abaqus GUI Toolkit. Click **Blog** and filter by **Process Automation** to browse plug-in examples in this community. You can also write your own plug-ins, as described in this section.

How can I get information about a plug-in?

The **Plug-ins** menu that appears in the Abaqus/CAE main menu bar contains an **About Plug-ins** item that displays the **About Plug-ins** dialog box. This dialog box lists all the plug-ins that are currently installed. As you click on each plug-in in the tree list, Abaqus/CAE displays information about that plug-in, such as the author and the version. In addition, you can click **View** in the **About Plug-ins** dialog box to view a document that describes the plug-in. For more information, see *How can I provide information about a plug-in?*.

This information is specified as optional arguments to the plug-in registration command. As a result, if the author of the plug-in chose not to supply some of these optional arguments, the corresponding information will not be available in the **About Plug-ins** dialog box.

An example of a Python module and a function

A kernel plug-in associates a Python module and a function with a menu item or a toolbox icon. For example, the kernel plug-in shown below is a simple function defined in the file `myUtils.py` that prints the current viewport to a PNG file. The file `myUtils.py` is a Python module.

```
def printCurrentVp():

    from abaqus import session, getInputs
    from abaqusConstants import PNG
    name = getInputs( (('File name:', ''),),
        'Print current viewport to PNG file')[0]
    vp = session.viewports[session.currentViewportName]
    session.printToFile(
        fileName=name, format=PNG, canvasObjects=(vp, ))
```

The first line of the example (`def printCurrentVp():`) is a function definition that contains the indented commands. A plug-in requires a function definition. As a result, if you want to create a kernel plug-in by extracting the commands written to the `abaqus.rpy` replay file, you must first wrap the commands in a function definition. For more details on writing kernel scripts and creating function definitions, refer to the *Abaqus Scripting User's Guide* and to the *Abaqus Scripting Reference Guide*.

After you have written a kernel plug-in, you can execute it from the **Plug-ins** menu in the Abaqus/CAE main menu bar after you register it with the Plug-in toolset. For more information, see *How do I make a plug-in available from Abaqus/CAE?*.

What can I do with a GUI plug-in?

A GUI plug-in is written using the Abaqus GUI Toolkit. For more details on writing GUI scripts, refer to the *Abaqus GUI Toolkit User's Guide* and to the *Abaqus GUI Toolkit Reference Guide*. In addition to GUI commands, a GUI plug-in usually contains a kernel module that defines the function that will be executed when the GUI issues the command to the kernel. For example, you could write a GUI plug-in that presents the user with a dialog box for entering the dimensions of a plate, as shown in *Figure 1*.

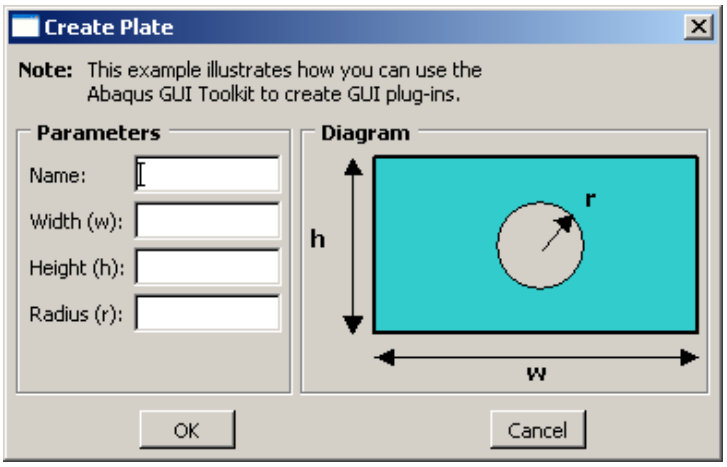


Figure 1: A dialog box for entering the dimensions of a plate.

When the user clicks the **OK** button to close the dialog box, the GUI plug-in constructs a command that is sent to the kernel for execution. The kernel function builds a part using the dimensions entered in the dialog box.

A GUI plug-in is similar to a kernel plug-in in that you must register it with the Plug-in toolset before you can execute it from the **Plug-ins** menu in the Abaqus/CAE main menu bar.

How do I make a plug-in available from Abaqus/CAE?

To make a plug-in available from Abaqus/CAE, you must place specially named files containing registration commands in one of the directories that Abaqus/CAE searches for plug-ins.
The registration commands can make a plug-in available from the **Plug-ins** menu in the main menu bar, from a separate plug-in toolbox, or from both. This section describes how you make a plug-in available from Abaqus/CAE.

In this section:

- *Where are plug-in files stored?*
- *What are the kernel and GUI registration commands?*
- *An example of adding a kernel plug-in to the Plug-ins menu*
- *An example of adding a kernel plug-in to a plug-ins toolbox*
- *An example of adding a GUI plug-in to the Plug-ins menu*

Where are plug-in files stored?

Abaqus plug-ins can be stored in several locations.
Native Abaqus/CAE plug-ins are included with the Abaqus/CAE installation. External plug-ins (any plug-in installed after the Abaqus/CAE installation) should not be placed inside the Abaqus/CAE installation.

Instead, you should install external plug-ins in any of the following locations:

- *home_dir*`\abaqus_plugins`, where *home_dir* is your home directory.
- *current_dir*`\abaqus_plugins`, where *current_dir* is the current directory.
- **plugin_central_dir** is an Abaqus environment parameter used to define one or more specific directories where plug-ins are stored. This is typically a central location accessed by all users at your site if the directory is mounted on a file system that all users can access. **plugin_central_dir** can be defined in the `abaqus_v6.env` file or the Abaqus solver `custom_v6.env` file. For example,

  ```
  plugin_central_dir = "C:\\SIMULIA\\CAE\\plugins;U:\\shared\\CAE\\plugins"
  ```

  Directories are separated by a semicolon on Windows systems or by a colon on Linux systems. For more information, see *Using the Abaqus environment files*.

Abaqus/CAE will import any files in these directories that match the naming convention `*_plugin.py`. These files must contain registration commands. The name of the file must follow the rules described in *Standard Abaqus Scripting Interface exceptions*.

What are the kernel and GUI registration commands?

The plug-in registration commands are located in the Plug-in toolset, which you can access from the main menu bar of Abaqus/CAE. To access the registration commands, your script should start with the following statements:

```
from abaqusGui import getAFXApp
toolset=getAFXApp().getAFXMainWindow().getPluginToolset()
```

You can use the `toolset` variable to do the following:

Add an item to the Plug-ins menu

The following statements add a kernel and a GUI plug-in to the **Plug-ins** menu:

```
toolset.registerKernelMenuButton()
toolset.registerGuiMenuButton()
```

Add an icon to a plug-ins toolbox

The following statements add a kernel and a GUI plug-in to a plug-ins toolbox:

```
toolset.registerKernelToolButton()
toolset.registerGuiToolButton()
```

The registration commands are described in *Plug-in registration commands*.

An example of adding a kernel plug-in to the Plug-ins menu

You use the `registerKernelMenuButton()` command to make a kernel plug-in available from the **Plug-ins** menu. The following example illustrates how you associate the simple function described in *An example of a Python module and a function*, with a **Print Current Viewport** item in the **Plug-ins** menu. (The function prints the current viewport to a PNG file.) The registration command references the kernel function `printCurrentVP` that is located in the Python module `myUtils`:

```
from abaqusGui import getAFXApp
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerKernelMenuButton(
    buttonText='Print Current Viewport',
    moduleName='myUtils', functionName='printCurrentVp()' )
```

The first two lines in the example provide access to the commands in the Plug-in toolset. The next line inserts a **Print Current Viewport** menu item under the **Plug-ins** menu in the main menu bar. When the user clicks **Print Current Viewport**, Abaqus/CAE issues the following command to the kernel:

```
myUtils.printCurrentVp()
```

The `buttonText` argument accepts a pipe-separated (|) list of words. The pipe separates submenu names from the menu button name. This allows you to group several plug-ins under one menu. For example, if you have defined more functions in `myUtils.py`, you can use the following version of `myUtils_plugin.py` to register them under a cascading menu:

```
from abaqusGui import getAFXApp
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerKernelMenuButton(
    buttonText='My Utils|Print Current Viewport',
    moduleName='myUtils', functionName='printCurrentVp()' )

toolset.registerKernelMenuButton(
    buttonText='My Utils|Print Sections',
    moduleName='myUtils', functionName='printSections()' )

toolset.registerKernelMenuButton(
    buttonText='My Utils|Print Materials',
    moduleName='myUtils', functionName='printMaterials()' )
```

The previous example results in the menu structure shown in *Figure 1*:

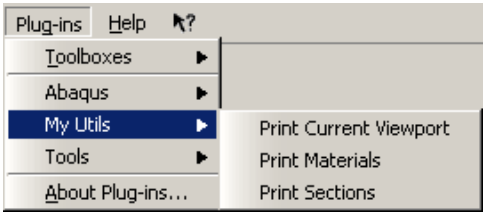


Figure 1: The effect of registering plug-ins under a cascading menu.

An example of adding a kernel plug-in to a plug-ins toolbox

You use the `registerKernelToolButton()` command to make a kernel plug-in available from a plug-ins toolbox. When you add a plug-in to a toolbox, an item appears in the **Plug-ins->Toolboxes** menu. When you select this item, Abaqus/CAE displays a plug-ins toolbox. Although you can display text in a toolbox, you will probably want to supply an icon instead and not use text (for more information on creating icons, see *Icons*).

The `registerKernelToolButton` registration command uses the same arguments as the `registerKernelMenuButton` command; however, it also requires a toolbox name. You can use the `buttonText` argument to specify only a tooltip by preceding the tip text with "\t" as shown in the following version of `myUtils_plugin.py`:

```
from abaqusGui import getAFXApp, FXXPMIcon
from myIcons import vpIconData
vpIcon = FXXPMIcon(getAFXApp(), vpIconData)

toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerKernelToolButton(toolboxName='My Utils',
    buttonText='\tPrint Current Viewport', icon=vpIcon,
    moduleName='myUtils', functionName='printCurrentVp()' )
```

This example creates a **My Utils** item under the **Plug-ins->Toolboxes** menu. When a user clicks **My Utils**, Abaqus/CAE displays a toolbox with **My Utils** shown in its title bar. When you click the icon in the toolbox, Abaqus/CAE sends the following command to the kernel:

```
myUtils.printCurrentVp()
```

The same command would be sent to the kernel if you registered the plug-in in the **Plug-ins** menu.

The **Toolboxes** item, if it exists, always appears first in the **Plug-ins** menu; and the **About Plug-ins** item always appears last. Other items in the **Plug-ins** menu are listed alphabetically.

An example of adding a GUI plug-in to the Plug-ins menu

You use the `registerGuiMenuButton()` command to make a GUI plug-in available from the **Plug-ins** menu. The following example shows how you can use registration commands to add a GUI plug-in that creates a form that posts a dialog box prompting the user for input. When the user selects **Plug-ins->My Utility**, Abaqus/CAE activates the form.

Registration commands for kernel plug-ins must reside in a file separate from the kernel plug-in code. In contrast, registration commands for GUI plug-ins can reside in a separate file or in the GUI plug-in file itself. This example shows how you can combine the registration commands and the GUI plug-in code into a single file.

```
from abaqusGui import AFXForm, getAFXApp

class MyForm(AFXForm):

    form code goes here

toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerGuiMenuButton(buttonText='My Utility',
    object=MyForm(toolset) )
```

When you make a GUI plug-in available from the **Plug-ins** menu or from a toolbox, the registration command references a GUI object that receives a message when the plug-in is selected. The selector of the message is generated by combining the message ID that you specify together with the `SEL_COMMAND` message type. In most cases you will supply a form or procedure as the GUI object and use `ID_ACTIVATE` (the default) for the message ID. Forms and procedures prompt the user for input, in most cases using a dialog box or the prompt line. After receiving input from the user, forms and procedures then issue commands to the kernel. For details on creating forms, procedures, and dialog boxes, see *Modes* and *Dialog boxes*. You can also create simple dialog boxes using the **Really Simple GUI Builder** plug-in; for more information, see *Creating dialog boxes with the Really Simple GUI (RSG) Dialog Builder*.

How are kernel plug-ins executed?

Abaqus executes a kernel plug-in by issuing a command to the kernel of the form *moduleName.functionName*. The module name and the function name are the names that you supplied in the registration command for that plug-in.

When Abaqus/CAE starts and imports a plug-in, the directory in which the plug-in is located is stored by Abaqus/CAE. The first time that the plug-in is invoked, Abaqus/CAE updates the kernel's *sys.path* list with that plug-in's directory. The next time that plug-in is invoked, Abaqus/CAE issues the commands inside the plug-in but does not update the *sys.path*.

For example, consider the `myUtils_plugin.py` and `myUtils.py` files described in previous sections. Assume that you stored these two files in a subdirectory called `myPlugins` in the `abaqus_plugins` directory in your home directory. The first time that you click **Print Current Viewport** in the **Plug-ins** menu, Abaqus/CAE sends the following commands to the kernel:

```
import sys
sys.path.append('path to your home dir/abaqus_plugins/myPlugins')
import myUtils
myUtils.printCurrentViewport()
```

The next time you click **Print Current Viewport** in the **Plug-ins** menu, Abaqus/CAE sends only the following command to the kernel:

```
myUtils.printCurrentViewport()
```

Since Abaqus/CAE updates the *sys.path* list, your plug-in code does not need to perform this task to import modules that it needs. This assumes that those modules are located in the same directory as the plug-in. If you have to import modules that are not located in the same directory, you must augment the *sys.path* list. If you need to augment the *sys.path* list, you should use functions that determine the file locations automatically, rather than hard-coding the path to the plug-ins in your file. This makes it easy to move plug-ins to different locations without having to modify their code. The following example of `myUtils.py` illustrates an example of augmenting the *sys.path* list:

```
import sys, os

# Full path (with name) to this file
absPath = os.path.abspath(__file__)

# Full directory specification
absDir = os.path.dirname(absPath)

# Full subdirectory specification
subDir = os.path.join(absDir, 'mySubDir')
sys.path.append(subDir)

# myModule is located in subDir
import myModule

rest of module code
```

Overwriting plug-ins

Upon startup, Abaqus/CAE searches for plug-ins in the Abaqus/CAE installation, your home directory, the current directory, and the directories specified by the **plugin_central_dir** Abaqus environment parameter. If Abaqus/CAE finds a plug-in with the same name and case in multiple directories, the plug-in found later in the search will overwrite one found earlier in the search.
Plug-in names are case sensitive. For example, a plug-in named `myPlugin` is different from a plug-in named `MyPlugin`. The overwriting approach enables you to use a customized version of a plug-in that resides only in your home directory or current directory; however, in most cases you should avoid naming conflicts with plug-ins.

Python searches for modules according to the order specified in the *sys.path* list and stops searching as soon as it finds a match for the module name. For example, if you have a utility module called `myUtils.py` stored in two different plug-in directories and these directories contain different code, both plug-ins will reference the same module—the one found first in the *sys.path* list.

You can use Python's package style for storing and referencing your modules to avoid unintentional overwriting of plug-ins or accessing of the wrong module. Start by creating a directory with a detailed name that is unlikely to be duplicated by others, and store your `_plugin.py` file in that directory. Create another directory of the same name within the first directory, and place your other plug-in files in the newer directory with an empty file named `__init__.py`, so that the directory structure looks as follows:

```
torqueCalculator
   torqueCalculator_plugin.py
   torqueCalculator
      __init__.py
      torqueUtils.py
      torqueForm.py
      torqueDB.py
```

When you import files in your code, use the package name to qualify the import. For example, inside `torqueForm.py`, you can qualify the import with the following statement:

```
from torqueCalculator.torqueDB import TorqueDB
```

How are GUI plug-ins executed?

Abaqus/CAE executes a GUI plug-in by sending a message to the GUI object specified in the registration command for that plug-in. As with the kernel plug-ins, the first time that a GUI plug-in is invoked, Abaqus/CAE updates the kernel's *sys.path* list with that plug-in's directory. In addition, Abaqus/CAE sends the plug-in's `kernelInitString` to the kernel and updates the GUI's *sys.path* list to include that plug-in's directory. You can use the `kernelInitString` to initialize the kernel for commands that will be sent by the plug-in's GUI. The next time that plug-in is invoked, only the message will be sent to the GUI object.

If you need to import files into the GUI that are not located in the same directory as the plug-in, you must augment the *sys.path* list as shown in the previous example. In addition, if you need to augment the kernel's *sys.path* list, you should supply code similar to the following in the plug-in's `kernelInitString`, in this example the name of the file containing the plug-in is `myForm_plugin.py`:

```
from abaqusGui import AFXForm, getAFXApp

class MyForm(AFXForm):

    form code goes here

import os

# Full path (with name) to this file
absPath = os.path.abspath(__file__)

# Full directory specification
absDir = os.path.dirname(absPath)

# Full subdirectory specification
subDir = os.path.join(absDir, 'mySubDir')
initString  = "sys.path.append('%s')\n" % subDir

# myModule is located in subDir
initString += 'import myModule'
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerGuiMenuButton(buttonText='My Utility',
    object=MyForm(toolset), kernelInitString=initString )
```

Hiding a plug-in's source code

If you do not want other users to see the source code of your plug-in, you can supply the compiled version of the plug-in instead of supplying the source code version. When Abaqus/CAE imports the source code version of a plug-in, Abaqus/CAE first parses the plug-in and then compiles it into a executable version of the file (called *filename*`.pyc`) before it executes the file. If the source code has not changed since the last time the plug-in was compiled, Abaqus/CAE executes the compiled version directly and bypasses the compilation step. Abaqus/CAE will still import the plug-in if you delete the file containing its source code but retain the compiled version of the plug-in.

If you have only compiled versions of some plug-ins, you may want to store them in a directory under the `abaqus_plugins` directory. This will avoid confusion by keeping compiled versions of plug-ins that do not have source code separate from the compiled versions of plug-ins that still have the source code.

Displaying exceptions for imported plug-ins at startup

By default, Abaqus/CAE does not display the exceptions associated with the import of plug-ins when you start the application. If you want to expose these exceptions for debugging purposes, set the environment variable `ABQ_PLUGIN_DEBUG` to 1 at a command prompt before launching Abaqus/CAE. When this environment variable is set, Abaqus/CAE provides more trackback information about plug-ins upon startup, including the location and nature of any failures that occur.

Abaqus/CAE modules and plug-ins

By default, a plug-in appears in all Abaqus/CAE modules. However, you can use the `applicableModules` argument of the registration commands to specify individual modules in which a plug-in is available. If a plug-in is not applicable to the current module, its menu item will be hidden in the **Plug-ins** menu. For example, if you specify that a plug-in is available only in the Part module, you will not see that menu item under the **Plug-ins** menu unless you are in the Part module. The following registration command makes the plug-in available in only the Part and Assembly modules:

```
toolset.registerKernelMenuButton(
    buttonText='Print Materials',
    moduleName='myUtils', functionName='printMaterials()',
    applicableModules=('Part','Assembly'))
```

Abaqus/CAE uses similar functionality to hide items in the **Tools** menu as you move between modules. For a full description of the arguments to the kernel and GUI registration commands, see *Plug-in registration commands*.

How can I provide information about a plug-in?

When you create a plug-in, you can supply the following arguments to the registration command:

author

A string specifying the author's name.

version

A string specifying the version of the plug-in.

description

A string specifying a brief description of the plug-in.

helpUrl

A string specifying a path to a URL that provides information about your plug-in.

For a full description of the arguments to the kernel and GUI registration commands, see *Plug-in registration commands*.

The `helpUrl` argument can be any valid web browser URL. In most cases the URL will point to an HTML file, but the URL can also point to a plain text file or to an external website. If you provide a help file with your plug-in that will be referenced by the `helpUrl` argument, you should avoid hard-coding directory names by constructing the path to the help file using the techniques shown in previous examples. The following example of `myUtils_plugin.py` shows how you can provide information about the plug-in and construct the relative path to the `helpUrl` argument:

```
from abaqusGui import getAFXApp
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()

import os
helpUrl = os.path.join(os.getcwd(), 'bridgeHelp.htm')

toolset.registerKernelMenuButton(
    buttonText='Print Current Viewport',
    moduleName='myUtils', functionName='printCurrentVp()',
    author='SIMULIA', version='1.0'
    description='Print current viewport to a PNG file', helpUrl=helpUrl)
```

Abaqus plug-ins

This section describes the plug-ins that are provided with Abaqus/CAE and appear in the **Plug-ins** menu. Select **Plug-ins->About Plug-ins** to display a **View** button that displays help on each plug-in.

In this section:

- *Running the Getting Started with Abaqus examples*
- *Creating a GUI plug-in*
- *Creating a kernel plug-in*
- *Creating dialog boxes with the Really Simple GUI (RSG) Dialog Builder*
- *Upgrading a script*
- *Displaying modal contribution factors*
- *Displaying the history of the adaptive remeshing error indicator*
- *Generating a report of postprocessing results*
- *Exchanging data between Abaqus/CAE and Microsoft Excel*
- *Importing a model from a file in stereolithography (STL) format*
- *Exporting a part or assembly in stereolithography (STL) format*
- *Plotting amplitude data*
- *Combining data from multiple output databases*
- *Finding the nearest node to a point*
- *Finding the average temperature of a set of elements*
- *Creating a planar constraint*
- *Resetting midside nodes*
- *ODB Reducer/Builder*
- *Additive manufacturing*

Running the Getting Started with Abaqus examples

This plug-in allows you to run the examples described in *Getting Started with Abaqus/CAE*. The plug-in creates a model that reproduces each example. The plug-in also creates a job associated with the model; however, the plug-in does not submit the job for analysis. If desired, you can go to the Job module and submit the job for analysis. You can then can go to the Visualization module to view the results of the analysis.

The plug-in fetches files associated with the examples and places those files in the current directory. You can run the following examples from the plug-in:

- *Example: creating a model of an overhead hoist*
- *Example: connecting lug*
- *Example: skew plate*
- *Example: cargo crane*
- *Example: cargo crane under dynamic loading*
- *Example: skew plate*
- *Example: stress wave propagation in a bar*
- *Example: connecting lug*
- *Example: blast loading on a stiffened plate*
- *Example: axisymmetric mount*
- *Example: vibration of a piping system*
- *Abaqus/Standard 3D example: shearing of a lap joint*
- *Abaqus/Explicit example: circuit board drop test*
- *Example: forming a channel in Abaqus/Explicit*

Creating a GUI plug-in

This plug-in illustrates how you can use the Abaqus GUI Toolkit to create GUI plug-ins. The plug-in displays a dialog box that prompts you for the name and the dimensions of a part. When you click **OK**, the plug-in creates the part. The plug-in also creates a toolbox icon that is displayed when you click **Plug-ins->Toolboxes->Examples**.

Creating a kernel plug-in

This plug-in illustrates how you can use the Abaqus Scripting Interface to create a plug-in that executes kernel commands. The plug-in toggles the visibility of the view triad in the current viewport. The plug-in also creates a toolbox icon that is displayed when you click **Plug-ins**->**Toolboxes**->**Examples**.

Creating dialog boxes with the Really Simple GUI (RSG) Dialog Builder

Using the RSG dialog builder is an alternative to using the Abaqus GUI Toolkit commands and a text editor to create dialog boxes.

The **Really Simple GUI (RSG) Dialog Builder** plug-in enables you to create dialog boxes and connect them to kernel commands without writing any code. You select items from a toolbox to add them to an empty dialog box and edit their properties.

The RSG dialog builder provides access to a subset of the commands in the Abaqus GUI Toolkit, but it requires no programming experience to produce a working dialog box. The dialog boxes that you create become new plug-ins to Abaqus/CAE. You can save them as either RSG plug-ins or standard plug-ins, but you can only edit RSG plug-ins with the RSG dialog builder. You must use a text editor to edit standard plug-ins. However, saving an RSG dialog box as a standard plug-in allows an experienced programmer to extend the dialog's functionality by selecting from the complete set of Abaqus GUI Toolkit commands.

The RSG dialog builder plug-in includes a “Five minute tour” that describes and shows some of the basic functions that you can use to build a custom dialog box. The tour icon  is located at the upper left corner of the **GUI** tabbed page in the **RSG Dialog Builder**.

Upgrading a script

The upgrade script plug-in allows you to upgrade a script from a previous release of Abaqus.

In this section:

- *Introduction*
- *Using the upgrade script plug-in*

Introduction

The upgrade script plug-in allows you to upgrade a script from a previous release of Abaqus to the current release of Abaqus. The plug-in allows you to do the following:

- Upgrade across several releases of Abaqus.
- Choose both the release of the original script and the release of the resulting upgraded script.
- Upgrade a single script or all the scripts in a specified directory.
- Preview the changes to your script that resulted from the upgrade process, and, if the changes are acceptable, proceed with the upgrade.

For more information, see *Upgrade script commands*. You can access the plug-in from the **Plug-ins** menu in any module.

Using the upgrade script plug-in

Start the plug-in by selecting **Plug-ins->Abaqus->Upgrade Scripts** from the main menu bar. Enter the following in the **Upgrade Scripts** dialog box:

- Do one of the following to choose the files to upgrade:
  - Choose **Files**, and enter the names of the files to upgrade.
  - Choose **Directory**, and enter the name of the directory to upgrade. Abaqus/CAE upgrades all of the scripts in the directory and any subdirectories.
- Toggle on **Create backups** to keep copies of the original scripts in their original directory.
- Do one of the following to choose the type of the commands to upgrade:
  - Choose **Kernel** to upgrade only Abaqus Scripting Interface commands.
  - Choose **GUI** to upgrade only Abaqus GUI Toolkit commands.
  - Choose **Both** to upgrade both Abaqus Scripting Interface and Abaqus GUI Toolkit commands.

  In most cases you should choose to upgrade both types of commands.
- Enter the name of the log file. The file will contain a log of the upgrade commands that are issued along with a list of the changes to the script and any errors or warnings that are issued. Abaqus/CAE stores the log file in the current directory and displays its contents on the screen during the upgrade process.
- Select the release of the original script and the release to which you would like to upgrade. If you are not sure when the script was created, you should choose the earliest release available. The default release for the original script is three releases before the current release.
- Do one of the following to select the application that displays a preview of the updated script when you click **Preview Changes**:
  - Choose **Use web browser** to preview your changes in the default web browser.
  - Choose **Use specified executable** and enter the path to the executable file that will display the differences between the script and the upgraded script; for example, `windiff`.

From the buttons at the bottom of the dialog box, do the following:

- Click **Preview Changes** to use the selected application to preview the changes to the script.
- If the changes are acceptable, click **Upgrade Scripts** to perform the upgrade and to close the **Upgrade Scripts** dialog box.

Displaying modal contribution factors

The modal contribution factors (MCF) plug-ins are a noise, vibration, and harshness (NVH) application that allows you to compute the output from a modal frequency response analysis and to study the contribution of each mode to the total structural or acoustic response.

In this section:

- *Introduction*
- *Preparing the structural and/or acoustic data*
- *An overview of the MCF plug-in*
- *Computing the modal contribution factors*
- *Displaying modal contribution factors*
- *Ranking criterion for modal contribution factors*

Introduction

When a modal frequency response analysis is performed in Abaqus/Standard, the response is calculated from a linear combination of mode shapes and the corresponding modal amplitudes. For a typical structural-acoustic or structural analysis, the number of modes being studied can number in the hundreds. The designer needs to closely examine these individual modes and find or rank the dominant modes (the modes with major contributions to the total response). The contribution of each mode to the total structural or acoustic response is called the Modal Contribution Factor (MCF).

The NVH application requires two plug-ins that you can access from the **Plug-ins** menu in the Visualization module. The first computes the modal contribution factors, and the second allows you to view the modal contribution factors.

Preparing the structural and/or acoustic data

A typical Abaqus/Standard modal frequency response analysis uses a frequency step followed by multiple steady-state dynamic steps (for multiple load cases) for each base state. The plug-in reads the following variables from the output database generated by the analysis:

- The field output from each frequency step must include the nodal displacement (U). In addition, if you are performing a coupled structural-acoustic analysis, the field output from each frequency step must include the nodal pore or acoustic pressure (POR).
- After each frequency step, there must be at least one modal steady-state dynamic step (not steady-state dynamic, direct or steady-state dynamic, subspace).
- In each modal steady-state dynamic step the history output must include the generalized displacements (GU) and the phase angle of the generalized displacements (GPU) for at least one mode.

You can find sample input files for a modal steady-state dynamic analysis in *Analysis of a rotating fan using substructures and cyclic symmetry* and *Coupled acoustic-structural analysis of a pick-up truck*.

An overview of the MCF plug-in

The modal frequency response, such as the sound pressure or displacement, can be expressed as

$$P(\bar{x};\Omega) = \sum_{\alpha=1}^{N} p_\alpha(\bar{x};\Omega) = \sum_{\alpha=1}^{N} \phi_\alpha(\bar{x})\, q_\alpha(\Omega).$$

The first plug-in uses a script to recover the fluid-structure coupled MCF ($\phi_{\alpha m} q_\alpha$ for each mode α at a response point m for a range of frequencies) based on the mode shapes, ϕ_α, (from the frequency step) and the modal amplitudes, q_α, (from the steady-state dynamic step with modal output requested) in the output database. The plug-in sums these partial pressures and compares them against the Abaqus total. In addition, the script creates a ranking criterion based on a sensitivity analysis.

The second plug-in uses a script that displays polar or vector plots or bar graphs that allow you to identify the most significant modes at each excitation frequency.

Computing the modal contribution factors

The first plug-in computes all the requested quantities (MCFs) and then adds them to the output database as history output. The plug-in is available only from the Visualization module. Enter the Visualization module and start the plug-in by selecting **Plug-ins->NVH->Compute modal Contribution Factors** from the main menu bar.

The plug-in displays the **Open ODB** dialog box, and you enter the name of the output database generated by the Abaqus/Standard analysis described in *Preparing the structural and/or acoustic data*. The plug-in adds the computed values of the modal contribution factors back to the same output database; you must have write access to the output database.

After you enter the output database name, the plug-in displays the **Compute MCF** dialog box, as shown in *Figure 1*:

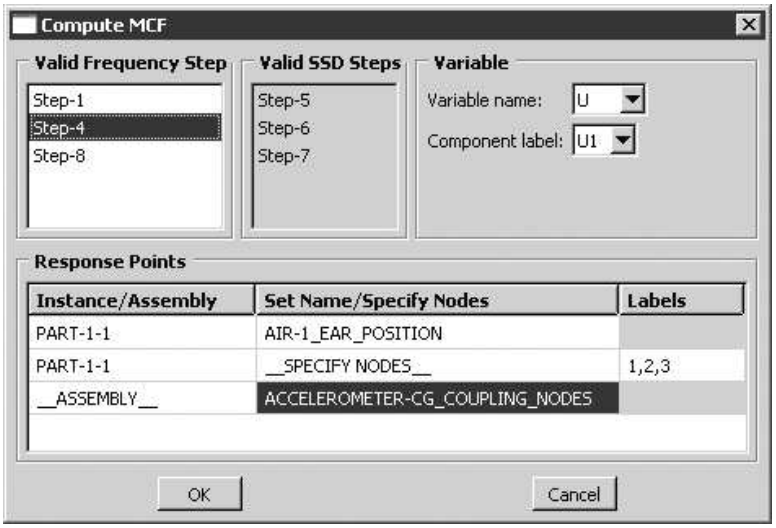


Figure 1: The Compute MCF dialog box.

You must select one of the valid frequency steps from the **Compute MCF** dialog box. A valid frequency step is one that is followed by a modal steady-state dynamic step (not steady-state dynamic, direct or steady-state dynamic, subspace). In addition, a valid frequency step includes GU and GPU output for at least one mode. The valid modal steady-state dynamic steps for each frequency step that you select are shown in the list of **Valid SSD Steps**.

The **Compute MCF** dialog box also asks for the **Response Points**. You can select any combination of the following formats:

- A part instance name followed by a list of node labels
- A part instance name followed by a list of instance-level node sets
- Assembly-level node sets

Finally, you must select the field output variable, which can be any one of the following: acoustic Pressure (POR), displacement (U), velocity (V), or acceleration (A).You can request only U or POR from the frequency step. For all output variables other than POR, you must also request the corresponding component label. For example, if you request U, you must also request either U1, U2, or U3.

After you click **OK** to close the **Compute MCF** dialog box, the plug-in processes all of the valid modal steady-state dynamic steps for the selected frequency step along with all of the specified response points and all of the modes in the frequency step.

Displaying modal contribution factors

After the first plug-in has analyzed the data and written the results back to the output database, you can view the modal contribution factors. Start the second plug-in by selecting **Plug-ins->NVH->Plot modal Contribution Factors** from the main menu bar in the Visualization module. The plug-in displays the **Open ODB** dialog box, and you enter the name of the output database from which to read. The plug-in then displays the **Select Steps** dialog box, as shown in *Figure 1*:

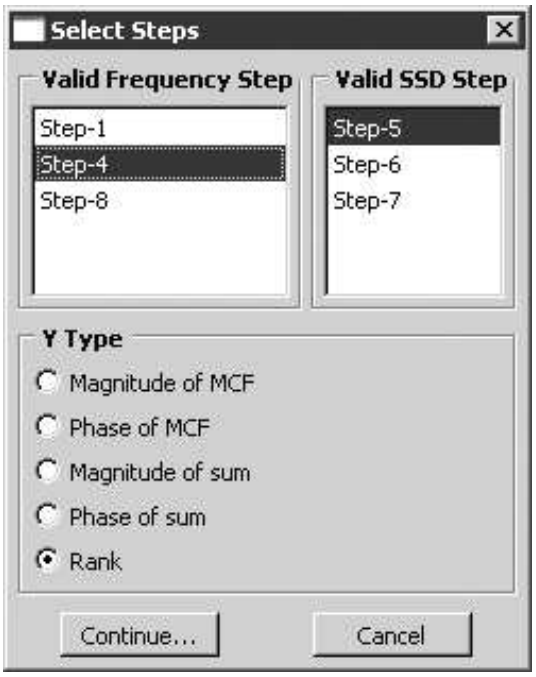


Figure 1: The Select Steps dialog box.

You must select the **Valid Frequency Step** and the **Valid SSD Step** from which you will generate plots. In addition, you must choose one of the following types of variables to plot:

- Magnitude of MCF
- Phase of MCF
- Magnitude of sum
- Phase of sum
- Rank

The magnitude or the phase of the sum indicates the summation of modal contribution factors over all the modes. This is same as the Abaqus output variable (POR, U, V, or A).

When you click **Continue**, the plug-in displays the **Plot MCF** dialog box. Click the **Frequency Step/SSD Step** button to select different frequency and steady-state dynamic step.

Click the **Selection** tab to select the following:

- The response point.
- The frequency of interest, if you are plotting a bar graph, a polar plot, or a vector plot.
- Any number of **Y** components to plot. You add a **Y** component by clicking **>**; you remove a **Y** component by clicking **<**. You can add or remove all of the **Y** components by clicking the **>>** or **<<** buttons respectively.

The **Selection** tabbed page is shown in *Figure 2*:

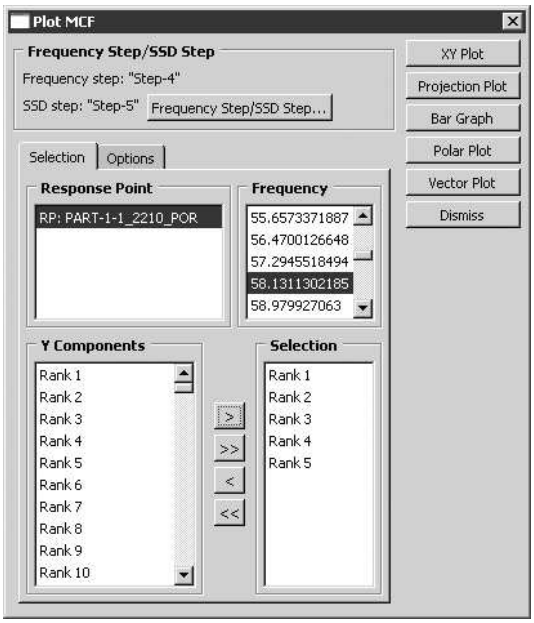


Figure 2: The Selection tabbed page.

Click the **Options** tab to select the general plot options. In addition, if you select POR-based results, you can select the **P Reference** value for XY and projection plots. From the buttons on the right side of the **Plot MCF** dialog box, select the type of plot to create.

- Click **XY Plot** to generate the spectrum for the magnitude or phase of MCF. For mode-ranking selection, the plug-in creates *X–Y* plots for the modes corresponding to the specified ranks.
- Click **Projection Plot** to plot the projection of the selected MCF onto the corresponding total sound pressure level across the spectrum. You can use this plot to view the contribution of a mode (positive or negative) to the total sound pressure level.
- Click **Bar Graph** to plot a bar graph of the MCF of each mode.
- Click **Polar Graph** to plot a polar graph of the MCF of each mode. The polar plot shows both the magnitude and phase information of each individual MCF on the complex plane for each frequency. When plotted with the total sound pressure vector, you can readily identify the modes that dominate the contribution to the total sound pressure.
- Click **Vector Graph** to plot a vector graph of the MCF of each mode. The vector for each mode is drawn in sequence. As more and more modes are plotted, the vector graph tends towards the magnitude of the total sound pressure vector.

The **Options** tabbed page is shown in *Figure 3*:

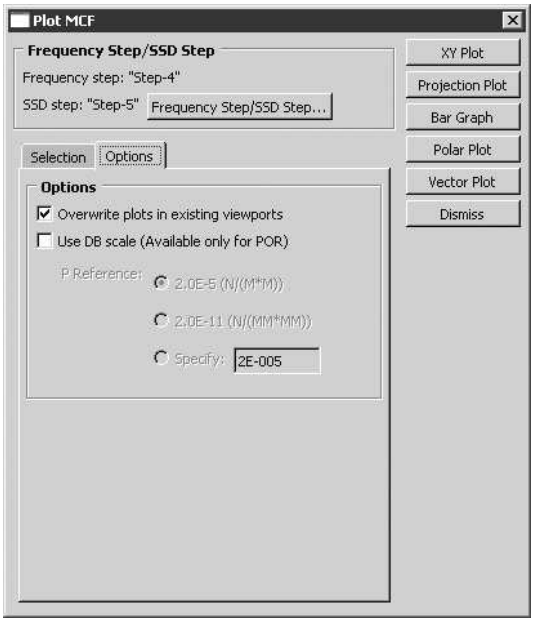


Figure 3: The Options tabbed page.

The plug-in creates each type of plot in a separate viewport and names the viewport accordingly. As a result, you can select **Viewport->Tile Vertically** from the main menu bar to view all the plots at the same time. The following figures were created by the plug-in using the output database generated by a modified version of *Coupled acoustic-structural analysis of a pick-up truck*. The variable selected was POR, and the bar graph, polar graph, and vector graph and the ranks were generated for a frequency of 35 Hz. The analysis examined the first five ranked modes (out of 180):

Figure 4 shows the model that was analyzed. The bar graph indicates that Mode 36 is the significant mode at this frequency.

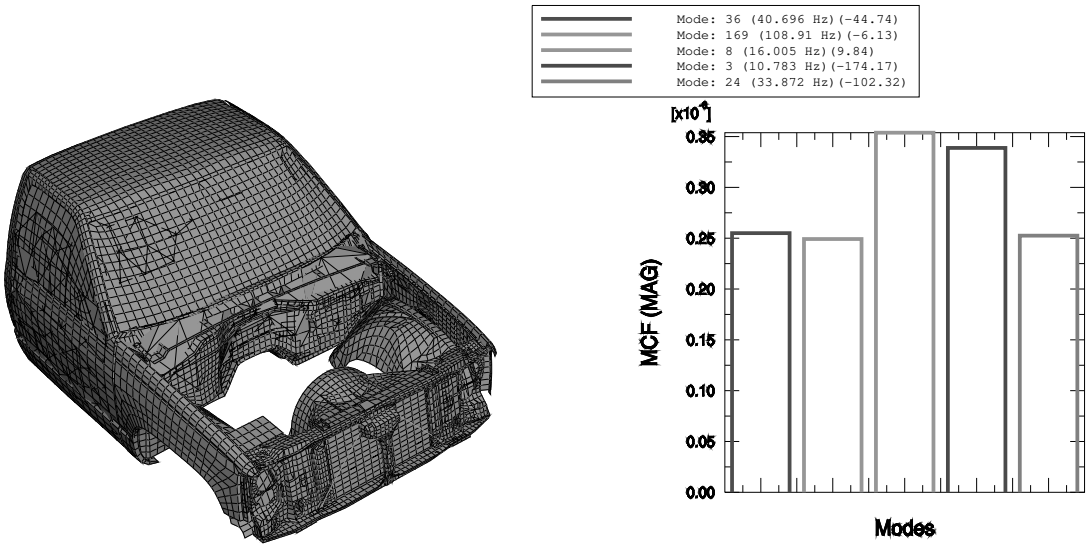


Figure 4: Bar graph of MCF for the coupled acoustic-structural analysis of a pick-up truck.

Figure 5 shows the *X–Y* plot and the projection plot. The *X–Y* plot shows the magnitude of the modal contribution factor at all frequencies. The projection plot shows the projection of the modal contribution factor on the total response at all frequencies. For example, mode 169 becomes more significant at 110 Hz.

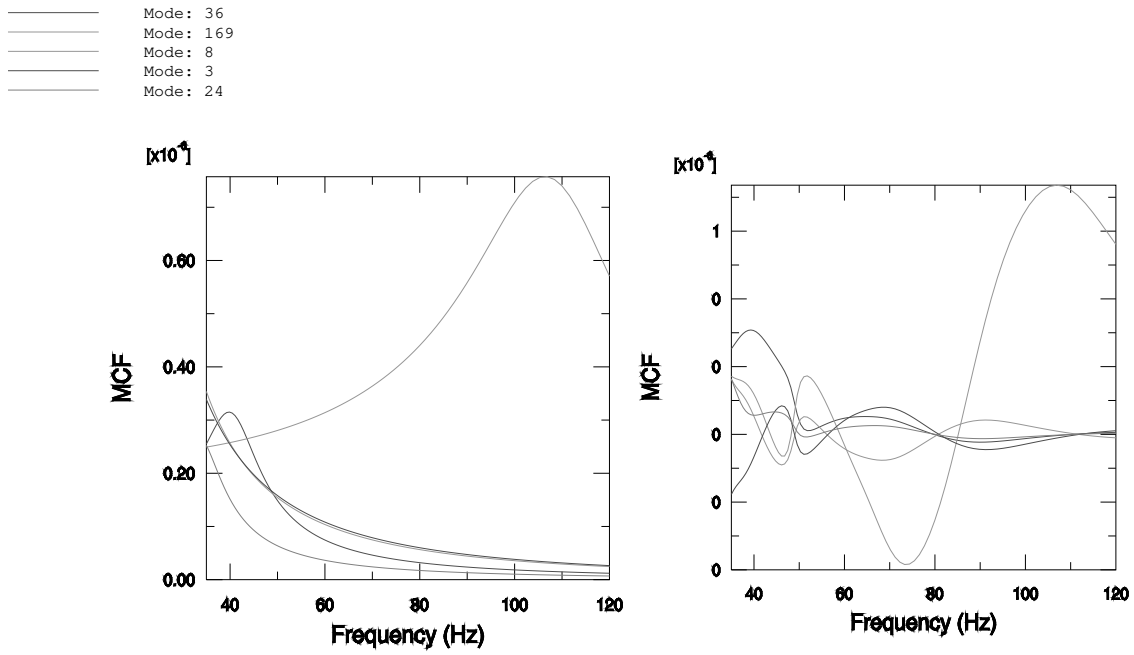


Figure 5: *X–Y* and projection plots of MCF for the coupled acoustic-structural analysis of a pick-up truck.

Figure 6 shows the polar graph and the vector graph. The bar graph shows that the magnitude of mode 36 is less than the magnitude of mode 8; however, the polar plot shows that mode 36 is more in phase with the total response (the vector sum of all the modal contributions). As a result, mode 36 contributes more to the total magnitude than mode 8 at 35 Hz and consequently ranks higher.

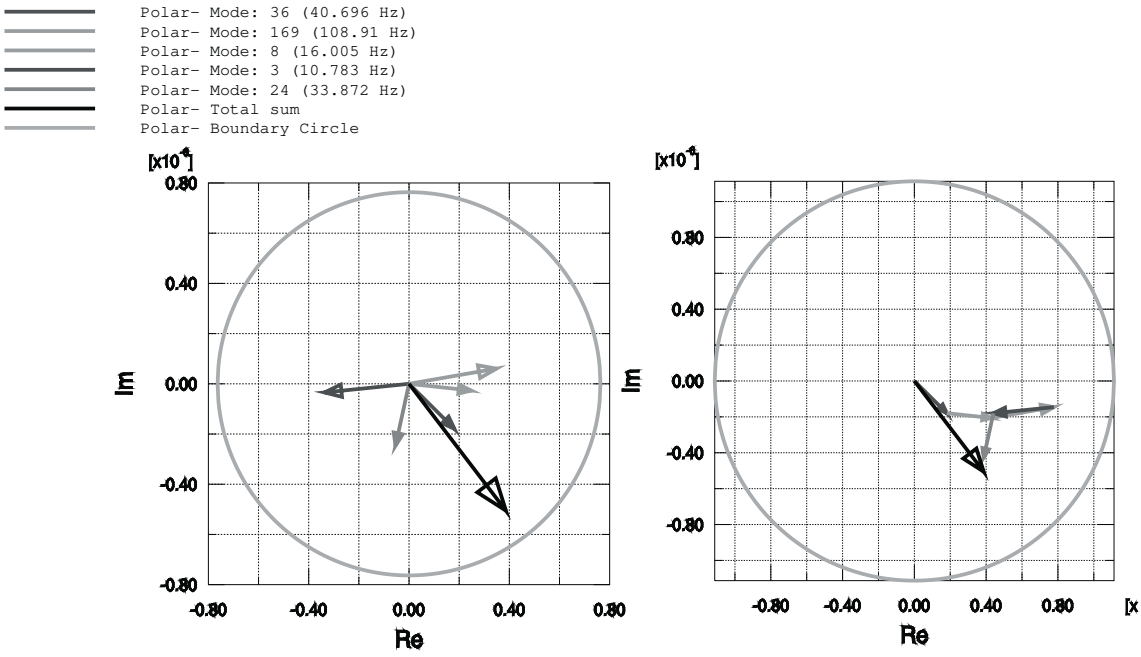


Figure 6: Polar and vector graphs of MCF for the coupled acoustic-structural analysis of a pick-up truck.

Ranking criterion for modal contribution factors

If you select **Rank** as the variable to plot, the plots generated by the plug-in allow you to identify and rank significant modes based on the change in total response magnitude that would result from a fixed percentage increase in the real and imaginary components of the MCF (or partial response). The plug-in implements the ranking criterion as follows:

- α = mode number
- P_α = partial response for mode α
- $P_{\alpha R}$ = real portion of partial response for mode α
- $P_{\alpha I}$ = imaginary portion of partial response for mode α
- $P_{\alpha M}$ = magnitude of partial response for mode α
- $P_{\alpha P}$ = phase of partial response for mode α
- T = total response P
- $TR = \sum_\alpha P_{\alpha R}$ = real portion of total response
- $TI = \sum_\alpha P_{\alpha I}$ = imaginary portion of total response
- $TM = \sqrt{TR^2 + TI^2}$ magnitude of total response
- $TP =$ phase of total response
- $\delta_\alpha = \sqrt{(TR + 0.1^* p_{\alpha R})^2 + (TI + 0.1^* p_{\alpha I})^2} - TM$ = delta in total response magnitude for 10% increase in any given partial response (i.e. sensitivity)

You can determine the top n modes based on the absolute value of δ_α and output $p_{\alpha M}$ and $p_{\alpha P}$ for these modes.

Displaying the history of the adaptive remeshing error indicator

The adaptivity plotter plug-in allows you to review the history of selected error indicators and the element count in a sequence of adaptive remeshing jobs.

In this section:

- *Introduction*
- *Using the adaptivity plotter plug-in*

Introduction

When you execute an adaptivity analysis, Abaqus/CAE submits a succession of Abaqus/Standard jobs as it attempts to reduce the mesh discretization error in your model. This error is characterized by error indicator measures calculated by Abaqus/Standard. Global norms of these indicators are calculated by Abaqus/CAE, and you can use these norms to evaluate the performance and effectiveness of the adaptive remeshing process. For more information, see *Selection of Error Indicators Influencing Adaptive Remeshing*.

The adaptivity plotter plug-in enables you to plot these error indicator norms and to view a graphical representation of the effectiveness of the adaptive remeshing process. You can access the plug-in from the **Plug-ins** menu in any module. The plug-in reads the model database (`.cae`) file associated with the adaptive remeshing jobs; therefore, the file must be available.

Using the adaptivity plotter plug-in

Start the plug-in by selecting **Plug-ins->Tools->Adaptivity Plotter** from the main menu bar. When you start the adaptivity plotter plug-in, it reads the current model database and displays the **Adaptivity Plotter** dialog box. Enter the following in the **Adaptivity Plotter** dialog box:

- Select the desired adaptivity process from the model database.
- Toggle on the remeshing rules that you want to plot. Where applicable, you can choose the following:
  - The error indicators generated by the remeshing rule.
  - The global norms of these error indicators.
- Toggle on **Plot in separate viewport** to create a new viewport for the *X–Y* plot. If you leave this option toggled off, the plug-in will create the *X–Y* plot in the current viewport.
- Toggle on **Show element count** to plot the element count, totaled over all the active remeshing rules.
- Toggle on **Show remeshing rule targets** to plot the target error indicator norm for each rule.
- Click **Plot** to create the *X–Y* plot, or click **Export** to create a table of the error indicator norms in HTML format. An example of an *X–Y* plot created by the plug-in is shown in *Figure 1*.

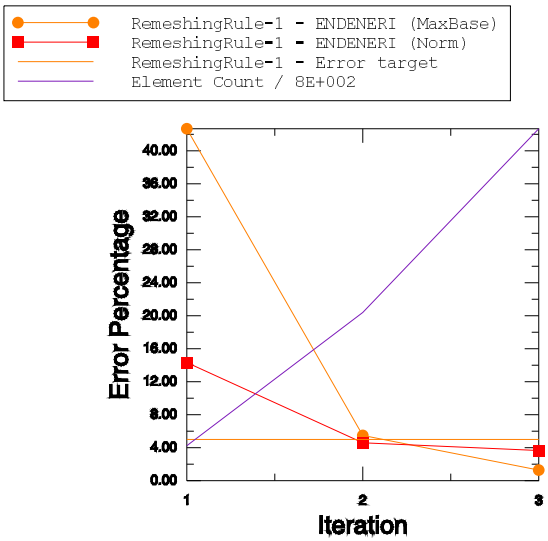


Figure 1: An *X–Y* plot created by the Adaptivity Plotter plug-in.

The following four values are plotted in *Figure 1*:

- The **MaxBase** norm for the error indicator ENDENERI for **RemeshingRule-1**. This norm represents the error indicator value at the location of maximum energy density, normalized by that density.
- The **global Norm** for the error indicator ENDENERI for **RemeshingRule-1**. This norm represents a normalized average of the error indicator over the entire remeshing rule domain.
- The error target, which is the target set when you create the remeshing rule. In this example the target is fixed at 5%.
- The element count, representing the sum of elements in each of the active remeshing rule regions. The *Y*-axis scale is percent, which is used by the plots of error indicator and error target. To determine the error count, you

must multiply the scaling factor indicated in the legend, 800 in this example, by the percent value on the *Y*-axis. For example, the element count at iteration 2 is approximately 20 × 800, or 16000 elements.

Generating a report of postprocessing results

The **Report Generator** plug-in enables you to create an HTML report using the model data and results data in an Abaqus output database (`.odb`) file. You can adjust the settings in the plug-in to select the content that you want to include in the report and to customize the appearance of the report. You can also save your commonly used report settings into a report options file, which can be loaded for use in another session or for another output database.

You can launch the **Report Generator** by selecting **Plug-ins->Tools->Report Generator** from the main menu bar in the Visualization module. Once you select the content that you want to include and click **Generate**, Abaqus/CAE generates the report and opens it automatically in your default browser.

By default, Abaqus/CAE generates report output in the current working directory, but you can specify a different location. When you generate a report, Abaqus/CAE overwrites any report content in the output directory that you select. If you want to retain a report, you should move all of the report files to a secure location where you are not likely to overwrite their contents. All of the files included in an HTML report are listed in the **File Summary** section of the report.

Many of the options in the **Report Generator** control the types of graphical data that you add to the report. You can include the following types of figures in a report:

- 3D XML figures: Toggle on **Include 3DXML** to include a figure in 3D XML format, a file format that enables you to manipulate a three-dimensional rendition of your assembly directly in the report. If your report includes content in 3D XML format, you must use a supported browser and the 3D XML Player to display the report properly. You can download the 3D XML Player from the Dassault Systèmes website at *http://www.3ds.com*.
- Two-dimensional custom views: Toggle on one or more of the **View Options** settings to generate figures that display the assembly from one or more of the predefined views or user-defined views in Abaqus/CAE.
- Additional figures: Use the **Save Current Viewport** plug-in to create a custom, named figure of the current viewport, which you can then include in a report by selecting it from one of the **Additional figures** lists in the **Report Generator** plug-in. Each **Additional figures** list displays only the custom figures that were created in the corresponding module; for example, the **Material Information** section of the report can include additional figures only when they were created in Property module.

  You can launch the **Save Current Viewport** plug-in by selecting **Plug-ins->Tools->Save Current Viewport**. This plug-in is available in the Property module, the Assembly module, the Load module, the Interaction module, and the Visualization module.

The options in the **Report Generator** are organized into tabbed pages that correspond, in general, to the sections in the HTML report. The following options are available:

General

You can select the directory in which the HTML report files will be generated, change the title that appears at the top of the report, and change the user name that appears in the report. You can also display all of the images in the report at full size or display thumbnail images in the report that link to full-sized versions.

Additional options allow you to include or exclude the top-level components of the report, while settings on other pages in this dialog box allow you to select the specific contents to include in each section. For example, toggling on **Include results** on the **General** page prompts Abaqus/CAE to include results data in the HTML report; the options on the **Results** page enable you to control the specific results data included in the report, such as results from individual output variables or selected *X*–*Y* plots.

Materials

You can include figures and a 3D XML–format file that display the assembly with its contents color coded to highlight the material data. Figures can be displayed in any of the default views or user-defined views in

Abaqus/CAE. You can also include any additional viewport images saved in the Property module using the **Save Current Viewport** plug-in.

You can also append a table of materials to the material figures in the HTML report. By default, the table of materials describes the material properties for the materials defined in the assembly and provides a color key to indicate which materials correspond to each color in the figures. You can include full descriptions of each hyperelastic and plastic material in the model; these descriptions are part of the material definition.

Instances

You can include figures and a 3D XML–format file that display the assembly with its contents color coded to highlight the part instances. You select the part instances that you want to display from a list and display the part instances together in a single image or individually in separate images. Figures can be displayed in any of the default views or user-defined views in Abaqus/CAE. You can also include any additional viewport images saved in the Assembly module using the **Save Current Viewport** plug-in.

You can append a table of part instances to the part instance figures in the HTML report. The table of part instances describes the part definition for the instances included in the assembly and provides a color key to indicate which part instances correspond to each color in the figures.

Loads

If you saved any viewport figures in the Load module using the **Save Current Viewport** plug-in, you can include any of those figures in the **Load Information** section of the report.

Interactions

If you saved any viewport figures in the Interaction module using the **Save Current Viewport** plug-in, you can include any of those figures in the **Interaction Information** section of the report.

Results

You can include figures and a 3D XML–format file that display contour plots of the assembly for a single step and increment in the analysis for the field output variables that you select. Figures can be displayed in any of the default views or user-defined views in Abaqus/CAE. You can also include any additional viewport images saved in the Visualization module using the **Save Current Viewport** plug-in. To select the field output variables for which you want to create contour plots, click **Populate**, then select variables and figure options from the dialog box that appears.

You can also include *X*–*Y* plots from the current output database or the current session in the report.

Format

You can customize the appearance of the report by generating the report using a custom style sheet and by displaying an image such as a company logo in the report.

Exchanging data between Abaqus/CAE and Microsoft Excel

The **Excel Utilities** plug-in enables you to import amplitude data or *X*–*Y* data from a Microsoft Excel workbook and to export either type of data from Abaqus/CAE to a Microsoft Excel workbook.



Note:

The **Excel Utilities** plug-in is available only for Windows.

Start the plug-in by selecting **Plug-ins->Tools->Excel Utilities** from the main menu bar. Enter the following in the **Excel Utilities** dialog box:

Object

Choose **XY Data** to transfer an *X*–*Y* data object, or choose **Amplitude** to transfer an amplitude definition.

Transfer Direction

Choose **From Abaqus/CAE to Excel** to export data from Abaqus/CAE to Microsoft Excel, or choose **From Excel to Abaqus/CAE** to import data from a Microsoft Excel spreadsheet into the current session or one of the open model databases.

Options for data import from Excel

To import amplitude data or *X*–*Y* data from Microsoft Excel, specify the name of the Excel file from which you want to import data, select one of the worksheets within the file, and identify the starting location of the data you want to import. By default, the plug-in imports only two columns of data—the data in the column you select and the data in the column immediately to its right—but if you toggle on **Import multiple curves** while importing *X*–*Y* data, the plug-in imports data in pairs of columns, creating an amplitude definition or an *X*–*Y* data object for each pair. If you are importing amplitude data, specify whether the amplitude data object you create should be a tabular or smooth step amplitude.

Amplitude objects are saved into the current model, while *X*–*Y* data objects are made available to the current session.

Options for data export to Excel

To export amplitude data to Microsoft Excel, select the model in which the data resides, then select one or more amplitude definitions and click **OK**. Abaqus/CAE creates a new Excel file and displays the exported data in a worksheet. If you are exporting a single amplitude object, Abaqus/CAE also plots the data in a chart; a chart is not created if you export multiple amplitude objects. Only tabular or smooth step amplitude definitions are available for export.

To export *X*–*Y* data to Microsoft Excel, select one or more *X*–*Y* data objects and click **OK**. Abaqus/CAE creates a new Excel file and displays the exported data in a worksheet. If you export a single *X*–*Y* data object, Abaqus/CAE also plots the data in a chart; a chart is not created if you export multiple *X*–*Y* data objects. You can export any *X*–*Y* data object in your session, including any temporary objects or the data that are currently displayed in the viewport.

Importing a model from a file in stereolithography (STL) format

Select **Plug-ins->Tools->STL Import** from the main menu bar to import a part from an external file in stereolithography (STL) format (`*.stl`) into Abaqus/CAE. The **STL Import** plug-in converts all part definitions and model data into Abaqus format, creating a new Abaqus/CAE model and a new Abaqus input file.

In practice, you might want to add the STL model data that you import to an existing Abaqus/CAE model. To add the STL parts or part instances to an existing Abaqus/CAE model after import, copy these objects from the newly created model to any other Abaqus/CAE model using the **Copy Objects** dialog box. For more information about copying model objects, see *Copying objects between models*.

You can import an STL file saved in ASCII format on Windows or Linux platforms. You can import an STL file saved in binary format only on Windows platforms.

1. From the main menu bar, select **Plug-ins->Tools->STL Import**.

   The **STL Import** dialog box appears.
2. Specify the name of the STL file from which you want to import model data.

   Abaqus/CAE populates the **Model name** field with the name of the STL file you selected.
3. If desired, edit the default model name specified in the **Model name** field for the new Abaqus/CAE model and the new Abaqus input file that will be created.
4. If desired, specify a node merge tolerance value. This tolerance must be positive.
5. Click **OK** to import the model data.

   The plug-in converts the model data in the stereolithography file and creates an Abaqus/CAE model in the current model database and an Abaqus input file in your current working directory. If any STL data cannot be converted, the plug-in records errors or warnings about unconverted data in the log file.

Exporting a part or assembly in stereolithography (STL) format

Select **Plug-ins->Tools->STL Export** from the main menu bar to export a part or assembly from an Abaqus/CAE model to a file in stereolithography (STL) format (`*.stl`). You can export data from either a single part or from the entire assembly, and you can export either the geometry or mesh data from your selection.

To export geometry data, perform the STL export from the Part module or the Assembly module. The plug-in converts any solid parts in your selection to shell geometry during the conversion to STL format. In addition, the plug-in uses the faceted representation of the part to create a mesh for the exported geometry. The accuracy of the generated mesh is dependent on the degree of faceting, which you can control using the **Curve refinement** option. For more information, see *Controlling curve refinement*.

To export mesh data, perform the STL export from the Mesh module or the Visualization module. During the conversion to STL format, the plug-in divides quadratic elements at their midside nodes to create linear elements and splits any quadrilateral elements into triangular elements. When you export assembly data from the Visualization module, the plug-in exports data according to the plot state in the current viewport.

You can export STL files in ASCII format or in binary format.

1. Switch to the Part module or the Mesh module to export part data; or switch to the Assembly module or the Visualization module to assembly data.
2. Display the part or assembly that you want to export in the current viewport.
3. From the main menu bar, select **Plug-ins->Tools->STL Export**.

   The **STL Export** dialog box appears.
4. From the **STL file name** options, specify the name of the STL format file you want to create and the location where you want to create it. If you do not specify a location, the plug-in creates the file in the current working directory.
5. From the **STL file type** options, specify whether you want to create an STL format file in **ASCII** or **Binary** format.
6. Click **OK** to export the model data.

   The plug-in converts the model data in the selected part or assembly and creates a new STL format file in the location that you specified. If any Abaqus data cannot be converted to STL format, the plug-in records errors or warnings about unconverted data in the log file.

Plotting amplitude data

The **Amplitude Plotter** plug-in enables you to create *X–Y* plots that provide a graphical display of the data from one or more of the amplitude definitions in your model. You can plot data from the following types of amplitude objects:

- Tabular
- Equally spaced
- Periodic
- Modulated
- Decay
- Smooth step

For more information about defining and using these amplitude types in Abaqus/CAE, see *The Amplitude toolset*.

You can launch the **Amplitude Plotter** by selecting **Plug-ins->Tools->Amplitude Plotter** from the main menu bar in the Interaction module or the Load module. To plot amplitude data, select one or more amplitude rows in the top half of the dialog box and click **Plot**. Multiple amplitude objects can be included in the same *X–Y* plot if they are of the same amplitude type.

By default, the **Amplitude Plotter** creates an *X–Y* plot that is sized to include all the data in your selected amplitude or amplitudes. You can zoom in on a particular area in the amplitude data when you create your *X–Y* plot by specifying a custom minimum or maximum value using the **Min X Value** or **Max X Value** options. As with any *X–Y* plot in Abaqus/CAE, you can also customize the plot, its axes, or the plot legend by double-clicking these components of the plot; see *Customizing X–Y plot appearance*, for more information about these customization options. The **Amplitude Plotter** also includes a **Display plot legend** option that you can toggle to display or hide the plot legend as you plot amplitude objects.

Toggle on the **Save XY Data** option to create named *X–Y* data objects for the amplitude objects you include in the *X–Y* plot. You can customize the names for the *X–Y* data objects you create; Abaqus/CAE displays these names in the plot legend, and the *X–Y* data objects will be saved as session objects and will be available for use in the Visualization module. The *X–Y* data objects generated by the **Amplitude Plotter** are session-specific and cannot be saved to the model database.

Combining data from multiple output databases

The **Combine ODBs** plug-in enables you to combine model data and results data from two or more Abaqus output database (`.odb`) files into a new output database. You can use this utility to combine any output database files, including those with model data and results from the following:

- Output from different substructures in the same model.
- Output from two different analysis products such as Abaqus/Standard and Abaqus/Explicit, as part of a co-simulation.

If you want to experiment with different combinations of output databases without creating a new combined output database, you can first attempt to display data from different output databases together by using an overlay plot. Data in an overlay plot cannot be saved, but plotting results data from different output databases together can help you decide which data you want to include in a combine operation. For more information about overlay plots, see *Overlaying multiple plots*."

All model data from the selected output databases are included in the output database. However, for results data you can choose to include a subset of the data from the output databases that you specify. Abaqus/CAE determines which results data are included in the combined output database based on two factors: the filtering options you specify and your selection of primary output database.

Filters

You can filter the data that the plug-in includes in the combined output database to include results only from selected steps or frames, from selected output variables, or from a combination of these options. For example, a filter can enable you to include results data only from the last step and the last frame of the specified output databases, and the same filter can dictate that only Mises stress results are included in the combined output database. You can also establish multiple filters if you want to set up different filtering conditions in different steps.

The **Combine ODBs** plug-in also provides two levels of filtering: output database–specific filters, which filter results from only a single output database, and default filters, which apply to the entire job. The output database–specific filters take precedence over the default filters, so Abaqus/CAE employs the settings in the default filters only when the default filter you define does not conflict with filters for one of the individual output databases.

The filtering syntax is flexible enough to allow you to specify multiple steps, frames, or output variable values. You can specify multiple step names in a comma-separated list, such as `Step-1, Step-2, Step-4`. For frames you can include ranges or individual values; for example, entering `1, 3, 5, 7:9` returns frames 1, 3, 5, 7, 8, and 9 to the combined output database.

You can also use the symbolic constants `"ALL"`, `"FIRST"`, and `"LAST"` as shortcuts to specify the data you want to include. These options enable you to include results data from all steps or frames and data from all output variables rather than one or more selected variables.

Primary output database

You designate one output database in every combine operation as the primary output database. **Combine ODBs** first transfers all results data, subject to filtering selections, from the primary output database to the combined output database. The plug-in then locates results data from matching steps and frames in the subsequent output databases and copies only those data into the combined output database. This strategy provides a more coherent structure for the combined results data.



Note:

Each time you combine output databases using the plug-in, Abaqus/CAE records your output database and filtering selections in XML format in a configuration file. You can upload this file into the plug-in for subsequent combine operations.

The **Combine ODBs** plug-in excludes the following types of data from the combined output database:

- History data
- Surface data
- Analytical rigid part instances
- Local coordinate systems associated with field output data

1. From the main menu bar in the Visualization module, select **Plug-ins->Tools->Combine ODBs**.

   The **Combine ODBs** dialog box appears.
2. In the **Job name** field, specify a name for the combine job.
3. If you want to specify your combine options using data in an XML configuration file, do the following:
   a. Click mouse button 3 anywhere in the dialog box, and select **Read from XML File** from the list that appears.

      The **Select XML File** dialog box appears.
   b. Navigate to the XML configuration file you want to use, and click **OK**. For more information, see *Using file selection dialog boxes*.

      Abaqus/CAE populates the table in the dialog box with the output database files and filtering selections specified in the configuration file. You can customize any of these settings further before running the combine operation.
4. Do the following to specify the primary output database:
   a. Click mouse button 3 anywhere in the dialog box, and select **Select Primary ODB** from the list that appears.

      The **Select Primary ODB** dialog box appears.
   b. Navigate to the output database you want to use, and click **OK**. For more information, see *Using file selection dialog boxes*.
5. If you want to add output databases to the combine operation one at a time, do the following:
   a. Click mouse button 3 anywhere in the table, and select **Add ODB** from the list that appears.

      Abaqus/CAE adds a row labeled **ODB** to the table and displays the **Select ODB** dialog box.
   b. Navigate to the output database you want to include, and click **OK**. For more information, see *Using file selection dialog boxes*.

      Abaqus/CAE adds the output database to the selected row.
   c. Repeat these steps for each output database you want to add to the combine operation.
6. If you want to add multiple output databases to the combine operation at once, do the following:
   a. Double-click the **Name** heading.

      The **Select ODBs** dialog box appears.
   b. Navigate to the folder containing the output databases you want to include, select the output database files, and click **OK**. For more information, see *Using file selection dialog boxes*.

      Abaqus/CAE adds the selected output databases in new **ODB** rows to the **Combine ODBs** dialog box.

7. If desired, define filtering options to specify the steps, frames, and output variables to be included in the combined output database.

   - Define a single step, frame, or output variable filter for all output databases:

     1. Double-click the **Step**, **Frame**, or **Variable** heading.
     2. Specify the filtering criteria in the **Set Values** dialog box.

        If you specify the symbolic constants `"ALL"`, `"FIRST"`, and `"LAST"`, you must include the surrounding quotes.
     3. Click **OK**.

     Abaqus/CAE adds the selected criteria to the **Step**, **Frame**, or **Variable** column for the default filters and for every output variable included in the dialog box.

   - Define default filtering options or filtering options for a single output database:

     1. In the **Default** row, **Primary** row, or in one of the **ODB** rows, specify the step, frame, and output variable filters that you want to use as default filters or filters for the selected output database only.
     2. If desired, add more default filters or output database–specific filters by clicking mouse button 3 in the appropriate row, selecting **Add filter** from the menu that appears, and populating the new row with additional filtering criteria.

8. Toggle on **Open the combined ODB after job completion** to display the combined output database in the viewport when the job completes.
9. Click **OK**.

   Abaqus/CAE opens the **Odb Combine Log Message** dialog box, which displays the progress of the combine job and displays any errors that may arise. If any of the output databases you select are from an earlier release, Abaqus/CAE upgrades the output databases, saves the upgraded copy as a new file, and uses the new file in the combine job. If you selected **Open the combined ODB after job completion** and the combine job completes without errors, Abaqus/CAE opens the combined output database in the viewport after the combine job is complete.

Finding the nearest node to a point

The **Find Nearest Node** plug-in lets you locate the node that is nearest to a given point in a meshed model or undeformed plot. You enter the *x*-, *y*-, and *z*-coordinates of any point, and Abaqus/CAE shows you the closest node in your meshed model or undeformed plot. In addition, you can optionally limit the search to a particular region of the model, which is useful in large models.

This plug-in can be used with a model database (`.cae`) file or with the undeformed plot of an output database (`.odb`) file. It cannot be used in a deformed plot.

1. From the main menu bar in the Mesh module or the Visualization module, select **Plug-ins->Tools->Find Nearest Node**.

   The **Find Nearest Node** dialog box appears.

2. Enter the *x*-, *y*-, and *z*-coordinates of the desired point, and click **Find**.

   Abaqus/CAE adds an annotation in the viewport, with an arrow pointing to the nearest node in the mesh. The node number and coordinates are also given in the message area at the bottom of the window; for example:
   ```
   Node 15 of Instance Hinge-solid-1 is nearest to the defined point (1,1,1) Coordinates of node are (0.104999999665,0.999999977648,0.04), distance from defined point is 1.64399665481
   ```

3. If you have a large model and want to restrict the search to a smaller region, click **Select Nodes for Reduced Search**. Follow the prompts to pick any subset of nodes in the viewport, either individually, by angle, or by feature edge. See *Selecting objects within the current viewport*, for instructions. See *Using the angle and feature edge method to select multiple objects*, for details about selecting by angle or feature edge.

   Click **Complete** when you are done selecting nodes.

   Note:

   If there are multiple part instances in the assembly, you may only select nodes for reduced search on one instance.

Finding the average temperature of a set of elements

The **Volume Weighted Average Temperature** plug-in lets you calculate the average temperature for all elements or any subset of elements in an analysis. The elements can be picked in the viewport or selected from a predefined named set. If there are multiple part instances in the assembly, you can select elements on only one instance. Results can be requested at the current time for the model displayed in the viewport or as an *X–Y* time curve over all steps in the analysis.

This plug-in can be used only with an output database (`.odb`) file. The field output variables IVOL and TEMP must be available in the output database for the whole model.

1. From the main menu bar in the Visualization module, select **Plug-ins->Tools->Volume Weighted Average Temperature**.

   The **Volume Weighted Average Temperature** dialog box appears.

2. Choose the time for which the average temperature will be calculated:

   - Choose **Current step/frame** to find the temperature only during the step and frame currently displayed in the viewport.
   - Choose **All steps** to calculate and plot the temperature over the course of the entire analysis.

3. Select the elements in which the average temperature will be calculated. You can select all of the model's elements or a subset.

   - Click **Select From Viewport**, and follow the prompts to pick the elements either individually, by angle, or by feature edge. See *Selecting objects within the current viewport*, for instructions. See *Using the angle and feature edge method to select multiple objects*, for details about selecting by angle or feature edge.
   - Click **Select Element Sets** to choose from a list of predefined element sets.

   Click **Complete** when you are done selecting elements.

4. Calculate the average temperature.

   - Click **Calculate** if you selected **Current step/frame**.

     The result is given in the message area at the bottom of the window, for example:

     ```
     Number of elements picked from viewport: 80 **Volume weighted
     average temperature for the selected elements No. of elements used
     in calculation: 80 Step: Step-1, Frame: 15, Current time: 15.0
     WtAvTemp: 199.883326817 Volume: 1575.0
     ```

   - Click **Plot** if you selected **All steps**.

     A time history *X–Y* plot is displayed in the current viewport. If desired, you can change the default curve name for the history plot.

Creating a planar constraint

The **Plane Remains Plane** plug-in lets you constrain a planar face on a solid body using one or more of the following techniques:

- **Create Planar Constraint**—a constraint is generated that causes the planar face to remain planar throughout an analysis. The plane can rotate and expand, but all nodes on the face will remain in-plane. This constraint is enforced using an analytical rigid surface that is placed in contact with the face with zero friction. A distributed coupling is also created between the analytical rigid surface reference node and all nodes on the face to constrain the rigid body motion of the analytical surface.
- **Create Parallel Planar Constraint**—in addition to the analytical rigid surface constraint described above, a constraint is generated that forces the planar face to remain parallel to its original configuration.
- **Create Beam Element Normal to Plane**—a beam element of any length is generated normal to the planar face and attached to the analytical rigid surface reference node through a kinematic coupling. This technique requires that you first create either of the constraints described above.

The plug-in has the following limitations:

- If picking multiple planes to constrain, you must pick planes that are parallel. There is no automatic check that multiple planes are parallel.
- When applying temperature fields, separate fields must be created for the solid and beam regions.
- The analytical rigid surface will be sized to cover the entire plane in most cases, but there can be cases in which the size is insufficient. You should visually confirm the size of the analytical rigid surface; if the surface is too small, you will have to edit its dimensions in the Part module.
- If the solid body is edited, the constraints must be deleted and redefined.

1. From the main menu bar in the Interaction module, select **Plug-ins->Tools->Plane Remains Plane->Create Planar Constraint** or **Create Parallel Planar Constraint**.
2. Follow the prompts to select the surface that is to remain planar. Select the surfaces either individually, by face angle, or by face curvature. See *Selecting objects within the current viewport*, for instructions.

   Click **Done** when you are finished selecting surfaces.
3. In the **Scaling Factor for ARS Plane** dialog box, enter your desired scaling factor for the analytical rigid surface, and click **OK**.
4. If desired, select **Plug-ins->Tools->Plane Remains Plane->Create Beam Element Normal to Plane** to add a beam element normal to the plane. Follow the prompts to select an analytical rigid surface (to define the beam direction), specify the end point of the beam, and enter the length of the beam.

Resetting midside nodes

Abaqus/CAE automatically projects midside nodes onto the geometry during meshing. In cases where the element quality is adversely affected, you can use the **Reset Midside Nodes** plug-in to reset the midside nodes of quadratic elements to the average position of the edge end nodes. You can define the region containing the nodes to reset by selecting from the geometry, mesh, sets, or surfaces. You can access the plug-in only from the Mesh module.

1. From the main menu bar in the Mesh module, select **Plug-ins->Tools->Reset Midside Nodes**.

   The **Reset Midside Nodes** dialog box appears.

2. Select one of the following methods to define the region containing the nodes to reset:

   - **Pick from viewport**
   - **Select sets**
   - **Select surfaces**

3. If you selected **Pick from viewport**, select regions from the viewport and click **Done**.

   Abaqus/CAE resets the midside nodes of the quadratic elements in the selected region to the average position of the edge end nodes.

4. If you are selecting sets or surfaces, select existing sets or surfaces from the **Region Selection** dialog box that appears, and click **OK**. To limit the lists for easier selection, type a filter pattern into the **Name filter** field, and press the **[Enter]** key to apply the filter.

   Tip: Toggle on **Highlight selections in viewport** to preview the selected sets or surfaces.

   Abaqus/CAE resets the midside nodes of the quadratic elements in the selected region to the average position of the edge end nodes.

5. You can undo and redo multiple changes that you make resetting the midside nodes.
6. Click **Dismiss** to close the **Reset Midside Nodes** dialog box.

ODB Reducer/Builder

The **ODB Reducer/Builder** plug-in provides a method to take portions of an Abaqus output database (`.odb`) file and put them into a new Abaqus output database.
The ability to choose among many options for data to include in a new output database makes this tool useful for a number of applications, including:

- Sharing portions of results files with downstream engineers, other teams, and external partners.
- Decreasing database load times.
- Reducing storage footprint by paring down the output database files to store only certain relevant results.

The plug-in offers control of the data to include in the new output database in several categories.

Mesh

Choose which elements and nodes are included in the new database.

Model features

Choose which set or surface definitions, sections, materials, profiles, and datums to include in the new database.

Results features

Choose which steps and frames (and which field output from those steps and frames) to include in the new database. Choose whether or not history output is included.



Note: Characters such as ':' and ';' are not allowed in step names within the ODB.

1. From the main menu in the Visualization module, select **Plug-ins->Tools->ODB Reducer/Builder**.
   The **ODB Reducer/Builder** dialog box appears.
2. From the **Model Features** frame, toggle off **Sets** to disable the **Sets** tab and exclude all the element, node, and surface set definitions from the resulting database. Toggle on **Sets** to access the options in the **Sets** tab.
3. From the **Model Features** frame, toggle off **Properties** to disable the **Properties** tab and exclude all section, material, and profile data from the resulting database. Toggle on **Properties** to access the options in the **Properties** tab.
4. From the **Model Features** frame, toggle off **Datums** to disable the **Datums** tab and exclude all the datums in the resulting database. Toggle on **Datums** to access the options in the **Datums** tab.
5. From the **Results Features** frame, toggle off **Steps/Frames** to disable the **Steps/Frames** tab and exclude all the steps/frames from the resulting database. Toggle on **Steps/Frames** to access the options in the **Steps/Frames** tab.

   Note: Because field output and history output are dependent on steps/frames, field and history output data will not be available in the resulting output database when **Steps/Frames** is toggled off.

6. From the **Results Features** frame, toggle off **Field outputs** to disable the **Field Outputs** tab and exclude all the field outputs from the resulting database. Toggle on **Field outputs** to access the options in the **Field Outputs** tab.

7. From the **Results Features** frame, toggle off **History outputs** to disable the **History** tab and exclude all the history outputs from the resulting database. Toggle on **History outputs** to include all the history outputs in the resulting database.

    **Note:** There are no filters for history outputs.

8. Use the **Mesh** tab to select which regions of the model to include in the new output database.
   a. Select **Currently displayed** to include areas of the mesh that are currently displayed in the current viewport.
   b. Select **From session file** to load a display group from a file. In the **Session file** text field, use the file selector or type in the name of an `.xml` file and type in the name of the display group in the **Display Group Name** text field.
   c. Select **Bounding box** to include the mesh within or outside a specified box. Specify numerical values for **x Min**, **x Max**, **y Min**, **y Max**, **z Min**, and **z Max** in their corresponding text fields. Choose **Inside** or **Outside** from the list to include the mesh lying within the bounding box or outside it.
   d. Select **Bounding cylinder** to include the mesh within or outside a specified cylinder. Enter three comma-separated numerical values for **Origin point** and **Axis vector** and a single numerical value for **Radius** in their corresponding text fields. Choose **Inside** or **Outside** from the list to include the mesh lying within the bounding cylinder or outside it.
   e. Select **Bounding sphere** to include the mesh within or outside a specified sphere. Enter three comma-separated numerical values for **Origin point** and a single numerical value for **Radius** in their corresponding text fields. Choose **Inside** or **Outside** from the list to include the mesh lying within the bounding sphere or outside it.
   f. Select **Instances**, and use the selection box to include the mesh only from the specified part instances.
   g. Select **Sets**, and use the selection box to include the mesh from the specified sets.

    **Note:** Use the **Mesh Preview Toggle** button to preview the selection.

9. When the **Sets** tab is enabled,
   a. Toggle on **Node sets** to write node set definitions to the new output database. Select **All** to include all the node sets, or use **Select** to specify a list of node sets add to the new database.
   b. Toggle on **Element sets** to write element set definitions to the new output database. Select **All** to include all the element sets, or use **Select** to specify a list of element sets to add to the new database.
   c. Toggle on **Surface sets** to write surface set definitions to the new output database. Select **All** to include all the surface sets, or use **Select** to specify a list of surface sets to add to the new database.

10. When the **Properties** tab is enabled,
    a. Toggle on **Sections** to write section data to the new output database. Select **Keep all data** to retain all the information. **Nullify All** and **Nullify selected** create the empty containers without data. Use these options to create section names without the section properties.
    b. Toggle on **Materials** to write material data to the new output database. Select **Keep all data** to retain all the information. **Nullify All** and **Nullify selected** create the empty containers without data. Use these options to create material names without the material properties.
    c. Toggle on **Profiles** to write profile data to the new output database. Select **Keep all data** to retain all the information. **Nullify All** and **Nullify selected** create the empty containers without data. Use these options to create profile names without the profile properties.

11. When the **Datums** tab is enabled, select **All** to include all the datums, or **Select** a list of datums from the selection box.

12. When the **Steps/Frames** tab is enabled,

a. Select **Match field requests** to use the list of steps and frames from the **Field Outputs** tab.
b. Select **Last frame of last step** to include only the last frame of the last step in the output database.
c. Select **Last frame of each step** to include the last frame of all the steps in the output database.
d. Use **Select below** to activate the **Steps** and **Frames** selection tables and buttons. After selecting the required frames for a particular step, click **Update Frame Selection** to include the selected frame in the output database. Repeat this process for each step. Use **Remove Selected Step** or **Remove All Steps** to exclude all or the selected step from the previously made choices.



Note: A preview of selected steps/frames is shown in the **Current individually selected steps/frames** box below.

13. When the **Field Outputs** tab is enabled,
    a. Select **All fields** to include all the field outputs in the output database.
    b. Use **Select below** to activate the **Steps**, **Frames**, and **Fields** selection tables and buttons. After selecting the required field outputs for a particular step and frame combination, click **Update Output Selection** to include the selected field outputs in the output database. Repeat this process for each step/frame combination. Click **Apply To Step** to include the selected field outputs in all the frames within the selected step. Click **All Steps/Frames** to include the selected field outputs in all the steps and frames. Use **Remove Selected** or **Remove All** under the **Steps** selection table to exclude all field outputs for all the steps or the selected step from the previously made choices. Use **Remove Selected** or **Remove All** under the **Frames** selection table to exclude all field outputs for all the frames or the selected frame from the previously made choices.



Note: A preview of selected field outputs is shown in the **Current individually selected field outputs** box below.

14. If desired, save the settings to a file by clicking **Save Settings** after specifying a filename in the **File** text field in the **Controls** box. You can load the settings later by clicking **Load Settings** after specifying the filename in the **File** text field.
15. Use the **ODB name append** text field to specify the name of the generated output database. The new database is saved to the same location as the currently open output database, using the original file name but with the specified text appended before the `.odb` extension.
16. Click **Generate ODB (and close)** to generate the output database and close the plug-in. Click **Generate ODB (and stay)** to generate the output database and keep the dialog box open.

    Check the Abaqus/CAE message area and the replay file for more information about the output database generation. Abaqus/CAE automatically opens the new output database after the process is completed.

Additive manufacturing

The additive manufacturing plug-in offers a comprehensive and scalable software solution that minimizes the risk of making mistakes when modeling the additive manufacturing process using Abaqus.

You activate the plug-in by selecting **AM Modeler (Show/Hide)** in the **Plug-ins** menu. The **AM Modeler** tab appears with the **Models** and **Results** tabs in the Model Tree area of the main window.

To create an additive manufacturing model, click mouse button 3 and select **Create AM Model** from the menu.

You must select the analysis type and the process type according to the additive manufacturing process you want to simulate. There are two analysis types available: eigenstrain and thermomechanical.

For the eigenstrain analysis type, you can choose between the trajectory-based or pattern-based process type.

For the thermomechanical analysis type, you can choose between four process types:

- Trajectory-based powder bed fabrication
- Pattern-based powder bed fabrication
- Laser direct energy deposition
- Fusion deposition modeling

In this section:

- *Trajectory-based thermomechanical process*
- *Pattern-based thermomechanical process*
- *Trajectory-based eigenstrain process*
- *Pattern-based eigenstrain process*
- *Event series*

Trajectory-based thermomechanical process

You can use the additive manufacturing plug-in to simulate a trajectory-based thermomechanical analysis.

1. In the **Create AM Model** dialog box, select **Thermo-Mechanical** as the analysis type.
2. For the process type, select **Trajectory-based powder bed fabrication**, **Laser direct energy deposition**, or **Fusion deposition modeling**.
3. Specify the thermal and structural models for the simulation, and click **OK**.

   A new AM model appears in the AM Modeler tree. If you expand the node for the AM model, four subnodes display: **Material Input**, **Moving Heat Source**, **Cooling**, and **Event Series**.
4. From the **Event Series** subnode, select **Create/Edit** from the mouse button 3 menu to specify any kind of trajectory or path of the tools (such as laser beam or roller) used in the additive manufacturing process.

   See *Event series* for more information.
5. From the **Material Input** subnode, select **Create** from the mouse button 3 menu to specify all the structural inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values. Enter your desired values, and click **OK**.
6. From the **Moving Heat Source** subnode, select **Create** from the mouse button 3 menu to specify all the thermal inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values. Enter your desired values, and click **OK**.
7. From the **Cooling** subnode, select **Create** from the mouse button 3 menu to specify thermal interactions (such as radiation and convection) on the model.

   Abaqus/CAE populates the field with default values. You can define radiation and convection cooling interaction. Enter your desired values, and click **OK**.

Pattern-based thermomechanical process

You can use the additive manufacturing plug-in to simulate a pattern-based thermomechanical analysis.

1. In the **Create AM Model** dialog box, select **Thermo-Mechanical** as the analysis type.
2. For the process type, select **Pattern-based powder bed fabrication**.
3. Specify the thermal and structural models for the simulation, and click **OK**.

   A new AM model appears in the AM Modeler tree. If you expand the node for the AM model, four subnodes display: **Material Input**, **Moving Heat Source**, **Cooling**, and **Event Series**.
4. From the **Event Series** subnode, select **Create/Edit** from the mouse button 3 menu to specify any kind of trajectory or path of the tools (such as laser beam or roller) used in the additive manufacturing process.

   See *Event series* for more information.
5. From the **Material Input** subnode, select **Create** from the mouse button 3 menu to specify all the structural inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values. Enter your desired values, and click **OK**.
6. From the **Moving Heat Source** subnode, select **Create** from the mouse button 3 menu to specify all the thermal inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values. Enter your desired values, and click **OK**.
7. From the **Cooling** subnode, select **Create** from the mouse button 3 menu to specify thermal interactions (such as radiation and convection) on the model.

   Abaqus/CAE populates the field with default values. You can define radiation and convection cooling interaction. Enter your desired values, and click **OK**.

Trajectory-based eigenstrain process

You can use the additive manufacturing plug-in to simulate a trajectory-based eigenstrain analysis.

1. In the **Create AM Model** dialog box, select **Eigenstrain** as the analysis type.
2. For the process type, select **Trajectory based**.
3. Specify the structural model for the simulation, and click **OK**.

   A new AM model appears in the AM Modeler tree. If you expand the node for the AM model, two subnodes display: **Material Input** and **Event Series**.
4. From the **Event Series** subnode, select **Create/Edit** from the mouse button 3 menu to specify any kind of trajectory or path of the tools (such as laser beam or roller) used in the additive manufacturing process.

   See *Event series* for more information.
5. From the **Material Input** subnode, select **Create** from the mouse button 3 menu to specify all the structural inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values. Enter your desired values, and click **OK**.

Pattern-based eigenstrain process

You can use the additive manufacturing plug-in to simulate a pattern-based eigenstrain analysis.

1. In the **Create AM Model** dialog box, select **Eigenstrain** as the analysis type.
2. For the process type, select **Pattern based**.
3. Specify the structural model for the simulation, and click **OK**.

   A new AM model appears in the AM Modeler tree. If you expand the node for the AM model, the **Material Input** subnode displays.
4. From the **Material Input** subnode, select **Create** from the mouse button 3 menu to specify all the structural inputs for modeling the additive manufacturing process.

   Abaqus/CAE populates the field with default values.
5. Enter your desired values, and click **OK**.

Event series

You can use the additive manufacturing plug-in to define an event series.

1. In the **AM Modeler** tab, select **Event Series->Create/Edit**.
2. Select the model in the **Model** box.
3. Select **Event Series Type**.

   Abaqus/CAE populates some default built-in **Event Series Types** for the additive manufacturing process.
4. **Optional:** You can select **Custom** in the **Event Series Type** box and define your own type.

   A new **Event Series Structure** tab appears. Specify the name of the **Event Series Type**, **fields number**, and **fields names**. The fields described below appear on the new **Event Series Data** tab.
5. Specify the **Event series name**.
6. Specify the time span as either **Step time** or **Total time**.
7. Specify the field values.

   You can specify field values through a `SIM`, `INP`, or `CSV` file or enter the values directly in a table. Select the appropriate entry from the **Data** source list, and specify the file or the table entries.
8. To define additional event series, create additional tabs by clicking the **+** symbol.

   Repeat the above step for each event series.
9. Click **OK** to finish defining the event series.

Appendix

The following sections provide information about the use of keywords, special element types, graphical symbols, and element and output variables in Abaqus/CAE.

In this section:

- *Abaqus keyword browser table*
- *Keyword support from the input file reader*
- *Special element types*
- *Special graphical symbols*
- *Element and output variable support*

Abaqus keyword browser table

Use the following table to determine which Abaqus/CAE module (or toolset) contains the functionality associated with a particular Abaqus keyword. To view documentation for the module (or toolset), click the module (or toolset) name shown in the table. Most currently unsupported keywords can be added to your model using the Keywords Editor.

To see keywords beginning with a particular letter of the alphabet, click that letter in the table below.

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*
N	*O*	*P*	Q	*R*	*S*	*T*	*U*	*V*	*W*	X	Y	Z

Table 38: Keyword browser table.

Keyword	Purpose	Module	Products
**ACOUSTIC CONTRIBUTION*	Request the computation of the acoustic contribution factors for the linear, eigenmode-based, steady-state dynamic procedure.	Unsupported	Abaqus/Standard
**ACOUSTIC FLOW VELOCITY*	Specify flow velocities as a predefined field for acoustic elements.	Unsupported	Abaqus/Standard
**ACOUSTIC MEDIUM*	Specify an acoustic medium.	*Property module*	Abaqus/Standard Abaqus/Explicit
**ACOUSTIC WAVE FORMULATION*	Specify the type of formulation in acoustic problems with incident wave loading.	*Model attribute*	Abaqus/Standard Abaqus/Explicit
**ACTIVATE ELEMENTS*	Activate elements within a step.	Unsupported	Abaqus/Standard
**ADAPTIVE MESH*	Define an adaptive mesh domain.	Supported in the *Step module*; only one adaptive mesh domain can be defined per step.	Abaqus/Standard Abaqus/Explicit
**ADAPTIVE MESH CONSTRAINT*	Specify constraints on the motion of the mesh for an adaptive mesh domain.	Displacement and velocity adaptive mesh constraints are supported in the *Step module*.	Abaqus/Standard Abaqus/Explicit
**ADAPTIVE MESH CONTROLS*	Specify controls for the adaptive meshing and advection algorithms.	*Step module*	Abaqus/Standard Abaqus/Explicit
**ADAPTIVE MESH REFINEMENT*	Activate adaptive mesh refinement in an Eulerian domain.	Unsupported	Abaqus/Explicit
**ADJUST*	Adjust user-specified nodal coordinates to lie on a given surface or to make adjustments based on a user-specified distribution.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**ALLOWABLE STRESS*	Specify the allowable stress of a composite at the initiation of damage in a multiscale material.	Unsupported	Abaqus/Standard
**AMPLITUDE*	Define an amplitude curve.	*Amplitude toolset*; bubble loading is not supported. Similar functionality is available in the *Interaction module*.	Abaqus/Standard Abaqus/Explicit
**ANISOTROPIC HYPERELASTIC*	Specify anisotropic hyperelastic properties for approximately incompressible materials.	*Property module*	Abaqus/Standard Abaqus/Explicit
**ANNEAL*	Anneal the structure.	*Step module*	Abaqus/Explicit
**ANNEAL TEMPERATURE*	Specify material properties for modeling annealing or melting.	*Property module*	Abaqus/Standard Abaqus/Explicit
**AQUA*	Define fluid variables for use in loading immersed beam-type structures.	Unsupported	Abaqus/Aqua
**ASSEMBLY*	Begin an assembly definition.	*Assembly module*	Abaqus/Standard Abaqus/Explicit
**ASYMMETRIC-AXISYMMETRIC*	Define areas of integration for contact elements used with CAXA*n* or SAXA*n* elements.	Unsupported	Abaqus/Standard
**AXIAL*	Used to define the axial behavior of beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**BASE MOTION*	Define the base motion for linear, eigenmode-based, dynamic procedures.	Unsupported	Abaqus/Standard
**BASELINE CORRECTION*	Include baseline correction.	*Amplitude toolset*	Abaqus/Standard
**BEAM ADDED INERTIA*	Define additional beam inertia.	Unsupported	Abaqus/Standard Abaqus/Explicit
**BEAM FLUID INERTIA*	Define additional beam inertia due to immersion in a fluid.	*Property module*	Abaqus/Standard Abaqus/Explicit
**BEAM GENERAL SECTION*	Specify a beam section when numerical integration over the section is not required.	General beam sections with linear response are supported in the *Property module*.	Abaqus/Standard Abaqus/Explicit
**BEAM SECTION*	Specify a beam section when numerical integration over the section is required.	*Property module*	Abaqus/Standard Abaqus/Explicit
**BEAM SECTION GENERATE*	Generate beam section properties for a meshed cross-section.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**BEAM SECTION OFFSET*	Define an offset for the beam cross-section origin.	*Property module*	Abaqus/Standard Abaqus/Explicit
**BIAXIAL TEST DATA*	Used to provide biaxial test data (compression and/or tension).	*Property module*	Abaqus/Standard Abaqus/Explicit
**BLOCKAGE*	Control contacting surfaces for blockage.	Unsupported	Abaqus/Explicit
**BOUNDARY*	Specify boundary conditions.	*Load module*; fluid cavity pressure and generalized plane strain boundary conditions are not supported.	Abaqus/Standard Abaqus/Explicit
**BRITTLE CRACKING*	Define brittle cracking properties.	*Property module*	Abaqus/Explicit
**BRITTLE FAILURE*	Specify brittle failure criterion.	*Property module*	Abaqus/Explicit
**BRITTLE SHEAR*	Define the postcracking shear behavior of a material used with the brittle cracking model.	*Property module*	Abaqus/Explicit
**BUCKLE*	Obtain eigenvalue buckling estimates.	*Step module*	Abaqus/Standard
**BUCKLING ENVELOPE*	Define a nondefault buckling envelope for buckling strut response of frame elements with PIPE sections.	Unsupported	Abaqus/Standard
**BUCKLING LENGTH*	Define buckling length data for buckling strut response of frame elements with PIPE sections.	Unsupported	Abaqus/Standard
**BUCKLING REDUCTION FACTORS*	Define buckling reduction factors for buckling strut response of frame elements with PIPE sections.	Unsupported	Abaqus/Standard
**BULK VISCOSITY*	Modify bulk viscosity parameters.	*Step module*	Abaqus/Explicit
**C ADDED MASS*	Specify concentrated added mass in a *FREQUENCY step.	Unsupported	Abaqus/Aqua
**CAP CREEP*	Specify a cap creep law and material properties.	*Property module*	Abaqus/Standard
**CAP HARDENING*	Specify Drucker-Prager/Cap plasticity hardening.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CAP PLASTICITY*	Specify the Modified Drucker-Prager/Cap plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**CAPACITY*	Define the molar heat capacity at constant pressure for an ideal gas species.	*Interaction module*	Abaqus/Explicit
**CAST IRON COMPRESSION HARDENING*	Specify hardening in compression for the gray cast iron plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CAST IRON PLASTICITY*	Specify plastic material properties for gray cast iron.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CAST IRON TENSION HARDENING*	Specify hardening in tension for the gray cast iron plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CAVITY DEFINITION*	Define a cavity for thermal radiation.	*Interaction module*	Abaqus/Standard
**CECHARGE*	Specify concentrated electric charges in piezoelectric analysis.	*Load module*	Abaqus/Standard
**CECURRENT*	Specify concentrated current in an electric conduction analysis.	*Load module*	Abaqus/Standard
**CENTROID*	Define the position of the centroid of the beam section.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CFILM*	Define film coefficients and associated sink temperatures at one or more nodes or vertices.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CFLOW*	Specify concentrated fluid flow.	*Load module*	Abaqus/Standard
**CFLUX*	Specify concentrated fluxes in heat transfer or mass diffusion analyses.	*Load module*	Abaqus/Standard Abaqus/Explicit
**CHANGE FRICTION*	Change friction properties.	*Interaction module*	Abaqus/Standard
**CHARACTERISTIC LENGTH*	Define characteristic element length at a material point.	Unsupported	Abaqus/Explicit
**CLAY HARDENING*	Specify hardening for the clay plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CLAY PLASTICITY*	Specify the extended Cam-clay plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CLEARANCE*	Specify a particular initial clearance value and a contact direction for the secondary nodes on a surface.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**CLOAD*	Specify concentrated forces and moments.	*Load module*	Abaqus/Standard Abaqus/Explicit Abaqus/Aqua
**CLUSTER MASS INERTIA TABLE*	Specify mass and inertia discrete particle clusters.	Unsupported	Abaqus/Explicit
**COHESIVE BEHAVIOR*	Specify contact cohesive behavior properties.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**COHESIVE SECTION*	Specify element properties for cohesive elements.	*Property module*	Abaqus/Standard Abaqus/Explicit
**COMBINATORIAL RULE*	Define controls associated with combinatorial rules to derive interaction properties from surface properties.	Unsupported	Abaqus/Explicit
**COMBINED TEST DATA*	Specify simultaneously the normalized shear and bulk compliances or relaxation moduli as functions of time.	*Property module*	Abaqus/Standard Abaqus/Explicit
**COMPLEX FREQUENCY*	Extract complex eigenvalues and modal vectors.	*Step module*	Abaqus/Standard
**COMPOSITE MODAL DAMPING*	Specify composite modal damping for modal analyses based on the SIM architecture.	Unsupported	Abaqus/Standard
**CONCENTRATION TENSOR*	Define the concentration tensor for an inclusion in the aggregate.	Unsupported	Abaqus/Standard
**CONCRETE*	Define concrete properties beyond the elastic range.	*Property module*	Abaqus/Standard
**CONCRETE COMPRESSION DAMAGE*	Define compression damage properties for the concrete damaged plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONCRETE COMPRESSION HARDENING*	Define hardening in compression for the concrete damaged plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONCRETE DAMAGED PLASTICITY*	Define flow potential, yield surface, and viscosity parameters for the concrete damaged plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONCRETE FAILURE*	Specify material failure criteria and allow element deletion for the concrete damaged plasticity material model.	Unsupported	Abaqus/Explicit

Keyword	Purpose	Module	Products
**CONCRETE TENSION DAMAGE*	Define postcracking damage properties for the concrete damaged plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONCRETE TENSION STIFFENING*	Define postcracking properties for the concrete damaged plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONDUCTIVITY*	Specify thermal conductivity.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR BEHAVIOR*	Begin the specification of a connector behavior.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR CONSTITUTIVE REFERENCE*	Define reference lengths and angles to be used in specifying connector constitutive behavior.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR DAMAGE EVOLUTION*	Specify connector damage evolution for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR DAMAGE INITIATION*	Specify connector damage initiation criteria for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR DAMPING*	Define connector damping behavior.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR DERIVED COMPONENT*	Specify user-defined components in connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR ELASTICITY*	Define connector elastic behavior.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR FAILURE*	Define a failure criterion for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR FRICTION*	Define friction forces and moments in connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR HARDENING*	Define the plasticity initial yield value and hardening behavior in connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR LOAD*	Specify loads for available components of relative motion in connector elements.	*Load module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR LOCK*	Define a locking criterion for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**CONNECTOR MOTION*	Specify the motion of available components of relative motion in connector elements.	*Load module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR PLASTICITY*	Define plasticity behavior in connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR POTENTIAL*	Specify user-defined potentials in connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR SECTION*	Specify connector attributes for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR STOP*	Specify connector stops for connector elements.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONNECTOR UNIAXIAL BEHAVIOR*	Define uniaxial behavior in connector elements.	*Interaction module*	Abaqus/Explicit
**CONSTITUENT*	Define a constituent in the multiscale material.	Unsupported	Abaqus/Standard Abaqus/Explicit
**CONSTRAINT CONTROLS*	Reset overconstraint checking controls.	Unsupported	Abaqus/Standard
**CONTACT*	Begin the definition of general contact.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT CLEARANCE*	Define contact clearance properties.	Unsupported	Abaqus/Explicit
**CONTACT CLEARANCE ASSIGNMENT*	Assign contact clearances between surfaces in the general contact domain.	Unsupported	Abaqus/Explicit
**CONTACT CONTROLS*	Specify additional controls for contact.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT CONTROLS ASSIGNMENT*	Assign contact controls for the general contact algorithm.	Unsupported	Abaqus/Standard Abaqus/Explicit
**CONTACT DAMPING*	Define viscous damping between contacting surfaces.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT EXCLUSIONS*	Specify self-contact surfaces or surface pairings to exclude from the general contact domain.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT FILE*	Define results file requests for contact variables.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard

Keyword	Purpose	Module	Products
**CONTACT FORMULATION*	Specify a nondefault contact formulation for the general contact algorithm.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT INCLUSIONS*	Specify self-contact surfaces or surface pairings to include in the general contact domain.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT INITIALIZATION ASSIGNMENT*	Assign contact initialization methods for general contact.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT INITIALIZATION DATA*	Define contact initialization methods for general contact.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT INTERFERENCE*	Prescribe time-dependent allowable interferences of contact pairs and contact elements.	*Interaction module*	Abaqus/Standard
**CONTACT MASS SCALING*	Specify contact mass scaling during the step.	*Interaction module*	Abaqus/Explicit
**CONTACT OUTPUT*	Specify contact variables to be written to the output database.	*Step module*	Abaqus/Standard Abaqus/Explicit
**CONTACT PAIR*	Define surfaces that contact each other.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT PERMEABILITY*	Specify fluid permeability contact property.	Unsupported	Abaqus/Standard
**CONTACT PRINT*	Define print requests for contact variables.	Unsupported	Abaqus/Standard
**CONTACT PROPERTY ASSIGNMENT*	Assign contact properties for the general contact algorithm.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CONTACT RESPONSE*	Define contact responses for design sensitivity analysis.	Unsupported	Abaqus/Design
**CONTACT STABILIZATION*	Define contact stabilization controls for general contact.	*Interaction module*	Abaqus/Standard
**CONTOUR INTEGRAL*	Provide contour integral estimates.	*Interaction module*	Abaqus/Standard
**CONTROLS*	Reset solution controls.	*Step module*	Abaqus/Standard
**CONWEP CHARGE PROPERTY*	Define a CONWEP charge for incident waves.	Unsupported	Abaqus/Explicit
**CORRELATION*	Define cross-correlation properties for random response loading.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**CO-SIMULATION*	Identify that the current step is a co-simulation step in Abaqus.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CO-SIMULATION REGION*	Identify the interface regions in the Abaqus model and specify the fields to be exchanged during co-simulation.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**COUPLED CONSTITUTIVE RESPONSE*	This option is used to introduce a user-defined constitutive behavior involving multiple physics fields.	Unsupported	Abaqus/Standard
**COUPLED TEMPERATURE-DISPLACEMENT*	Fully coupled, simultaneous heat transfer and stress analysis.	*Step module*	Abaqus/Standard
**COUPLED THERMAL-ELECTRICAL*	Fully coupled, simultaneous heat transfer and electrical analysis.	*Step module*	Abaqus/Standard
**COUPLED THERMAL-ELECTROCHEMICAL*	Define a coupled thermal-electrochemical analysis.	Unsupported	Abaqus/Standard
**COUPLING*	Define a surface-based coupling constraint.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CRADIATE*	Specify radiation conditions and associated sink temperatures at one or more nodes or vertices.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**CREEP*	Define a creep law.	*Property module*	Abaqus/Standard
**CREEP STRAIN RATE CONTROL*	Control loadings based on the maximum equivalent creep strain rate.	Unsupported	Abaqus/Standard
**CRUSH STRESS*	Define the crush stress of a material.	Unsupported	Abaqus/Explicit
**CRUSH STRESS VELOCITY FACTOR*	Define how the approach velocity at a crushing interface influences a material's resistance to crushing.	Unsupported	Abaqus/Explicit
**CRUSHABLE FOAM*	Specify the crushable foam plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CRUSHABLE FOAM HARDENING*	Specify hardening for the crushable foam plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CURE GLASS TRANSITION TEMPERATURE*	This option is used to specify the glass transition temperature as a function of the degree of cure.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**CURE HEAT GENERATION*	Define the specific heat of the reaction for cure modeling capabilities.	Unsupported	Abaqus/Standard
**CURE KINETICS*	Define the reaction kinetics for cure modeling capabilities.	Unsupported	Abaqus/Standard
**CURE MAX CONVERSION*	Define the maximum value of the degree of conversion for cure modeling capabilities.	Unsupported	Abaqus/Standard
**CURE SHRINKAGE*	Define the cure shrinkage coefficients for cure modeling capabilities.	Unsupported	Abaqus/Standard
**CYCLED PLASTIC*	Specify cycled yield stress data for the *ORNL model.	*Property module*	Abaqus/Standard
**CYCLIC*	Define cyclic symmetry for a cavity radiation heat transfer analysis.	*Interaction module*	Abaqus/Standard
**CYCLIC HARDENING*	Specify the size of the elastic range for the combined hardening model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**CYCLIC SYMMETRY MODEL*	Define the number of sectors and the axis of symmetry for a cyclic symmetric structure.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**D ADDED MASS*	Specify distributed added mass in a *FREQUENCY step.	Unsupported	Abaqus/Aqua
**D EM POTENTIAL*	Specify distributed surface magnetic vector potential.	*Load module*	Abaqus/Standard
**DAMAGE EVOLUTION*	Specify material properties to define the evolution of damage.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DAMAGE INITIATION*	Specify material and contact properties to define the initiation of damage.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DAMAGE STABILIZATION*	Specify viscosity coefficients for the damage model for fiber-reinforced materials, surface-based cohesive behavior or cohesive behavior in enriched elements.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DAMPING*	Specify material damping.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DAMPING CONTROLS*	Specify damping controls.	Supported in the *Step module* only for substructure generation.	Abaqus/Standard

Keyword	Purpose	Module	Products
**DASHPOT*	Define dashpot behavior.	*Property module* and *Interaction module*; supported only for linear behavior independent of field variables. For nonlinear behavior or to include field variables, model connectors in the *Interaction module.*	Abaqus/Standard Abaqus/Explicit
**DEBOND*	Activate crack propagation capability and specify debonding amplitude curve.	Unsupported	Abaqus/Standard
**DECHARGE*	Input distributed electric charges for piezoelectric analysis.	*Load module*	Abaqus/Standard
**DECURRENT*	Specify distributed current densities in an electromagnetic analysis.	*Load module*	Abaqus/Standard
**DEFORMATION PLASTICITY*	Specify the deformation plasticity model.	*Property module*	Abaqus/Standard
**DENSITY*	Specify material mass density.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DEPVAR*	Specify solution-dependent state variables.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DESIGN GRADIENT*	Directly specify design gradients for design sensitivity analysis.	Unsupported	Abaqus/Design
**DESIGN PARAMETER*	Specify design parameters with respect to which sensitivities are calculated.	Unsupported	Abaqus/Design
**DESIGN RESPONSE*	Specify responses for design sensitivity analysis.	Unsupported	Abaqus/Standard Abaqus/Design
**DETONATION POINT*	Define detonation points for a JWL explosive equation of state.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DFLOW*	Specify distributed seepage flows for consolidation analysis.	*Load module*	Abaqus/Standard
**DFLUX*	Specify distributed fluxes in computational fluid dynamics and mass diffusion analyses and specify distributed fluxes and nonuniform concentrated fluxes in heat transfer analyses.	*Load module*	Abaqus/Standard Abaqus/Explicit
**DIAGNOSTICS*	Control diagnostic messages.	Unsupported	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**DIELECTRIC*	Specify dielectric material properties.	*Property module*	Abaqus/Standard
**DIFFUSIVITY*	Specify mass diffusivity or slurry concentration diffusivity.	*Property module*	Abaqus/Standard
**DIRECT CYCLIC*	Obtain the stabilized cyclic response of a structure directly.	*Step module*	Abaqus/Standard
**DISCRETE ELASTICITY*	Specify effective elastic material properties for discrete particles.	Unsupported	Abaqus/Explicit
**DISCRETE SECTION*	Specify element properties for discrete elements.	Unsupported	Abaqus/Explicit
**DISPLAY BODY*	Define a part instance that will be used for display only.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**DISTRIBUTING*	Define a distributing coupling constraint.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**DISTRIBUTING COUPLING*	Specify nodes and weighting for distributing coupling elements.	Unsupported; this option has been superseded by coupling constraints used in conjunction with the distributing option.	Abaqus/Standard
**DISTRIBUTION*	Define spatial distributions.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DISTRIBUTION TABLE*	Define a distribution table.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DLOAD*	Specify distributed loads.	*Load module*	Abaqus/Standard Abaqus/Explicit Abaqus/Aqua
**DOMAIN DECOMPOSITION*	Define a region for domain decomposition and/or define constraints on the domain decomposition.	Unsupported	Abaqus/Explicit
**DRAG CHAIN*	Specify parameters for drag chain elements.	Unsupported	Abaqus/Standard
**DRUCKER PRAGER*	Specify the extended Drucker-Prager plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**DRUCKER PRAGER CREEP*	Specify a Drucker-Prager creep law and material properties.	*Property module*	Abaqus/Standard
**DRUCKER PRAGER HARDENING*	Specify hardening for Drucker-Prager plasticity models.	*Property module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**DSA CONTROLS*	Set DSA solution controls.	Unsupported	Abaqus/Design
**DSECHARGE*	Input distributed electric surface charges for piezoelectric analysis.	*Load module*	Abaqus/Standard
**DSECURRENT*	Specify distributed current densities over a surface in an electromagnetic analysis.	*Load module*	Abaqus/Standard
**DSFLOW*	Specify distributed seepage flows normal to a surface.	*Load module*	Abaqus/Standard
**DSFLUX*	Specify distributed surface fluxes for heat transfer analysis.	*Load module*	Abaqus/Standard Abaqus/Explicit
**DSLOAD*	Specify distributed surface loads.	*Load module*	Abaqus/Standard Abaqus/Explicit
**DYNAMIC*	Dynamic stress/displacement analysis.	*Step module*	Abaqus/Standard Abaqus/Explicit
**DYNAMIC TEMPERATURE-DISPLACEMENT*	Dynamic coupled thermal-stress analysis using explicit integration.	*Step module*	Abaqus/Explicit
**EIGENSTRAIN*	Define your eigenstrain based on different physics.	Unsupported	Abaqus/Standard
**EL FILE*	Define results file requests for element variables.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard Abaqus/Explicit
**EL PRINT*	Define data file requests for element variables.	Unsupported	Abaqus/Standard
**ELASTIC*	Specify elastic material properties.	*Property module*	Abaqus/Standard Abaqus/Explicit
**ELCOPY*	Create elements by copying from an existing element set.	Not applicable; copying portions of sketches and instancing of parts serve similar purposes.	Abaqus/Standard Abaqus/Explicit
**ELECTRIC MACHINE LOAD*	Define electromagnetic forces loads for an electric machine.	Unsupported	Abaqus/Standard
**ELECTRIC MACHINE PROPERTY*	Define electric machine properties (or parameters) needed to apply electric machine loads.	*Property module*	Abaqus/Standard
**ELECTRICAL CONDUCTIVITY*	Specify electrical conductivity.	*Property module*	Abaqus/Standard
**ELECTRICAL RESISTIVITY*	Specify electrical resistivity.	Unsupported	Abaqus/Standard
**ELECTROMAGNETIC*	Electromagnetic response.	*Step module*	Abaqus/Standard

Keyword	Purpose	Module	Products
**ELEMENT*	Define elements by giving their nodes.	*Mesh module*	Abaqus/Standard Abaqus/Explicit
**ELEMENT MATRIX OUTPUT*	Write element stiffness matrices and mass matrices to a file.	Unsupported	Abaqus/Standard
**ELEMENT OPERATOR OUTPUT*	Write element operator output to a SIM document.	Unsupported	Abaqus/Standard
**ELEMENT OUTPUT*	Define output database requests for element variables.	*Step module*	Abaqus/Standard Abaqus/Explicit
**ELEMENT PROGRESSIVE ACTIVATION*	Define the progressive element activation feature and its properties.	Unsupported	Abaqus/Standard
**ELEMENT RECOVERY MATRIX*	Generate modal recovery matrices for a substructure.	Unsupported	Abaqus/Standard
**ELEMENT RESPONSE*	Define element responses for design sensitivity analysis.	Unsupported	Abaqus/Standard Abaqus/Design
**ELEMENT SOLUTION-DEPENDENT VARIABLES*	Allocate space for element solution-dependent variables, and specify optional output names and descriptions for some or all variables allocated.	Unsupported	Abaqus/Standard
**ELEMENT USER OUTPUT VARIABLES*	Specify the number of user-defined element output variables.	Unsupported	Abaqus/Standard
**ELGEN*	Incremental element generation.	Not applicable; elements are generated when you *mesh*the model.	Abaqus/Standard Abaqus/Explicit
**ELSET*	Assign elements to an element set.	*Set toolset*	Abaqus/Standard Abaqus/Explicit
**EMBEDDED ELEMENT*	Specify an element or a group of elements that lie embedded in a group of “host” elements in a model.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**EMISSIVITY*	Specify surface emissivity.	*Interaction module*	Abaqus/Standard
**END ASSEMBLY*	End the definition of an assembly.	*Assembly module*	Abaqus/Standard Abaqus/Explicit
**END INSTANCE*	End the definition of an instance.	*Assembly module* for part instances not imported from a previous analysis; *Load module* for part instances imported from a previous analysis	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**END LOAD CASE*	End the definition of a load case for multiple load case analysis.	*Load module*	Abaqus/Standard
**END PART*	End the definition of a part.	*Part module*	Abaqus/Standard Abaqus/Explicit
**END STEP*	End the definition of a step.	*Step module*	Abaqus/Standard Abaqus/Explicit
**ENERGY FILE*	Write energy output to the results file.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard Abaqus/Explicit
**ENERGY OUTPUT*	Define output database requests for whole model or element set energy data.	*Step module*	Abaqus/Standard Abaqus/Explicit
**ENERGY PRINT*	Print a summary of the total energies.	Unsupported	Abaqus/Standard
**ENRICHMENT*	Specify an enriched feature and the properties of the enrichment.	*Interaction module*	Abaqus/Standard
**ENRICHMENT ACTIVATION*	Activate or deactivate an enriched feature.	*Interaction module*	Abaqus/Standard
**EOS*	Specify an equation of state model.	*Property module*	Abaqus/Explicit
**EOS COMPACTION*	Specify plastic compaction behavior for an equation of state model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**EPJOINT*	Define properties for elastic-plastic joint elements.	Unsupported	Abaqus/Standard
**EQUATION*	Define linear multi-point constraints.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**EQUIVALENT RADIATED SURFACE PROPERTIES*	Specify radiated surface properties.	*Property module*	Abaqus/Standard
**EULERIAN BOUNDARY*	Define inflow and outflow conditions at Eulerian mesh boundaries.	*Load module*	Abaqus/Explicit
**EULERIAN MESH MOTION*	Define the motion of an Eulerian mesh.	*Load module*	Abaqus/Explicit
**EULERIAN SECTION*	Specify element properties for Eulerian elements.	*Property module*	Abaqus/Explicit
**EVENT SERIES*	Define the event series data.	Unsupported	Abaqus/Standard
**EVENT SERIES TYPE*	Define the type of an event series.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**EXPANSION*	Specify thermal or field expansion.	*Property module*	Abaqus/Standard Abaqus/Explicit
**EXTERNAL FIELD*	Import external fields to define distributions, initial conditions, and history-dependent fields or to perform a submodel analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**EXTREME ELEMENT VALUE*	Define element variables to be monitored.	Unsupported	Abaqus/Explicit
**EXTREME NODE VALUE*	Define nodal variables to be monitored.	Unsupported	Abaqus/Explicit
**EXTREME VALUE*	Define element and nodal variables to be monitored.	Unsupported	Abaqus/Explicit
**FABRIC*	Specify the in-plane response of a fabric material.	Unsupported	Abaqus/Explicit
**FAIL STRAIN*	Define parameters for strain-based failure measures.	*Property module*	Abaqus/Standard Abaqus/Explicit
**FAIL STRESS*	Define parameters for stress-based failure measures.	*Property module*	Abaqus/Standard Abaqus/Explicit
**FAILURE RATIOS*	Define the shape of the failure surface for a *CONCRETE model.	*Property module*	Abaqus/Standard
**FASTENER*	Define mesh-independent fasteners.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FASTENER PROPERTY*	Prescribe mesh-independent fastener properties.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FATIGUE*	Define a fatigue crack growth analysis in bulk brittle materials and in brittle material interfaces.	Unsupported	Abaqus/Standard
**FIBER DISPERSION*	Define the second-order orientation tensor of fiber in a composite material.	Unsupported	Abaqus/Standard Abaqus/Explicit
**FIELD*	Specify predefined field variable values.	*Load module*	Abaqus/Standard Abaqus/Explicit
**FIELD IMPORT*	Import external fields to apply over an analysis step in a sequentially coupled multiphysics analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**FIELD MAPPER CONTROLS*	Define controls for the mapper.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
			Abaqus/Explicit
**FIELD OPERATIONS*	Define optional field operations to perform when importing a predefined field.	Unsupported	Abaqus/Standard Abaqus/Explicit
**FILE FORMAT*	Specify format for results file output and invoke zero-increment results file output.	Unsupported; Abaqus/CAE does not use the results file.	Abaqus/Standard
**FILE OUTPUT*	Define output written to the results file.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Explicit
**FILM*	Define film coefficients and associated sink temperatures.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FILM PROPERTY*	Define a film coefficient as a function of temperature and field variables.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FILTER*	Define a filter and/or operator for output filtering and/or operating.	*Filter toolset*	Abaqus/Explicit
**FIXED MASS SCALING*	Specify mass scaling at the beginning of the step.	*Step module*	Abaqus/Explicit
**FLEXIBLE BODY*	Generate a flexible body from an Abaqus/Standard substructure or a natural frequency extraction.	Unsupported	Abaqus/Standard
**FLOW*	Define seepage coefficients and associated sink pore pressures.	Unsupported	Abaqus/Standard
**FLUID BEHAVIOR*	Define fluid behavior for a fluid cavity.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID BULK MODULUS*	Define compressibility for a hydraulic fluid.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID CAVITY*	Define a fluid cavity.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID DENSITY*	Specify hydrostatic fluid density.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID EXCHANGE*	Define fluid exchange.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID EXCHANGE ACTIVATION*	Activate fluid exchange definitions.	*Interaction module*	Abaqus/Explicit

Keyword	Purpose	Module	Products
**FLUID EXCHANGE PROPERTY*	Define the fluid exchange property for flow in or out of a fluid cavity.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID EXPANSION*	Specify the thermal expansion coefficient for a hydraulic fluid.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**FLUID FLUX*	Change the amount of fluid in a fluid-filled cavity.	Unsupported	Abaqus/Standard Abaqus/Explicit
**FLUID ID FIELD*	Associate a fluid identifier with a predefined field variable in a multi-fluid definition.	Unsupported	Abaqus/Standard
**FLUID INFLATOR*	Define a fluid inflator.	*Interaction module*	Abaqus/Explicit
**FLUID INFLATOR ACTIVATION*	Activate fluid inflator definitions.	*Interaction module*	Abaqus/Explicit
**FLUID INFLATOR MIXTURE*	Define gas species used for a fluid inflator.	*Interaction module*	Abaqus/Explicit
**FLUID INFLATOR PROPERTY*	Define a fluid inflator property.	*Interaction module*	Abaqus/Explicit
**FLUID LEAKOFF*	Define fluid leak-off coefficients for pore pressure cohesive elements and enriched elements.	*Property module*	Abaqus/Standard
**FLUID PIPE CONNECTOR LOSS*	Specify fluid pipe connector element loss.	Unsupported	Abaqus/Standard
**FLUID PIPE CONNECTOR SECTION*	Specify fluid pipe connector element section properties.	Unsupported	Abaqus/Standard
**FLUID PIPE CONNECTOR THERMAL LOSS*	Define the method to solve for the convective heat transfer solution in thermal fluid pipe connector elements.	Unsupported	Abaqus/Standard
**FLUID PIPE CONNECTOR WALL*	Specify wall geometry for thermal fluid pipe connector elements.	Unsupported	Abaqus/Standard
**FLUID PIPE FLOW LOSS*	Specify fluid pipe element loss.	Unsupported	Abaqus/Standard
**FLUID PIPE SECTION*	Specify fluid pipe element section properties.	Unsupported	Abaqus/Standard
**FLUID PIPE THERMAL*	Define the method to solve for the convective heat transfer solution in the thermal fluid pipe elements.	Unsupported	Abaqus/Standard
**FLUID PIPE WALL*	Specify wall geometry for thermal fluid pipe elements.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**FOUNDATION*	Prescribe element foundations.	*Interaction module*	Abaqus/Standard
**FRACTURE CRITERION*	Specify crack propagation criteria.	*Property module* and *Interaction module*	Abaqus/Standard Abaqus/Explicit
**FRAME SECTION*	Specify a frame section.	Unsupported	Abaqus/Standard
**FREQUENCY*	Extract natural frequencies and modal vectors.	*Step module*	Abaqus/Standard
**FRICTION*	Specify a friction model.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**GAP*	Specify clearance and local geometry for GAP-type elements.	Unsupported; similar functionality is available by modeling connectors.	Abaqus/Standard
**GAP CONDUCTANCE*	Introduce heat conductance between interface surfaces.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**GAP CONVECTION*	Define Nusselt number for gap convection.	Unsupported	Abaqus/Standard
**GAP DIFFUSIVITY*	Specify ion or species diffusion contact properties.	Unsupported	Abaqus/Standard
**GAP ELECTRICAL CONDUCTANCE*	Specify electrical conductance between surfaces.	Unsupported	Abaqus/Standard
**GAP FLOW*	Define constitutive parameters for tangential flow in pore pressure cohesive elements and enriched elements.	*Property module*	Abaqus/Standard
**GAP HEAT GENERATION*	Introduce heat generation due to energy dissipation at the interface.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**GAP RADIATION*	Introduce heat radiation between surfaces.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**GAS SPECIFIC HEAT*	Define reacted product's specific heat for an ignition and growth equation of state.	*Property module*	Abaqus/Explicit
**GASKET BEHAVIOR*	Begin the specification of a gasket behavior.	*Property module*	Abaqus/Standard
**GASKET CONTACT AREA*	Specify a gasket contact area or contact width for average pressure output.	*Property module*	Abaqus/Standard
**GASKET ELASTICITY*	Specify elastic properties for the membrane and transverse shear behaviors of a gasket.	*Property module*	Abaqus/Standard

Keyword	Purpose	Module	Products
**GASKET SECTION*	Specify element properties for gasket elements.	*Property module*	Abaqus/Standard
**GASKET THICKNESS BEHAVIOR*	Specify a gasket thickness-direction behavior.	*Property module*	Abaqus/Standard
**GEL*	Define a swelling gel.	*Property module*	Abaqus/Standard
**GEOSTATIC*	Obtain a geostatic stress field.	*Step module*	Abaqus/Standard
**GLOBAL DAMPING*	Specify global damping.	Supported in the *Step module* only for substructure generation.	Abaqus/Standard
**GLOBAL RESPONSE*	Specify global design response.	Unsupported	Abaqus/Standard
**HEADING*	Print a heading on the output.	*Job module*	Abaqus/Standard Abaqus/Explicit
**HEAT GENERATION*	Include volumetric heat generation in heat transfer analyses.	*Property module*	Abaqus/Standard Abaqus/Explicit
**HEAT TRANSFER*	Transient or steady-state uncoupled heat transfer analysis.	*Step module*	Abaqus/Standard
**HEATCAP*	Specify a point capacitance.	*Property module* and *Interaction module*.	Abaqus/Standard Abaqus/Explicit
**HOURGLASS STIFFNESS*	Specify nondefault hourglass stiffness.	*Mesh module*	Abaqus/Standard
**HYPERELASTIC*	Specify elastic properties for approximately incompressible elastomers.	*Property module*	Abaqus/Standard Abaqus/Explicit
**HYPERFOAM*	Specify elastic properties for a hyperelastic foam.	*Property module*	Abaqus/Standard Abaqus/Explicit
**HYPOELASTIC*	Specify hypoelastic material properties.	*Property module*	Abaqus/Standard
**HYSTERESIS*	Specify a rate-dependent elastomer model.	*Property module*	Abaqus/Standard
**IMPEDANCE*	Define impedances for acoustic analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**IMPEDANCE PROPERTY*	Define the impedance parameters for an acoustic medium boundary.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**IMPERFECTION*	Introduce geometric imperfections for postbuckling analysis.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**IMPORT*	Import information from a previous Abaqus/Explicit or Abaqus/Standard analysis.	Supported for use in conjunction with part instances; importing selected part instances stored in an output database is supported using the **File** menu and importing the initial state of part instances is supported in the *Load module*.	Abaqus/Standard Abaqus/Explicit
**IMPORT CONTROLS*	Specify the full-model input format, the tie constraint formation configuration, and the tolerance for error checking on shell normals.	Unsupported	Abaqus/Standard Abaqus/Explicit
**IMPORT ELSET*	Import element set definitions from a previous Abaqus/Explicit or Abaqus/Standard analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**IMPORT NSET*	Import node set definitions from a previous Abaqus/Explicit or Abaqus/Standard analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**IMPORT SURFACE*	Import surface definitions from a previous Abaqus/Explicit or Abaqus/Standard analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE*	Define incident wave loading for a blast or scattering load on a boundary.	Unsupported; this option has been superseded by incident wave interactions.	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE FLUID PROPERTY*	Define the fluid properties associated with an incident wave.	Unsupported; this option has been superseded by incident wave interaction properties.	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE INTERACTION*	Define incident wave loading for a blast or scattering load on a surface.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE INTERACTION PROPERTY*	Define the geometric data and fluid properties describing an incident wave.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE PROPERTY*	Define the geometric data describing an incident wave.	Unsupported; this option has been superseded by incident wave interaction properties.	Abaqus/Standard Abaqus/Explicit
**INCIDENT WAVE REFLECTION*	Define the reflection load on a surface caused by incident wave fields.	Unsupported	Abaqus/Standard Abaqus/Explicit
**INCLUDE*	Reference an external file containing Abaqus input data.	Several input data options in Abaqus/CAE provide the capability to reference external files; for example, the material editor can read material properties from an ASCII file.	Abaqus/Standard Abaqus/Explicit
**INCREMENTATION OUTPUT*	Define output database requests for time incrementation data.	*Step module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**INELASTIC HEAT FRACTION*	Define the fraction of the rate of inelastic dissipation that appears as a heat source.	*Property module*	Abaqus/Standard Abaqus/Explicit
**INERTIA RELIEF*	Apply inertia-based load balancing.	*Load module*	Abaqus/Standard
**INITIAL CONDITIONS*	Specify initial conditions for the model.	*Load module*	Abaqus/Standard Abaqus/Explicit Abaqus/Aqua
**INSTANCE*	Begin an instance definition.	*Assembly module* for part instances not imported from a previous analysis; *Load module* for part instances imported from a previous analysis	Abaqus/Standard Abaqus/Explicit
**INTEGRATED OUTPUT*	Specify variables integrated over a surface to be written to the output database.	*Step module*	Abaqus/Standard Abaqus/Explicit
**INTEGRATED OUTPUT SECTION*	Define an integrated output section over a surface with a local coordinate system and a reference point.	*Step module*	Abaqus/Standard Abaqus/Explicit
**INTERFACE*	Define properties for contact elements.	*Property module*; supported for two-dimensional, three-dimensional, and axisymmetric acoustic interface elements. Contact elements are not supported.	Abaqus/Standard
**INTERFACE REACTION*	Introduce an interface reaction between surfaces.	Unsupported	Abaqus/Standard
**INTERNAL FLUX GENERATION*	This option is used to define user-defined internal source terms at the material integration point across multiple physics fields.	Unsupported	Abaqus/Standard
**ISOTROPIZATION PARAMETERS*	Define isotropization parameters for a multiscale material.	Unsupported	Abaqus/Standard
**ITS*	Define properties for ITS elements.	Unsupported; similar functionality (with the exception of friction) is available by modeling connectors.	Abaqus/Standard
**JOINT*	Define properties for JOINTC elements.	Unsupported; similar functionality is available by modeling connectors.	Abaqus/Standard
**JOINT ELASTICITY*	Specify elastic properties for elastic-plastic joint elements.	Unsupported	Abaqus/Standard
**JOINT PLASTICITY*	Specify plastic properties for elastic-plastic joint elements.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**JOINTED MATERIAL*	Specify the jointed material model.	Unsupported	Abaqus/Standard
**JOULE HEAT FRACTION*	Define the fraction of electric energy released as heat.	*Property module*	Abaqus/Standard
**KAPPA*	Specify the material parameters for mass diffusion driven by gradients of temperature and equivalent pressure stress.	*Property module*	Abaqus/Standard
**KINEMATIC*	Define a kinematic coupling constraint.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**KINEMATIC COUPLING*	Constrain all or specific degrees of freedom of a set of nodes to the rigid body motion of a reference node.	Unsupported; this option has been superseded by coupling constraints used in conjunction with the kinematic option.	Abaqus/Standard
**LATENT HEAT*	Specify latent heats.	*Property module*	Abaqus/Standard Abaqus/Explicit
**LOAD CASE*	Begin a load case definition for multiple load case analysis.	*Load module*	Abaqus/Standard
**LOADING DATA*	Provide loading data for uniaxial behavior models in connectors or provide data from a uniaxial or a shear loading test for fabric materials.	Unsupported	Abaqus/Explicit
**LOW DENSITY FOAM*	Specify properties for a low-density foam.	*Property module*	Abaqus/Standard Abaqus/Explicit
**M1*	Define the first bending moment behavior of beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**M2*	Define the second bending moment behavior of beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**MAGNETIC PERMEABILITY*	Specify magnetic permeability.	*Property module*	Abaqus/Standard
**MAGNETOSTATIC*	Magnetostatic analysis.	Unsupported	Abaqus/Standard
**MANIFEST*	Specify the input files to execute in an evolution analysis.	Unsupported	Abaqus/Standard
**MAP SOLUTION*	Map a solution from an old mesh to a new mesh.	Unsupported	Abaqus/Standard
**MASS*	Specify a point mass.	*Property module* and *Interaction module*.	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**MASS ADJUST*	Adjust and/or redistribute the mass of an element set.	Unsupported	Abaqus/Explicit
**MASS DIFFUSION*	Transient or steady-state uncoupled mass diffusion analysis.	*Step module*	Abaqus/Standard
**MASS FLOW RATE*	Specify fluid mass flow rate in a heat transfer analysis.	Unsupported	Abaqus/Standard
**MATERIAL*	Begin the definition of a material.	*Property module*	Abaqus/Standard Abaqus/Explicit
**MATRIX*	Read in the stiffness or mass matrix for a linear user element.	Unsupported	Abaqus/Standard
**MATRIX ASSEMBLE*	Define stiffness, mass, or damping matrices for a part of the model.	Unsupported	Abaqus/Standard
**MATRIX CHECK*	Check the quality of the generated stiffness and mass matrices.	Unsupported	Abaqus/Standard
**MATRIX EXPORT*	Write generated matrices to files in specific formats for selected workflows.	Unsupported	Abaqus/Standard
**MATRIX GENERATE*	Generate global or element matrices.	Unsupported	Abaqus/Standard
**MATRIX INPUT*	Read in a matrix for a part of the model.	Unsupported	Abaqus/Standard
**MATRIX OUTPUT*	Output generated matrices in various forms.	Unsupported	Abaqus/Standard
**MEAN FIELD DAMAGE*	Define the impact of microlevel damage on the behavior of the composite material at the macro level.	Unsupported	Abaqus/Standard Abaqus/Explicit
**MEAN FIELD HOMOGENIZATION*	Begin the definition of a multiscale material modeled with mean-field homogenization.	Unsupported	Abaqus/Standard Abaqus/Explicit
**MEDIA TRANSPORT*	Activate or deactivate a periodic media.	Unsupported	Abaqus/Explicit
**MEMBRANE SECTION*	Specify section properties for membrane elements.	*Property module*	Abaqus/Standard Abaqus/Explicit
**MODAL DAMPING*	Specify damping for modal dynamic analysis.	*Step module*	Abaqus/Standard

Keyword	Purpose	Module	Products
**MODAL DYNAMIC*	Dynamic time history analysis using modal superposition.	*Step module*	Abaqus/Standard
**MODAL FILE*	Write generalized coordinate (modal amplitude) data or eigendata to the results file during a mode-based dynamic or eigenvalue extraction procedure.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard
**MODAL OUTPUT*	Write generalized coordinate (modal amplitude) data to the output database during a mode-based dynamic or complex eigenvalue extraction procedure.	*Step module*	Abaqus/Standard
**MODAL PRINT*	Print generalized coordinate (modal amplitude) data during a mode-based dynamic procedure.	Unsupported	Abaqus/Standard
**MODEL CHANGE*	Remove or reactivate elements and contact pairs.	*Interaction module*	Abaqus/Standard
**MOHR COULOMB*	Specify the Mohr-Coulomb plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**MOHR COULOMB HARDENING*	Specify hardening for the Mohr-Coulomb plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**MOISTURE SWELLING*	Define moisture-driven swelling.	*Property module*	Abaqus/Standard
**MOLECULAR WEIGHT*	Define the molecular weight of an ideal gas species.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**MONITOR*	Define a degree of freedom to monitor.	*Step module*	Abaqus/Standard Abaqus/Explicit
**MOTION*	Specify motions as a predefined field.	Unsupported	Abaqus/Standard
**MPC*	Define multi-point constraints.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**MULLINS EFFECT*	Specify Mullins effect material parameters for elastomers.	*Property module*	Abaqus/Standard Abaqus/Explicit
**MULTIPHYSICS LOAD*	Specify a Butler-Volmer surface load to model solid electrodes.	Unsupported	Abaqus/Standard
**NCOPY*	Create nodes by copying.	Not applicable; copying portions of sketches and instancing of parts serve similar purposes.	Abaqus/Standard Abaqus/Explicit
**NETWORK STIFFNESS RATIO*	Specify stiffness ratios for viscoelastic networks.	Unsupported	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**NFILL*	Fill in nodes in a region.	Not applicable; nodes are generated when you *mesh* the model.	Abaqus/Standard Abaqus/Explicit
**NGEN*	Generate incremental nodes.	Not applicable; nodes are generated when you *mesh*the model.	Abaqus/Standard Abaqus/Explicit
**NMAP*	Map nodes from one coordinate system to another and rotate, translate, or scale the nodal coordinates.	Unsupported; meshing techniques in the *Mesh module* are usually preferable.	Abaqus/Standard Abaqus/Explicit
**NO COMPRESSION*	Introduce a compressive failure theory (tension only materials).	*Property module*	Abaqus/Standard Abaqus/Explicit
**NO TENSION*	Introduce a tension failure theory (compression only material).	*Property module*	Abaqus/Standard Abaqus/Explicit
**NODAL ENERGY RATE*	Define critical energy release rates at nodes.	Unsupported	Abaqus/Standard Abaqus/Explicit
**NODAL THICKNESS*	Define shell or membrane thickness at nodes.	*Property module*	Abaqus/Standard Abaqus/Explicit
**NODE*	Specify nodal coordinates.	*Mesh module*	Abaqus/Standard Abaqus/Explicit
**NODE FILE*	Define results file requests for nodal data.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard Abaqus/Explicit
**NODE OUTPUT*	Define output database requests for nodal data.	*Step module*	Abaqus/Standard Abaqus/Explicit
**NODE PRINT*	Define print requests for nodal variables.	Unsupported	Abaqus/Standard
**NODE RESPONSE*	Define nodal responses for design sensitivity analysis.	Unsupported	Abaqus/Design
**NONLINEAR BH*	Specify nonlinear magnetic behavior of a soft magnetic material.	*Property module*	Abaqus/Standard
**NONSTRUCTURAL MASS*	Specify mass contribution to the model from nonstructural features.	*Property module* and *Interaction module*.	Abaqus/Standard Abaqus/Explicit
**NORMAL*	Specify a particular normal direction.	Unsupported	Abaqus/Standard Abaqus/Explicit
**NSET*	Assign nodes to a node set.	*Set toolset*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**ONE-STEP INVERSE*	Perform a one-step inverse analysis of a sheet metal part.	Unsupported	Abaqus/Standard
**OPERATOR OUTPUT*	Write operator output to a SIM document.	Unsupported	Abaqus/Standard
**ORIENTATION*	Define a local axis system for material or element property definition, for kinematic coupling constraints, for free directions for inertia relief loads, or for connectors.	*Property module*, *Interaction module*, and *Load module*	Abaqus/Standard Abaqus/Explicit
**ORNL*	Specify constitutive model developed by Oak Ridge National Laboratory.	*Property module*	Abaqus/Standard
**OUTPUT*	Define output requests to the output database.	*Step module*	Abaqus/Standard Abaqus/Explicit
**PAPERBOARD HARDENING*	Specify the hardening data of the in-plane paperboard plasticity model.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PAPERBOARD PLASTICITY*	Specify the in-plane paperboard plasticity model.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PAPERBOARD THICKNESS COMPRESSION ELASTIC*	Define the out-of-plane compression elasticity for the paperboard model.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PAPERBOARD THICKNESS COMPRESSION PLASTICITY*	Define the out-of-plane compression yield surface for the paperboard model.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PAPERBOARD TRANSVERSE SHEAR PLASTICITY*	Define the transverse shear yield surface for the paperboard model.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PARAMETER*	Define parameters for input parametrization.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PARAMETER DEPENDENCE*	Define dependence table for tabularly dependent parameters.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PARAMETER SHAPE VARIATION*	Define parametric shape variations.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PARAMETER TABLE*	Define a parameter table.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PARAMETER TABLE TYPE*	Define the type of parameter table.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**PART*	Begin a part definition.	*Part module*	Abaqus/Standard Abaqus/Explicit
**PARTICLE GENERATOR*	Specify a particle generator.	Unsupported	Abaqus/Explicit
**PARTICLE GENERATOR FLOW*	Specify flow speed and mass flow rate per unit inlet area for a particle species.	Unsupported	Abaqus/Explicit
**PARTICLE GENERATOR INLET*	Specify a particle generator inlet surface.	Unsupported	Abaqus/Explicit
**PARTICLE GENERATOR MIXTURE*	Specify particle generator species mixture.	Unsupported	Abaqus/Explicit
**PARTICLE OUTLET*	Specify a particle outlet.	Unsupported	Abaqus/Explicit
**PARTICLE OUTLET FLOW*	Specify nonreflecting or pressure boundary conditions.	Unsupported	Abaqus/Explicit
**PERFECTLY MATCHED LAYER*	Specify perfectly matched layer properties.	Unsupported	Abaqus/Standard
**PERIODIC*	Define periodic symmetry for a cavity radiation heat transfer analysis.	*Interaction module*	Abaqus/Standard
**PERIODIC MEDIA*	Specify a periodic media.	Unsupported	Abaqus/Explicit
**PERMANENT MAGNETIZATION*	Specify permanent magnetization.	Unsupported	Abaqus/Standard
**PERMEABILITY*	Define permeability for pore fluid flow.	*Property module*	Abaqus/Standard
**PHYSICAL CONSTANTS*	Specify physical constants.	*Model attribute*	Abaqus/Standard Abaqus/Explicit
**PIEZOELECTRIC*	Specify piezoelectric material properties.	*Property module*	Abaqus/Standard
**PIEZOELECTRIC DAMPING*	Specify piezoelectric material damping for a piezoelectric material.	Unsupported	Abaqus/Standard
**PIEZORESISTIVITY*	Specify piezoresistivity coefficients.	Unsupported	Abaqus/Standard
**PIPE-SOIL INTERACTION*	Specify element properties for pipe-soil interaction elements.	Unsupported	Abaqus/Standard
**PIPE-SOIL STIFFNESS*	Define constitutive behavior for pipe-soil interaction elements.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**PLANAR TEST DATA*	Used to provide planar test (or pure shear) data (compression and/or tension).	*Property module*	Abaqus/Standard Abaqus/Explicit
**PLASTIC*	Specify a metal plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**PLASTIC AXIAL*	Define plastic axial force for frame elements.	Unsupported	Abaqus/Standard
**PLASTIC M1*	Define the first plastic bending moment behavior for frame elements.	Unsupported	Abaqus/Standard
**PLASTIC M2*	Define the second plastic bending moment behavior for frame elements.	Unsupported	Abaqus/Standard
**PLASTIC TORQUE*	Define the plastic torsional moment behavior for frame elements.	Unsupported	Abaqus/Standard
**PLASTICITY CORRECTION*	Specify the plastic response used to compute Neuber and Glinka plasticity corrections.	*Property module*	Abaqus/Standard
**PLY FABRIC FAILURE*	Defines the conditions that trigger material failure for the ply fabric damage model for bidirectional fabric-reinforced composites.	Unsupported	Abaqus/Explicit
**PLY FABRIC HARDENING*	Specify hardening for the in-plane shear plasticity response of bidirectional fabric-reinforced composite materials.	Unsupported	Abaqus/Explicit
**PML COEFFICIENT*	Specify perfectly matched layer coefficients.	Unsupported	Abaqus/Standard
**PORE FLUID PRESSURE*	Specify a known pore fluid pressure field in a static or in an explicit dynamic stress analysis.	Unsupported	Abaqus/Standard Abaqus/Explicit
**POROUS BULK MODULI*	Define bulk moduli for soils and rocks.	*Property module*	Abaqus/Standard Abaqus/Explicit
**POROUS ELASTIC*	Specify elastic material properties for porous materials.	*Property module*	Abaqus/Standard
**POROUS ELECTRODE THEORY*	Activate use of the electrochemistry framework to model a porous electrode.	Unsupported	Abaqus/Standard
**POROUS FAILURE CRITERIA*	Define porous material failure criteria for a *POROUS METAL PLASTICITY model.	*Property module*	Abaqus/Explicit

Keyword	Purpose	Module	Products
**POROUS METAL PLASTICITY*	Specify a porous metal plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**POST OUTPUT*	Postprocess for output from the restart file.	Unsupported	Abaqus/Standard
**POTENTIAL*	Define the stress potential of a yield/creep model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**PREPRINT*	Select printout for the analysis input file processor.	*Job module*	Abaqus/Standard Abaqus/Explicit
**PRESSURE PENETRATION*	Specify pressure penetration loads with surface-based contact.	*Interaction module*	Abaqus/Standard
**PRESSURE STRESS*	Specify equivalent pressure stress as a predefined field for a mass diffusion analysis.	Unsupported	Abaqus/Standard
**PRESTRESS HOLD*	Keep rebar prestress constant during initial equilibrium solution.	Unsupported	Abaqus/Standard
**PRE-TENSION SECTION*	Associate a pre-tension node with a pre-tension section.	*Load module*	Abaqus/Standard
**PRINT*	Request or suppress output to the message file in an Abaqus/Standard analysis or to the status file in an Abaqus/Explicit analysis.	*Step module*	Abaqus/Standard Abaqus/Explicit
**PROBABILITY DENSITY FUNCTION*	Specify a probability density function.	Unsupported	Abaqus/Explicit
**PROPERTY TABLE*	Define a property table.	Unsupported	Abaqus/Standard Abaqus/Explicit
**PROPERTY TABLE TYPE*	Define the type of property table.	Unsupported	Abaqus/Standard
**PSD-DEFINITION*	Define a cross-spectral density frequency function for random response loading.	Unsupported	Abaqus/Standard
**RADIATE*	Specify radiation conditions in heat transfer analyses.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**RADIATION FILE*	Define results file requests for cavity radiation heat transfer.	Unsupported; Abaqus/CAE reads output from the output database file only.	Abaqus/Standard
**RADIATION OUTPUT*	Define output database requests for cavity radiation variables.	*Step module*	Abaqus/Standard

Keyword	Purpose	Module	Products
**RADIATION PRINT*	Define print requests for cavity radiation heat transfer.	Unsupported	Abaqus/Standard
**RADIATION SYMMETRY*	Define cavity symmetries for radiation heat transfer analysis.	*Interaction module*	Abaqus/Standard
**RADIATION VIEW FACTOR*	Control cavity radiation and view factor calculations.	*Interaction module*	Abaqus/Standard
**RANDOM RESPONSE*	Calculate response to random loading.	*Step module*	Abaqus/Standard
**RATE DEPENDENT*	Define a rate-dependent viscoplastic model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**RATIOS*	Define anisotropic swelling.	*Property module*	Abaqus/Standard
**REACTION RATE*	Define the reaction rate for an ignition and growth equation of state.	*Property module*	Abaqus/Explicit
**REBAR*	Define rebar as an element property.	Unsupported	Abaqus/Standard Abaqus/Explicit
**REBAR LAYER*	Define layers of reinforcement in membrane, shell, surface, and continuum elements.	*Property module*; supported only for membrane and shell elements.	Abaqus/Standard Abaqus/Explicit
**REBAR LINE*	Define rebar reinforcement in beam elements.	Unsupported	Abaqus/Standard Abaqus/Explicit
**REDUCED BASIS GENERATE*	Reduced basis generation analysis.	Unsupported	Abaqus/Standard
**REFLECTION*	Define reflection symmetries for a cavity radiation heat transfer analysis.	*Interaction module*	Abaqus/Standard
**RELEASE*	Release rotational degrees of freedom at one or both ends of a beam element.	Unsupported	Abaqus/Standard
**RESPONSE SPECTRUM*	Calculate the response based on user-supplied response spectra.	*Step module*	Abaqus/Standard
**RESTART*	Save and reuse data and analysis results.	*Step module* for saving restart data; *Job module* for performing a restart analysis	Abaqus/Standard Abaqus/Explicit
**RETAINED NODAL DOFS*	Specify the degrees of freedom to retain as external to a substructure.	*Load module*	Abaqus/Standard

Keyword	Purpose	Module	Products
**RIGID BODY*	Define a set of elements as a rigid body and define rigid element properties.	*Part module* and *Interaction module*	Abaqus/Standard Abaqus/Explicit
**RIGID SURFACE*	Define an analytical rigid surface.	*Part module*	Abaqus/Standard
**ROTARY INERTIA*	Define rigid body rotary inertia.	*Property module* and *Interaction module*.	Abaqus/Standard Abaqus/Explicit
**SCALE MASS*	Specify scale factors and, for adjoint design sensitivity analysis, quantities that represent the derivatives of the scale factors divided by the scale factors for material mass.	Unsupported	Abaqus/Standard
**SCALE STIFFNESS*	Specify scale factors and, for adjoint design sensitivity analysis, quantities that represent the derivatives of the scale factors divided by the scale factors for material stiffness.	Unsupported	Abaqus/Standard
**SCALE STRESS DESIGN*	Specify scale factors and quantities that represent the derivatives of the scale factors divided by the scale factors for material stress design response.	Unsupported	Abaqus/Standard
**SCALE THERMAL CONDUCTIVITY*	Specify scale factors for thermal conductivity.	Unsupported	Abaqus/Standard
**SECTION CONTROLS*	Specify section controls.	*Mesh module*	Abaqus/Standard Abaqus/Explicit
**SECTION CURVATURE*	Define the precurvatures and the pretorsion of the beam centerline for 3-DOF warping elements.	Unsupported	Abaqus/Standard
**SECTION FILE*	Define results file requests of accumulated quantities on user-defined surface sections.	Unsupported	Abaqus/Standard
**SECTION INERTIA*	Define inertia properties for meshed general beam sections.	Unsupported	Abaqus/Standard Abaqus/Explicit
**SECTION ORIGIN*	Define a meshed cross-section origin.	Unsupported	Abaqus/Standard
**SECTION POINTS*	Locate points in the beam section for which stress and strain output are required.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SECTION PRINT*	Define print requests of accumulated quantities on user-defined surface sections.	Unsupported	Abaqus/Standard

Keyword	Purpose	Module	Products
**SECTION STIFFNESS*	Define stiffness properties for meshed general beam sections.	Unsupported	Abaqus/Standard Abaqus/Explicit
**SECTION TANGENT*	Define the inclination of the tangent of the beam centerline with respect to the cross-section normal direction for 3-DOF warping elements.	Unsupported	Abaqus/Standard
**SELECT CYCLIC SYMMETRY MODES*	Specify the cyclic symmetry modes in an eigenvalue analysis of a cyclic symmetric structure.	*Interaction module*	Abaqus/Standard
**SELECT EIGENMODES*	Select the modes to be used in a modal dynamic, complex eigenvalue extraction, or substructure generation analysis.	Supported in the *Step module* only for substructure generation.	Abaqus/Standard
**SFILM*	Define film coefficients and associated sink temperatures over a surface.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**SFLOW*	Define seepage coefficients and associated sink pore pressures normal to a surface.	Unsupported	Abaqus/Standard
**SHEAR CENTER*	Define the position of the shear center of a beam section.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SHEAR FAILURE*	Specify a shear failure model and criterion.	Unsupported	Abaqus/Explicit
**SHEAR RETENTION*	Define the reduction of the shear modulus associated with crack surfaces in a *CONCRETE model as a function of the tensile strain across the crack.	*Property module*	Abaqus/Standard
**SHEAR TEST DATA*	Used to provide shear test data.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SHELL GENERAL SECTION*	Define a general, arbitrary, elastic shell section.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SHELL SECTION*	Specify a shell cross-section.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SHELL TO SOLID COUPLING*	Define a surface-based coupling between a shell edge and a solid face.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**SIMPEDANCE*	Define impedances of acoustic surfaces.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**SIMPLE SHEAR TEST DATA*	Used to provide simple shear test data.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SLIDE LINE*	Specify slide line surfaces on which deformable structures may interact.	Unsupported; the *Interaction module* uses surface-based contact.	Abaqus/Standard
**SLOAD*	Apply loads to a substructure.	Unsupported	Abaqus/Standard
**SOFT ROCK HARDENING*	Specify hardening for the soft rock plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SOFT ROCK PLASTICITY*	Specify the soft rock plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SOFTENING REGULARIZATION*	Specify softening regularization for the clay plasticity model and the soft rock plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SOILS*	Effective stress analysis for fluid-filled porous media.	*Step module*	Abaqus/Standard
**SOLID ELECTROLYTE THEORY*	Activate use of the electrochemistry framework to model a solid electrolyte.	Unsupported	Abaqus/Standard
**SOLID SECTION*	Specify element properties for solid, infinite, acoustic, particle, and truss elements.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SOLUBILITY*	Specify solubility.	*Property module*	Abaqus/Standard
**SOLUTION TECHNIQUE*	Specify alternative solution methods.	*Step module*	Abaqus/Standard
**SOLVER CONTROLS*	Specify controls for the iterative and direct linear solvers.	*Step module*	Abaqus/Standard
**SORPTION*	Define absorption and exsorption behavior.	*Property module*	Abaqus/Standard
**SPECIFIC HEAT*	Define specific heat.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SPECTRUM*	Define or create a response spectrum.	*Amplitude toolset*; supported only for defining a spectrum.	Abaqus/Standard
**SPH SURFACE BEHAVIOR*	Define boundary surface interaction properties between SPH particles and Lagrangian surfaces.	Unsupported	Abaqus/Explicit
**SPRING*	Define spring behavior.	*Property module* and *Interaction module*; supported only for linear behavior independent of field	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
		variables. For nonlinear behavior or to include field variables, model connectors in the *Interaction module.*	
**SRADIATE*	Specify surface radiation conditions in heat transfer analysis.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**STATIC*	Static stress/displacement analysis.	*Step module*	Abaqus/Standard
**STEADY STATE CRITERIA*	Specify steady-state criteria for terminating a quasi-static unidirectional simulation.	Unsupported	Abaqus/Explicit
**STEADY STATE DETECTION*	Specify steady-state requirements for terminating a quasi-static unidirectional simulation.	Unsupported	Abaqus/Explicit
**STEADY STATE DYNAMICS*	Steady-state dynamic response based on harmonic excitation.	*Step module*	Abaqus/Standard
**STEADY STATE TRANSPORT*	Steady-state transport analysis.	Unsupported	Abaqus/Standard
**STEP*	Begin a step.	*Step module*	Abaqus/Standard Abaqus/Explicit
**STEP CONTROL*	Control progress of the simulation based on the solution state.	Unsupported	Abaqus/Standard
**STEP CYCLING*	Define a step or series of steps to be repeated.	Unsupported	Abaqus/Standard
**STEP CYCLING CONTROL*	Specify sensor-based or contact wear-based criteria to control early termination of step cycling.	Unsupported	Abaqus/Standard
**SUBCYCLING*	Define a subcycling zone.	Unsupported	Abaqus/Explicit
**SUBMODEL*	Specify driven boundary nodes in submodeling analysis.	*Load module* and *model attribute*	Abaqus/Standard Abaqus/Explicit
**SUBMODEL CONDITIONS*	Specify that submodel conditions are to be imposed on cut surfaces during this step.	Unsupported	Abaqus/Standard
**SUBMODEL CUT*	Define a submodel cut surface.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE CHANGE*	Change substructure properties for the current analysis step.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE DAMPING*	Specify mass proportional and stiffness proportional damping for a substructure.	Unsupported	Abaqus/Standard Abaqus/AMS

Keyword	Purpose	Module	Products
**SUBSTRUCTURE DAMPING CONTROLS*	Specify damping controls for a substructure property.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE GENERATE*	Substructure generation analysis.	*Step module*	Abaqus/Standard
**SUBSTRUCTURE LOAD CASE*	Begin the definition of a substructure load case.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE MATRIX OUTPUT*	Write a substructure's recovery matrix, reduced stiffness matrix, mass matrix, load case vectors, and gravity load vectors to a file.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE MODAL DAMPING*	Specify damping for a substructure property.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE OUTPUT*	Output the solution in the substructure usage analysis.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE PATH*	Enter into a substructure to obtain output or return back from a previously entered substructure.	Unsupported	Abaqus/Standard
**SUBSTRUCTURE PROPERTY*	Translate, rotate, and/or reflect substructures.	Unsupported	Abaqus/Standard
**SUPERELASTIC*	Define the data for the superelastic model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SUPERELASTIC HARDENING*	Specify hardening data for martensite.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SUPERELASTIC HARDENING MODIFICATIONS*	Specify the stress-strain response of superelastic materials as a function of plastic strain.	*Property module*	Abaqus/Standard Abaqus/Explicit
**SURFACE*	Define a surface or region in a model.	Element-based surfaces are supported by the *Surface toolset*. Node-based surfaces are not supported; if node-based surfaces are imported into Abaqus/CAE, they are treated as sets.	Abaqus/Standard Abaqus/Explicit
**SURFACE BEHAVIOR*	Define alternative pressure-overclosure relationships for contact.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**SURFACE FLAW*	Define geometry of surface flaws.	Unsupported	Abaqus/Standard
**SURFACE INTERACTION*	Define surface interaction properties.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**SURFACE PROPERTY*	Define named surface properties for cavity radiation or contact-based wear.	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**SURFACE PROPERTY ASSIGNMENT*	Assign surface properties to a surface for the general contact algorithm.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**SURFACE SECTION*	Specify section properties for surface elements.	*Property module*	Abaqus/Standard Abaqus/Explicit Abaqus/Aqua
**SURFACE SMOOTHING*	Define surface smoothing methods.	*Interaction module*	Abaqus/Standard
**SURFACE TENSION*	Define the surface tension of an element-based surface.	Unsupported	Abaqus/Explicit
**SURFACE TENSION CONTACT ANGLE*	Specify the surface tension contact angle property for a given contact interaction.	Unsupported	Abaqus/Explicit
**SWELLING*	Specify time-dependent volumetric swelling.	*Property module*	Abaqus/Standard
**SYMMETRIC MODEL GENERATION*	Create a three-dimensional model from an axisymmetric or partial three-dimensional model.	Unsupported	Abaqus/Standard
**SYMMETRIC RESULTS TRANSFER*	Import results from an axisymmetric or partial three-dimensional analysis.	Unsupported	Abaqus/Standard
**SYSTEM*	Specify a local coordinate system in which to define nodes.	Not applicable; instancing a part in the *Assembly module* creates a local coordinate system.	Abaqus/Standard Abaqus/Explicit
**TABLE COLLECTION*	Define a table collection.	Unsupported	Abaqus/Standard Abaqus/Explicit
**TEMPERATURE*	Specify temperature as a predefined field.	*Load module*	Abaqus/Standard Abaqus/Explicit
**TENSILE FAILURE*	Specify a tensile failure model and criterion.	*Property module*	Abaqus/Explicit
**TENSION CUTOFF*	Specify tension cutoff data for the Mohr-Coulomb plasticity model.	*Property module*	Abaqus/Standard Abaqus/Explicit
**TENSION STIFFENING*	Define the retained tensile stress normal to a crack in a *CONCRETE model.	*Property module*	Abaqus/Standard
**THERMAL EXPANSION*	Define the thermal expansion behavior of beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**TIE*	Define surface-based tie and cyclic symmetry constraints or	*Interaction module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
	coupled acoustic-structural interactions.		
**TIME POINTS*	Specify time points at which data are written to the output database file, or specify time points in the loading history at which the response of a structure will be evaluated in a direct cyclic analysis.	*Step module*	Abaqus/Standard Abaqus/Explicit
**TORQUE*	Define the torsional behavior of beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**TORQUE PRINT*	Print a summary of the total torque that can be transmitted across axisymmetric slide lines.	Unsupported	Abaqus/Standard
**TRACER PARTICLE*	Define tracer particles for tracking the location of and results at material points during a step.	Unsupported	Abaqus/Explicit
**TRANSFORM*	Specify a local coordinate system at nodes.	Define nodal coordinate systems for prescribed conditions in the *Load module*.	Abaqus/Standard Abaqus/Explicit
**TRANSPORT VELOCITY*	Specify angular transport velocity.	Unsupported	Abaqus/Standard
**TRANSVERSE SHEAR*	Specify the transverse shear moduli for computing the transverse shear stiffness of shells and beams.	Unsupported	Abaqus/Standard Abaqus/Explicit
**TRANSVERSE SHEAR STIFFNESS*	Define transverse shear stiffness for beams and shells.	*Property module*	Abaqus/Standard Abaqus/Explicit
**TRIAXIAL TEST DATA*	Provide triaxial test data.	*Property module*	Abaqus/Standard Abaqus/Explicit
**TRS*	Used to define temperature-time shift for time history viscoelastic analysis.	*Property module*	Abaqus/Standard Abaqus/Explicit
**UEL PROPERTY*	Define property values to be used with a user element type.	Actuator/sensor interaction properties can be defined in the *Interaction module.*	Abaqus/Standard Abaqus/Explicit
**UNDEX CHARGE PROPERTY*	Define an UNDEX charge for incident waves.	*Interaction module*	Abaqus/Standard Abaqus/Explicit
**UNIAXIAL*	Characterize fabric materials through loading and unloading test data.	Unsupported	Abaqus/Explicit

Keyword	Purpose	Module	Products
**UNIAXIAL TEST DATA*	Used to provide uniaxial test data (compression and/or tension).	*Property module*	Abaqus/Standard Abaqus/Explicit
**UNIFORM*	Define a uniform coupling constraint.	Unsupported	Abaqus/Standard Abaqus/Explicit
**UNIT SYSTEM*	Define a unit system for an Abaqus model for translation to other solvers.	Unsupported	Abaqus/Standard Abaqus/Explicit
**UNLOADING DATA*	Provide unloading data for uniaxial behavior models in connectors or unloading data from uniaxial and shear tests for fabrics.	Unsupported	Abaqus/Explicit
**USER DEFINED FIELD*	Redefine field variables at a material point.	*Property module*	Abaqus/Standard Abaqus/Explicit
**USER ELEMENT*	Introduce a user-defined element type.	Actuator/sensor interactions can be defined in the *Interaction module.*	Abaqus/Standard Abaqus/Explicit
**USER MATERIAL*	Define material constants for use in subroutine UMAT, UMATHT, VUMAT, or VUMATHT.	*Property module*	Abaqus/Standard Abaqus/Explicit
**USER OUTPUT VARIABLES*	Specify number of user variables.	*Property module*	Abaqus/Standard
**VARIABLE MASS SCALING*	Specify mass scaling during the step.	*Step module*	Abaqus/Explicit
**VIEW FACTOR OUTPUT*	Write radiation view factors to the results file in cavity radiation heat transfer analysis.	Unsupported	Abaqus/Standard
**VISCO*	Transient, static, stress/displacement analysis with time-dependent material response (creep, swelling, and viscoelasticity).	*Step module*	Abaqus/Standard
**VISCOELASTIC*	Specify dissipative behavior for use with elasticity.	*Property module*	Abaqus/Standard Abaqus/Explicit
**VISCOSITY*	Specify material shear viscosity.	*Property module*	Abaqus/Standard Abaqus/Explicit
**VISCOUS*	Specify viscous material properties for the two-layer viscoplastic model.	*Property module*	Abaqus/Standard
**VOID NUCLEATION*	Define the nucleation of voids in a porous material.	*Property module*	Abaqus/Standard Abaqus/Explicit

Keyword	Purpose	Module	Products
**VOLUMETRIC TEST DATA*	Provide volumetric test data.	*Property module*	Abaqus/Standard Abaqus/Explicit
**WAVE*	Define gravity waves for use in immersed structure calculations.	Unsupported	Abaqus/Aqua
**WEAR SURFACE PROPERTIES*	Specify wear properties to activate wear distance accumulation at contact surface nodes.	Unsupported	Abaqus/Standard Abaqus/Explicit
**WIND*	Define wind velocity profile for wind loading.	Unsupported	Abaqus/Aqua

Keyword support from the input file reader

To see keywords beginning with a particular letter of the alphabet, click that letter in the table below.

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*
N	*O*	*P*	Q	*R*	*S*	*T*	*U*	*V*	*W*	X	Y	Z

- **ACOUSTIC CONTRIBUTION*

  Not supported
- **ACOUSTIC FLOW VELOCITY*

  Not supported
- **ACOUSTIC MEDIUM*

  BULK MODULUS, DEPENDENCIES, VOLUMETRIC DRAG
- **ACOUSTIC WAVE FORMULATION*

  TYPE
- **ACTIVATE ELEMENTS*

  Not supported
- **ADAPTIVE MESH*

  CONTROLS, ELSET, FREQUENCY, INITIAL MESH SWEEPS, MESH SWEEPS, OP
- **ADAPTIVE MESH CONSTRAINT*

  AMPLITUDE, CONSTRAINT TYPE, OP, TYPE, USER
- **ADAPTIVE MESH CONTROLS*

  ADVECTION, CURVATURE REFINEMENT, GEOMETRIC ENHANCEMENT, INITIAL FEATURE ANGLE, MESH CONSTRAINT ANGLE, MESHING PREDICTOR, MOMENTUM ADVECTION, NAME, RESET, SMOOTHING OBJECTIVE, TRANSITION FEATURE ANGLE
- **ADAPTIVE MESH REFINEMENT*

  Not supported
- **ADJUST*

  NODE SET, SURFACE
- **ALLOWABLE STRESS*

  Not supported
- **AMPLITUDE*

  BEGIN, DEFINITION (All values except BUBBLE), FIXED INTERVAL, NAME, SMOOTH, TIME, VARIABLES
- **ANISOTROPIC HYPERELASTIC*

  DEPENDENCIES, FORMULATION, LOCAL DIRECTIONS, MODULI, PROPERTIES, TYPE
- **ANNEAL*

  TEMPERATURE
- **ANNEAL TEMPERATURE*

Not supported

- **AQUA*

  Not supported

- **ASSEMBLY*

  NAME

- **ASYMMETRIC-AXISYMMETRIC*

  Not supported

- **AXIAL*

  Not supported

- **BASE MOTION*

  Not supported

- **BASELINE CORRECTION*

  (No parameters)

- **BEAM ADDED INERTIA*

  Not supported

- **BEAM FLUID INERTIA*

  FULL, HALF

- **BEAM GENERAL SECTION*

  DENSITY, DEPENDENCIES, ELSET, LUMPED, MATERIAL, POISSON, SECTION (All values except MESHED, NONLINEAR GENERAL), TAPER, ZERO

- **BEAM SECTION*

  ELSET, MATERIAL, POISSON, SECTION (All values except ELBOW), TEMPERATURE

- **BEAM SECTION GENERATE*

  Not supported

- **BEAM SECTION OFFSET*

  (No parameters)

- **BIAXIAL TEST DATA*

  DEPENDENCIES, SMOOTH

- **BLOCKAGE*

  Not supported

- **BOUNDARY*

  AMPLITUDE, FIXED, IMAGINARY, INC, LOAD CASE, OP, REAL, SCALE, STEP, SUBMODEL, TIMESCALE, TYPE

- **BRITTLE CRACKING*

  DEPENDENCIES, TYPE

- **BRITTLE FAILURE*

  CRACKS, DEPENDENCIES

- **BRITTLE SHEAR*

DEPENDENCIES, TYPE

- **BUCKLE*

  EIGENSOLVER
- **BUCKLING ENVELOPE*

  Not supported
- **BUCKLING LENGTH*

  Not supported
- **BUCKLING REDUCTION FACTORS*

  Not supported
- **BULK VISCOSITY*

  (No parameters)
- **C ADDED MASS*

  Not supported
- **CAP CREEP*

  DEPENDENCIES, LAW, MECHANISM, TIME
- **CAP HARDENING*

  DEPENDENCIES
- **CAP PLASTICITY*

  DEPENDENCIES
- **CAPACITY*

  DEPENDENCIES, TYPE
- **CAST IRON COMPRESSION HARDENING*

  DEPENDENCIES
- **CAST IRON PLASTICITY*

  DEPENDENCIES
- **CAST IRON TENSION HARDENING*

  DEPENDENCIES
- **CAVITY DEFINITION*

  AMBIENT TEMP, NAME, SET PROPERTY
- **CECHARGE*

  AMPLITUDE, IMAGINARY, OP, REAL
- **CECURRENT*

  AMPLITUDE, OP
- **CENTROID*

  (No parameters)
- **CFILM*

  AMPLITUDE, FILM AMPLITUDE, OP, REGION TYPE, USER

- **CFLOW*

  Not supported
- **CFLUX*

  AMPLITUDE, OP
- **CHANGE FRICTION*

  INTERACTION, RESET
- **CHARACTERISTIC LENGTH*

  Not supported
- **CLAY HARDENING*

  DEPENDENCIES, TYPE
- **CLAY PLASTICITY*

  DEPENDENCIES, HARDENING, INTERCEPT
- **CLEARANCE*

  BOLT, CPSET, HANDEDNESS, MAIN, NORMAL ADJUSTMENT, SECONDARY, TABULAR, VALUE
- **CLOAD*

  AMPLITUDE, CYCLIC MODE, FOLLOWER, IMAGINARY, LOAD CASE, OP, REAL
- **CLUSTER MASS INERTIA TABLE*

  Not supported
- **COHESIVE BEHAVIOR*

  DEPENDENCIES, ELIGIBILITY, REPEATED CONTACTS, TYPE
- **COHESIVE SECTION*

  CONTROLS, ELSET, MATERIAL, ORIENTATION, RESPONSE, THICKNESS
- **COMBINATORIAL RULE*

  Not supported
- **COMBINED TEST DATA*

  SHRINF, VOLINF
- **COMPLEX FREQUENCY*

  FRICTION DAMPING, PROPERTY EVALUATION
- **COMPOSITE MODAL DAMPING*

  Not supported
- **CONCENTRATION TENSOR*

  Not supported
- **CONCRETE*

  DEPENDENCIES
- **CONCRETE COMPRESSION DAMAGE*

  DEPENDENCIES, TENSION RECOVERY
- **CONCRETE COMPRESSION HARDENING*

DEPENDENCIES

- **CONCRETE DAMAGED PLASTICITY*

  DEPENDENCIES

- **CONCRETE FAILURE*

  Not supported

- **CONCRETE TENSION DAMAGE*

  COMPRESSION RECOVERY, DEPENDENCIES

- **CONCRETE TENSION STIFFENING*

  DEPENDENCIES, TYPE

- **CONDUCTIVITY*

  DEPENDENCIES, TYPE

- **CONNECTOR BEHAVIOR*

  EXTRAPOLATION, INTEGRATION, NAME, REGULARIZE, RTOL

- **CONNECTOR CONSTITUTIVE REFERENCE*

  (No parameters)

- **CONNECTOR DAMAGE EVOLUTION*

  AFFECTED COMPONENTS, DEGRADATION, DEPENDENCIES, EXTRAPOLATION, REGULARIZE, RTOL, SOFTENING, TYPE

- **CONNECTOR DAMAGE INITIATION*

  COMPONENT, CRITERION, DEPENDENCIES, EXTRAPOLATION, RATE FILTER FACTOR, RATE INTERPOLATION, REGULARIZE, RTOL

- **CONNECTOR DAMPING*

  COMPONENT, DEPENDENCIES, EXTRAPOLATION, INDEPENDENT COMPONENTS, NONLINEAR, REGULARIZE, RTOL, TYPE

- **CONNECTOR DERIVED COMPONENT*

  DEPENDENCIES, EXTRAPOLATION, INDEPENDENT COMPONENTS, NAME, OPERATOR, REGULARIZE, RTOL, SIGN

- **CONNECTOR ELASTICITY*

  COMPONENT, DEPENDENCIES, EXTRAPOLATION, INDEPENDENT COMPONENTS, NONLINEAR, REGULARIZE, RIGID, RTOL

- **CONNECTOR FAILURE*

  COMPONENT, RELEASE

- **CONNECTOR FRICTION*

  COMPONENT, CONTACT FORCE, DEPENDENCIES, EXTRAPOLATION, INDEPENDENT COMPONENTS, PREDEFINED, REGULARIZE, RTOL, STICK STIFFNESS

- **CONNECTOR HARDENING*

  DEFINITION, DEPENDENCIES, EXTRAPOLATION, RATE FILTER FACTOR, RATE INTERPOLATION, REGULARIZE, RTOL, TYPE

- **CONNECTOR LOAD*

  AMPLITUDE, IMAGINARY, LOAD CASE, OP, REAL

- **CONNECTOR LOCK*

  COMPONENT, EXTRAPOLATION, LOCK, REGULARIZE, RTOL
- **CONNECTOR MOTION*

  AMPLITUDE, FIXED, IMAGINARY, LOAD CASE, OP, REAL, TYPE, USER
- **CONNECTOR PLASTICITY*

  COMPONENT
- **CONNECTOR POTENTIAL*

  EXPONENT, OPERATOR
- **CONNECTOR SECTION*

  BEHAVIOR, ELSET
- **CONNECTOR STOP*

  COMPONENT
- **CONNECTOR UNIAXIAL BEHAVIOR*

  Not supported
- **CONSTITUENT*

  Not supported
- **CONSTRAINT CONTROLS*

  Not supported
- **CONTACT*

  OP
- **CONTACT CLEARANCE*

  Not supported
- **CONTACT CLEARANCE ASSIGNMENT*

  Not supported
- **CONTACT CONTROLS*

  ABSOLUTE PENETRATION TOLERANCE, CPSET, FASTLOCALTRK, GLOBTRKINC, MAIN, RELATIVE PENETRATION TOLERANCE, RESET, SCALE PENALTY, SECONDARY, STABILIZE, STIFFNESS SCALE FACTOR, TANGENT FRACTION, WARP CHECK PERIOD, WARP CUT OFF
  (This option is supported for Abaqus/Explicit only; only one definition per step is supported.)
- **CONTACT CONTROLS ASSIGNMENT*

  BEAM CROSS SECTION
- **CONTACT DAMPING*

  DEFINITION, TANGENT FRACTION
- **CONTACT EXCLUSIONS*

  (No parameters)
- **CONTACT FILE*

  Not supported
- **CONTACT FORMULATION*

FORMULATION, TYPE

- **CONTACT INCLUSIONS*

  ALL EXTERIOR

- **CONTACT INITIALIZATION ASSIGNMENT*

  (No parameters)

- **CONTACT INITIALIZATION DATA*

  ADJUST, INITIAL CLEARANCE, INTERFERENCE FIT, NAME, SEARCH ABOVE, SEARCH BELOW, SEARCH NSET, STEP FRACTION

- **CONTACT INTERFERENCE*

  AMPLITUDE, OP, SHRINK, TYPE (All values except ELEMENT)

- **CONTACT MASS SCALING*

  LOCATION

- **CONTACT OUTPUT*

  NSET, SURFACE, VARIABLE

- **CONTACT PAIR*

  ADJUST, CPSET, GEOMETRIC CORRECTION, HCRIT, INTERACTION, MECHANICAL CONSTRAINT, NO THICKNESS, OP, SMALL SLIDING, SMOOTH, SUPPLEMENTARY CONSTRAINTS, TIED, TRACKING, TYPE, WEIGHT

- **CONTACT PERMEABILITY*

  Not supported

- **CONTACT PRINT*

  Not supported

- **CONTACT PROPERTY ASSIGNMENT*

  (No parameters)

- **CONTACT RESPONSE*

  Not supported

- **CONTACT STABILIZATION*

  Not supported

- **CONTOUR INTEGRAL*

  CONTOURS, CRACK NAME, CRACK TIP NODES, DIRECTION, FREQUENCY, NORMAL, OUTPUT, RESIDUAL STRESS STEP, SYMM, TYPE, XFEM

- **CONTROLS*

  ANALYSIS, FIELD, PARAMETERS, RESET, TYPE

- **CONWEP CHARGE PROPERTY*

  (No parameters)

- **CORRELATION*

  Not supported

- **CO-SIMULATION*

  NAME, PROGRAM (All values except)

- **CO-SIMULATION REGION*

  EXPORT, IMPORT, TYPE
- **COUPLED CONSTITUTIVE RESPONSE*

  Not supported
- **COUPLED TEMPERATURE-DISPLACEMENT*

  ALLSDTOL, CETOL, CONTINUE, CREEP, DELTMX, FACTOR, STABILIZE, STEADY STATE
- **COUPLED THERMAL-ELECTRICAL*

  DELTMX, END, MXDEM, STEADY STATE
- **COUPLED THERMAL-ELECTROCHEMICAL*

  DELTMX, STEADY STATE
- **COUPLING*

  CONSTRAINT NAME, INFLUENCE RADIUS, ORIENTATION, REF NODE, SURFACE
- **CRADIATE*

  AMPLITUDE, OP, REGION TYPE
- **CREEP*

  DEPENDENCIES, LAW, TIME
- **CREEP STRAIN RATE CONTROL*

  Not supported
- **CRUSH STRESS*

  DEPENDENCIES
- **CRUSH STRESS VELOCITY FACTOR*

  (No parameters)
- **CRUSHABLE FOAM*

  DEPENDENCIES, HARDENING
- **CRUSHABLE FOAM HARDENING*

  DEPENDENCIES
- **CURE GLASS TRANSITION TEMPERATURE*

  Not supported
- **CURE HEAT GENERATION*

  Not supported
- **CURE KINETICS*

  Not supported
- **CURE MAX CONVERSION*

  Not supported
- **CURE SHRINKAGE*

  Not supported
- **CYCLED PLASTIC*

(No parameters)

- **CYCLIC*

  NC, TYPE

- **CYCLIC HARDENING*

  DEPENDENCIES, PARAMETERS

- **CYCLIC SYMMETRY MODEL*

  N

- **D ADDED MASS*

  Not supported

- **D EM POTENTIAL*

  Not supported

- **DAMAGE EVOLUTION*

  DEGRADATION, DEPENDENCIES, FAILURE INDEX, MIXED MODE BEHAVIOR, MODE MIX RATIO, POWER, RATE DEPENDENT, REF ENERGY, SOFTENING, TYPE

- **DAMAGE INITIATION*

  ACCUMULATION POWER, ALPHA, ANGLEMAX, ANGSMOOTH, CRITERION, DEFINITION, DEPENDENCIES, FAILURE MECHANISMS, FEQ, FNN, FNT, FREQUENCY, GROWTH TOLERANCE, INISMOOTH, KS, LODE DEPENDENT, NORMAL DIRECTION, NPOLY, NUMBER IMPERFECTIONS, OMEGA, PEINC, POSITION, PROPERTIES, R CRACK DIRECTION, RATE DEPENDENT, REF ENERGY, SMOOTHING, TOLERANCE, UNSTABLE GROWTH TOLERANCE, WEIGHTING METHOD

- **DAMAGE STABILIZATION*

  (No parameters)

- **DAMPING*

  ALPHA, BETA, COMPOSITE, STRUCTURAL

- **DAMPING CONTROLS*

  STRUCTURAL, VISCOUS

- **DASHPOT*

  ELSET, ORIENTATION, RTOL

- **DEBOND*

  DEBONDING FORCE, FREQUENCY, MAIN, SECONDARY

- **DECHARGE*

  AMPLITUDE, IMAGINARY, OP, REAL

- **DECURRENT*

  AMPLITUDE, OP

- **DEFORMATION PLASTICITY*

  (No parameters)

- **DENSITY*

  DEPENDENCIES

- **DEPVAR*

DELETE

- **DESIGN GRADIENT*

  Not supported

- **DESIGN PARAMETER*

  Not supported

- **DESIGN RESPONSE*

  Not supported

- **DETONATION POINT*

  (No parameters)

- **DFLOW*

  AMPLITUDE, OP

- **DFLUX*

  AMPLITUDE, OP

- **DIAGNOSTICS*

  Not supported

- **DIELECTRIC*

  DEPENDENCIES, TYPE

- **DIFFUSIVITY*

  DEPENDENCIES, LAW, TYPE

- **DIRECT CYCLIC*

  CETOL, CONTINUE, DELTMX, FATIGUE, TIME POINTS

- **DISCRETE ELASTICITY*

  Not supported

- **DISCRETE SECTION*

  Not supported

- **DISPLAY BODY*

  INSTANCE

- **DISTRIBUTING*

  COUPLING, ROTATIONAL COUPLING, WEIGHTING METHOD

- **DISTRIBUTING COUPLING*

  Not supported

- **DISTRIBUTION*

  LOCATION, NAME, TABLE

- **DISTRIBUTION TABLE*

  NAME

- **DLOAD*

  AMPLITUDE, CONSTANT RESULTANT, CYCLIC MODE, FOLLOWER, IMAGINARY, OP, ORIENTATION, REAL

- **DOMAIN DECOMPOSITION*

  Not supported
- **DRAG CHAIN*

  Not supported
- **DRUCKER PRAGER*

  DEPENDENCIES, ECCENTRICITY, SHEAR CRITERION, TEST DATA
- **DRUCKER PRAGER CREEP*

  DEPENDENCIES, LAW, TIME
- **DRUCKER PRAGER HARDENING*

  DEPENDENCIES, RATE, TYPE
- **DSA CONTROLS*

  Not supported
- **DSECHARGE*

  AMPLITUDE, IMAGINARY, OP, REAL
- **DSECURRENT*

  AMPLITUDE, OP
- **DSFLOW*

  AMPLITUDE, OP
- **DSFLUX*

  AMPLITUDE, OP
- **DSLOAD*

  AMPLITUDE, CONSTANT RESULTANT, CYCLIC MODE, FOLLOWER, IMAGINARY, INC, OP, ORIENTATION, REAL, REF NODE, STEP, SUBMODEL
- **DYNAMIC*

  ADIABATIC, ALPHA, APPLICATION, DIRECT USER CONTROL, ELEMENT BY ELEMENT, EXPLICIT, FIXED TIME INCREMENTATION, HAFTOL, HALFINC SCALE FACTOR, IMPROVED DT METHOD, INITIAL, NOHAF, SCALE FACTOR, SUBSPACE
- **DYNAMIC TEMPERATURE-DISPLACEMENT*

  DIRECT USER CONTROL, ELEMENT BY ELEMENT, EXPLICIT, FIXED TIME INCREMENTATION, IMPROVED DT METHOD, SCALE FACTOR
- **EIGENSTRAIN*

  Not supported
- **EL FILE*

  Not supported
- **EL PRINT*

  Not supported
- **ELASTIC*

  DEPENDENCIES, MODULI, TYPE (All values except SHORT FIBER)

- **ELCOPY*

  Not supported
- **ELECTRIC MACHINE LOAD*

  Not supported
- **ELECTRIC MACHINE PROPERTY*

  Not supported
- **ELECTRICAL CONDUCTIVITY*

  DEPENDENCIES, TYPE
- **ELECTRICAL RESISTIVITY*

  Not supported
- **ELECTROMAGNETIC*

  Not supported
- **ELEMENT*

  ELSET, FILE, TYPE
- **ELEMENT MATRIX OUTPUT*

  Not supported
- **ELEMENT OPERATOR OUTPUT*

  Not supported
- **ELEMENT OUTPUT*

  DIRECTIONS, ELSET, EXTERIOR, POSITION, REBAR, VARIABLE
- **ELEMENT PROGRESSIVE ACTIVATION*

  Not supported
- **ELEMENT RECOVERY MATRIX*

  Not supported
- **ELEMENT RESPONSE*

  Not supported
- **ELEMENT SOLUTION-DEPENDENT VARIABLES*

  Not supported
- **ELEMENT USER OUTPUT VARIABLES*

  Not supported
- **ELGEN*

  ALL NODES, ELSET
- **ELSET*

  ELSET, GENERATE, INSTANCE, INTERNAL
- **EMBEDDED ELEMENT*

  ABSOLUTE EXTERIOR TOLERANCE, EXTERIOR TOLERANCE, HOST ELSET, ROUNDOFF TOLERANCE
- **EMISSIVITY*

DEPENDENCIES

- **END ASSEMBLY*

  (No parameters)
- **END INSTANCE*

  (No parameters)
- **END LOAD CASE*

  (No parameters)
- **END PART*

  (No parameters)
- **END STEP*

  (No parameters)
- **ENERGY FILE*

  Not supported
- **ENERGY OUTPUT*

  ELSET, VARIABLE
- **ENERGY PRINT*

  Not supported
- **ENRICHMENT*

  ELSET, ENRICHMENT RADIUS, INTERACTION, NAME, TYPE
- **ENRICHMENT ACTIVATION*

  ACTIVATE, NAME, TYPE
- **EOS*

  DETONATION ENERGY, TYPE (All values except TABULAR)
- **EOS COMPACTION*

  (No parameters)
- **EPJOINT*

  Not supported
- **EQUATION*

  (No parameters. This option is supported if the data lines contain only node sets; the use of node labels for this option is not yet supported in Abaqus/CAE. In addition, the second node set (the fourth data item) can contain only a single node.)
- **EQUIVALENT RADIATED SURFACE PROPERTIES*

  Not supported
- **EULERIAN BOUNDARY*

  INFLOW, OP, OUTFLOW
- **EULERIAN MESH MOTION*

  ASPECT RATIO MAX, BUFFER, CONTRACT, ELSET, OP, ORIENTATION, SURFACE, VMAX FACTOR, VOLFRAC MIN

- **EULERIAN SECTION*

  CONTROLS, ELSET
- **EVENT SERIES*

  Not supported
- **EVENT SERIES TYPE*

  Not supported
- **EXPANSION*

  DEPENDENCIES, PORE FLUID, TYPE (All values except SHORT FIBER), USER, ZERO
- **EXTERNAL FIELD*

  Not supported
- **EXTREME ELEMENT VALUE*

  Not supported
- **EXTREME NODE VALUE*

  Not supported
- **EXTREME VALUE*

  Not supported
- **FABRIC*

  Not supported
- **FAIL STRAIN*

  DEPENDENCIES
- **FAIL STRESS*

  DEPENDENCIES
- **FAILURE RATIOS*

  DEPENDENCIES
- **FASTENER*

  ADJUST ORIENTATION, ATTACHMENT METHOD, COUPLING, ELSET, INTERACTION NAME, NUMBER OF LAYERS, ORIENTATION, PROPERTY, RADIUS OF INFLUENCE, REFERENCE NODE SET, SEARCH RADIUS, UNSORTED, WEIGHTING METHOD
- **FASTENER PROPERTY*

  NAME
- **FATIGUE*

  Not supported
- **FIBER DISPERSION*

  Not supported
- **FIELD*

  AMPLITUDE, BINC, BSTEP, EINC, ESTEP, FILE, INPUT, INTERPOLATE, OP, OUTPUT VARIABLE, USER, VARIABLE
- **FIELD IMPORT*

Not supported

- **FIELD MAPPER CONTROLS*

  Not supported
- **FIELD OPERATIONS*

  Not supported
- **FILE FORMAT*

  Not supported
- **FILE OUTPUT*

  Not supported
- **FILM*

  AMPLITUDE, FILM AMPLITUDE, OP
- **FILM PROPERTY*

  DEPENDENCIES, NAME
- **FILTER*

  HALT, INVARIANT, LIMIT, NAME, OPERATOR, TYPE
- **FIXED MASS SCALING*

  DT, ELSET, FACTOR, TYPE
- **FLEXIBLE BODY*

  TYPE
- **FLOW*

  Not supported
- **FLUID BEHAVIOR*

  NAME
- **FLUID BULK MODULUS*

  DEPENDENCIES
- **FLUID CAVITY*

  ADIABATIC, AMBIENT PRESSURE, BEHAVIOR, CHECK NORMALS, NAME, REF NODE, SURFACE
- **FLUID DENSITY*

  (No parameters)
- **FLUID EXCHANGE*

  EFFECTIVE AREA, NAME, PROPERTY
- **FLUID EXCHANGE ACTIVATION*

  AMPLITUDE, BLOCKAGE, DELTA LEAKAGE AREA, OP, OUTFLOW ONLY
- **FLUID EXCHANGE PROPERTY*

  DEPENDENCIES, NAME, TYPE (All values except ENERGY FLUX, ENERGY RATE LEAKAGE, FABRIC LEAKAGE, ORIFICE, USER)
- **FLUID EXPANSION*

  DEPENDENCIES, ZERO

- **FLUID FLUX*

  Not supported
- **FLUID ID FIELD*

  Not supported
- **FLUID INFLATOR*

  NAME, PROPERTY
- **FLUID INFLATOR ACTIVATION*

  INFLATION TIME AMPLITUDE, MASS FLOW AMPLITUDE, OP
- **FLUID INFLATOR MIXTURE*

  NUMBER SPECIES, TYPE
- **FLUID INFLATOR PROPERTY*

  DISCHARGE COEFFICIENT, EFFECTIVE AREA, NAME, TANK VOLUME, TYPE
- **FLUID LEAKOFF*

  DEPENDENCIES, USER
- **FLUID PIPE CONNECTOR LOSS*

  Not supported
- **FLUID PIPE CONNECTOR SECTION*

  Not supported
- **FLUID PIPE CONNECTOR THERMAL LOSS*

  Not supported
- **FLUID PIPE CONNECTOR WALL*

  Not supported
- **FLUID PIPE FLOW LOSS*

  Not supported
- **FLUID PIPE SECTION*

  Not supported
- **FLUID PIPE THERMAL*

  Not supported
- **FLUID PIPE WALL*

  Not supported
- **FOUNDATION*

  (No parameters)
- **FRACTURE CRITERION*

  DEPENDENCIES, MIXED MODE BEHAVIOR, NORMAL DIRECTION, TOLERANCE, TYPE (All values except COD, CRACK LENGTH, CRITICAL STRESS, FATIGUE), UNSTABLE GROWTH TOLERANCE, VISCOSITY
- **FRAME SECTION*

  Not supported

- **FREQUENCY*

  ACOUSTIC COUPLING, DAMPING PROJECTION, EIGENSOLVER, NORMALIZATION, NSET, PROPERTY EVALUATION, RESIDUAL MODES, SIM
- **FRICTION*

  ANISOTROPIC BEHAVIOR, DEPENDENCIES, DEPVAR, ELASTIC SLIP, EXPONENTIAL DECAY, LAGRANGE, PROPERTIES, ROUGH, SHEAR TRACTION SLOPE, SLIP TOLERANCE, TAUMAX, TEST DATA, USER
- **GAP*

  Not supported
- **GAP CONDUCTANCE*

  DEPENDENCIES, PRESSURE, USER
- **GAP CONVECTION*

  DEPENDENCIES, TYPE
- **GAP DIFFUSIVITY*

  Not supported
- **GAP ELECTRICAL CONDUCTANCE*

  DEPENDENCIES, PRESSURE, TYPE, USER
- **GAP FLOW*

  DEPENDENCIES, KMAX, TYPE
- **GAP HEAT GENERATION*

  (No parameters)
- **GAP RADIATION*

  (No parameters)
- **GAS SPECIFIC HEAT*

  DEPENDENCIES
- **GASKET BEHAVIOR*

  NAME
- **GASKET CONTACT AREA*

  DEPENDENCIES
- **GASKET ELASTICITY*

  COMPONENT, DEPENDENCIES, VARIABLE
- **GASKET SECTION*

  BEHAVIOR, ELSET, MATERIAL, ORIENTATION, STABILIZATION STIFFNESS
- **GASKET THICKNESS BEHAVIOR*

  DEPENDENCIES, DIRECTION, SLOPE DROP, TENSILE STIFFNESS FACTOR, TYPE, VARIABLE, YIELD ONSET
- **GEL*

  (No parameters)

- **GEOSTATIC*

  UTOL
- **GLOBAL DAMPING*

  ALPHA, BETA, FIELD, STRUCTURAL
- **GLOBAL RESPONSE*

  Not supported
- **HEADING*

  (No parameters)
- **HEAT GENERATION*

  0 (No parameters)
- **HEAT TRANSFER*

  DELTMX, END, MXDEM, STEADY STATE
- **HEATCAP*

  DEPENDENCIES, ELSET
- **HOURGLASS STIFFNESS*

  (No parameters)
- **HYPERELASTIC*

  ARRUDA-BOYCE, BETA, MARLOW, MODULI, MOONEY-RIVLIN, N, NEO HOOKE, OGDEN, POISSON, POLYNOMIAL, PROPERTIES, REDUCED POLYNOMIAL, TEST DATA INPUT, TYPE, USER, VALANIS-LANDEL, VAN DER WAALS, YEOH
- **HYPERFOAM*

  MODULI, N, POISSON, TEST DATA INPUT
- **HYPOELASTIC*

  USER
- **HYSTERESIS*

  (No parameters)
- **IMPEDANCE*

  Not supported
- **IMPEDANCE PROPERTY*

  DATA, NAME
- **IMPERFECTION*

  FILE, INC, INPUT, NSET, STEP, SYSTEM
- **IMPORT*

  Not supported
- **IMPORT CONTROLS*

  Not supported
- **IMPORT ELSET*

Not supported

- **IMPORT NSET*

  Not supported

- **IMPORT SURFACE*

  Not supported

- **INCIDENT WAVE*

  Not supported

- **INCIDENT WAVE FLUID PROPERTY*

  Not supported

- **INCIDENT WAVE INTERACTION*

  ACCELERATION AMPLITUDE, CONWEP, IMAGINARY, PRESSURE AMPLITUDE, PROPERTY, REAL, UNDEX

- **INCIDENT WAVE INTERACTION PROPERTY*

  NAME, TYPE

- **INCIDENT WAVE PROPERTY*

  Not supported

- **INCIDENT WAVE REFLECTION*

  Not supported

- **INCLUDE*

  (No parameters)

- **INCREMENTATION OUTPUT*

  Not supported

- **INELASTIC HEAT FRACTION*

  (No parameters)

- **INERTIA RELIEF*

  FIXED, ORIENTATION, REMOVE

- **INITIAL CONDITIONS*

  FILE, FULL TENSOR, GEOSTATIC, INC, MIDSIDE, NUMBER BACKSTRESSES, OUTPUT VARIABLE, REBAR, SECTION POINTS, STEP, TYPE (All values except ACOUSTIC STATIC PRESSURE, CONCENTRATION, MASS FLOW RATE, POROSITY, PRESSURE STRESS, RELATIVE DENSITY, SOLUTION, SPECIFIC ENERGY), USER, VARIABLE

- **INSTANCE*

  NAME, PART

- **INTEGRATED OUTPUT*

  SECTION, VARIABLE

- **INTEGRATED OUTPUT SECTION*

  NAME, ORIENTATION, POSITION, PROJECT ORIENTATION, REF NODE MOTION, REF NODE, SURFACE

- **INTERFACE*

ELSET, NAME

- **INTERFACE REACTION*

  Not supported
- **INTERNAL FLUX GENERATION*

  Not supported
- **ISOTROPIZATION PARAMETERS*

  Not supported
- **ITS*

  Not supported
- **JOINT*

  Not supported
- **JOINT ELASTICITY*

  Not supported
- **JOINT PLASTICITY*

  Not supported
- **JOINTED MATERIAL*

  Not supported
- **JOULE HEAT FRACTION*

  (No parameters)
- **KAPPA*

  DEPENDENCIES, TYPE
- **KINEMATIC*

  ALPHA
- **KINEMATIC COUPLING*

  Not supported
- **LATENT HEAT*

  (No parameters)
- **LOAD CASE*

  NAME
- **LOADING DATA*

  Not supported
- **LOW DENSITY FOAM*

  FAIL, RATE EXTRAPOLATION, STRAIN RATE, TENSION CUTOFF
- **M1*

  Not supported
- **M2*

Not supported

- **MAGNETIC PERMEABILITY*

  Not supported
- **MAGNETOSTATIC*

  Not supported
- **MANIFEST*

  Not supported
- **MAP SOLUTION*

  Not supported
- **MASS*

  ALPHA, COMPOSITE, ELSET
- **MASS ADJUST*

  Not supported
- **MASS DIFFUSION*

  DCMAX, END, STEADY STATE
- **MASS FLOW RATE*

  Not supported
- **MATERIAL*

  NAME, RTOL
- **MATRIX*

  Not supported
- **MATRIX ASSEMBLE*

  Not supported
- **MATRIX CHECK*

  Not supported
- **MATRIX EXPORT*

  Not supported
- **MATRIX GENERATE*

  Not supported
- **MATRIX INPUT*

  Not supported
- **MATRIX OUTPUT*

  Not supported
- **MEAN FIELD DAMAGE*

  Not supported
- **MEAN FIELD HOMOGENIZATION*

  Not supported

- **MEDIA TRANSPORT*

  Not supported
- **MEMBRANE SECTION*

  CONTROLS, ELSET, MATERIAL, MEMBRANE THICKNESS, ORIENTATION, POISSON (All values except MATERIAL)
- **MODAL DAMPING*

  DEFINITION, MODAL, RAYLEIGH, STRUCTURAL
- **MODAL DYNAMIC*

  CONTINUE
- **MODAL FILE*

  Not supported
- **MODAL OUTPUT*

  VARIABLE
- **MODAL PRINT*

  Not supported
- **MODEL CHANGE*

  ACTIVATE, ADD, REMOVE, TYPE
- **MOHR COULOMB*

  DEPENDENCIES, DEVIATORIC ECCENTRICITY, ECCENTRICITY
- **MOHR COULOMB HARDENING*

  DEPENDENCIES
- **MOISTURE SWELLING*

  (No parameters)
- **MOLECULAR WEIGHT*

  (No parameters)
- **MONITOR*

  DOF, FREQUENCY, NODE
- **MOTION*

  Not supported
- **MPC*

  MODE, USER
- **MULLINS EFFECT*

  BETA, DEPENDENCIES, M, PROPERTIES, R, TEST DATA INPUT
- **MULTIPHYSICS LOAD*

  Not supported
- **NCOPY*

  CHANGE NUMBER, MULTIPLE, NEW SET, OLD SET, POLE, REFLECT, SHIFT

- **NETWORK STIFFNESS RATIO*

  Not supported
- **NFILL*

  BIAS, NSET, SINGULAR, TWO STEP
- **NGEN*

  LINE, NSET
- **NMAP*

  Not supported
- **NO COMPRESSION*

  (No parameters)
- **NO TENSION*

  (No parameters)
- **NODAL ENERGY RATE*

  Not supported
- **NODAL THICKNESS*

  (No parameters)
- **NODE*

  NSET, SYSTEM
- **NODE FILE*

  Not supported
- **NODE OUTPUT*

  EXTERIOR, GLOBAL, NSET, VARIABLE
- **NODE PRINT*

  Not supported
- **NODE RESPONSE*

  Not supported
- **NONLINEAR BH*

  Not supported
- **NONSTRUCTURAL MASS*

  DISTRIBUTION, ELSET, UNITS
- **NORMAL*

  Not supported
- **NSET*

  GENERATE, INSTANCE, INTERNAL, NSET
- **ONE-STEP INVERSE*

  Not supported
- **OPERATOR OUTPUT*

Not supported

- **ORIENTATION*

  DEFINITION, NAME, SYSTEM

- **ORNL*

  A, H, MATERIAL, RESET

- **OUTPUT*

  FIELD, FILTER, FREQUENCY, HISTORY, MODE LIST, NAME, NUMBER INTERVAL, OP, SENSOR, TIME INTERVAL, TIME MARKS, TIME POINTS, VARIABLE

- **PAPERBOARD HARDENING*

  Not supported

- **PAPERBOARD PLASTICITY*

  Not supported

- **PAPERBOARD THICKNESS COMPRESSION ELASTIC*

  Not supported

- **PAPERBOARD THICKNESS COMPRESSION PLASTICITY*

  Not supported

- **PAPERBOARD TRANSVERSE SHEAR PLASTICITY*

  Not supported

- **PARAMETER*

  Not supported

- **PARAMETER DEPENDENCE*

  Not supported

- **PARAMETER SHAPE VARIATION*

  Not supported

- **PARAMETER TABLE*

  Not supported

- **PARAMETER TABLE TYPE*

  Not supported

- **PART*

  NAME

- **PARTICLE GENERATOR*

  Not supported

- **PARTICLE GENERATOR FLOW*

  Not supported

- **PARTICLE GENERATOR INLET*

  Not supported

- **PARTICLE GENERATOR MIXTURE*

Not supported

- **PARTICLE OUTLET*

  Not supported
- **PARTICLE OUTLET FLOW*

  Not supported
- **PERFECTLY MATCHED LAYER*

  Not supported
- **PERIODIC*

  NR, TYPE
- **PERIODIC MEDIA*

  Not supported
- **PERMANENT MAGNETIZATION*

  Not supported
- **PERMEABILITY*

  SPECIFIC, TYPE
- **PHYSICAL CONSTANTS*

  ABSOLUTE ZERO, STEFAN BOLTZMANN, UNIVERSAL GAS CONSTANT
- **PIEZOELECTRIC*

  TYPE
- **PIEZOELECTRIC DAMPING*

  Not supported
- **PIEZORESISTIVITY*

  Not supported
- **PIPE-SOIL INTERACTION*

  Not supported
- **PIPE-SOIL STIFFNESS*

  Not supported
- **PLANAR TEST DATA*

  DEPENDENCIES, SMOOTH
- **PLASTIC*

  DATA TYPE, DEPENDENCIES, EXTRAPOLATION, HARDENING, NUMBER BACKSTRESSES, PROPERTIES, RATE, SCALESTRESS, STATIC RECOVERY
- **PLASTIC AXIAL*

  Not supported
- **PLASTIC M1*

  Not supported
- **PLASTIC M2*

  Not supported

- **PLASTIC TORQUE*

  Not supported
- **PLASTICITY CORRECTION*

  DEFINITION, DEPENDENCIES
- **PLY FABRIC FAILURE*

  Not supported
- **PLY FABRIC HARDENING*

  Not supported
- **PML COEFFICIENT*

  Not supported
- **PORE FLUID PRESSURE*

  Not supported
- **POROUS BULK MODULI*

  (No parameters)
- **POROUS ELASTIC*

  DEPENDENCIES, SHEAR
- **POROUS ELECTRODE THEORY*

  Not supported
- **POROUS FAILURE CRITERIA*

  (No parameters)
- **POROUS METAL PLASTICITY*

  DEPENDENCIES, RELATIVE DENSITY
- **POST OUTPUT*

  Not supported
- **POTENTIAL*

  DEPENDENCIES
- **PREPRINT*

  CONTACT, ECHO, HISTORY, MODEL
- **PRESSURE PENETRATION*

  AMPLITUDE, IMAGINARY, MAIN, OP, PENETRATION TIME, REAL, SECONDARY
- **PRESSURE STRESS*

  Not supported
- **PRESTRESS HOLD*

  Not supported
- **PRE-TENSION SECTION*

  ELEMENT, NODE, SURFACE
- **PRINT*

ALLKE, CONTACT, CRITICAL ELEMENT, FREQUENCY, MODEL CHANGE, PLASTICITY, RESIDUAL, SOLVE

- **PROBABILITY DENSITY FUNCTION*

  Not supported
- **PROPERTY TABLE*

  Not supported
- **PROPERTY TABLE TYPE*

  Not supported
- **PSD-DEFINITION*

  Not supported
- **RADIATE*

  AMPLITUDE, OP
- **RADIATION FILE*

  Not supported
- **RADIATION OUTPUT*

  ELSET, VARIABLE
- **RADIATION PRINT*

  Not supported
- **RADIATION SYMMETRY*

  NAME
- **RADIATION VIEW FACTOR*

  BLOCKING, CAVITY, INTEGRATION, OFF, RANGE, REFLECTION, SYMMETRY, VTOL
- **RANDOM RESPONSE*

  (No parameters)
- **RATE DEPENDENT*

  DEPENDENCIES, TYPE
- **RATIOS*

  DEPENDENCIES
- **REACTION RATE*

  (No parameters)
- **REBAR*

  Not supported
- **REBAR LAYER*

  GEOMETRY, ORIENTATION
- **REBAR LINE*

  Not supported
- **REDUCED BASIS GENERATE*

Not supported

- **REFLECTION*

  TYPE

- **RELEASE*

  Not supported

- **RESPONSE SPECTRUM*

  COMP, SUM

- **RESTART*

  FREQUENCY, NUMBER INTERVAL, OVERLAY, TIME MARKS, WRITE

- **RETAINED NODAL DOFS*

  SORTED (All values except YES)

- **RIGID BODY*

  ANALYTICAL SURFACE, ELSET, ISOTHERMAL, PIN NSET, POSITION, REF NODE, TIE NSET

- **RIGID SURFACE*

  Not supported

- **ROTARY INERTIA*

  ALPHA, COMPOSITE, ELSET, ORIENTATION

- **SCALE MASS*

  Not supported

- **SCALE STIFFNESS*

  Not supported

- **SCALE STRESS DESIGN*

  Not supported

- **SCALE THERMAL CONDUCTIVITY*

  Not supported

- **SECTION CONTROLS*

  DISTORTION CONTROL, ELEMENT DELETION, HOURGLASS, INITIAL GAP OPENING, KINEMATIC SPLIT, LENGTH RATIO, LINEAR KINEMATIC CONVERSION, LKC RATIO, MAX DEGRADATION, NAME, SECOND ORDER ACCURACY, VISCOSITY, WEIGHT FACTOR

- **SECTION CURVATURE*

  Not supported

- **SECTION FILE*

  Not supported

- **SECTION INERTIA*

  Not supported

- **SECTION ORIGIN*

  Not supported

- **SECTION POINTS*

(No parameters)

- **SECTION PRINT*

  Not supported

- **SECTION STIFFNESS*

  Not supported

- **SECTION TANGENT*

  Not supported

- **SELECT CYCLIC SYMMETRY MODES*

  NMAX, NMIN

- **SELECT EIGENMODES*

  DEFINITION, GENERATE

- **SFILM*

  AMPLITUDE, FILM AMPLITUDE, OP

- **SFLOW*

  Not supported

- **SHEAR CENTER*

  (No parameters)

- **SHEAR FAILURE*

  Not supported

- **SHEAR RETENTION*

  DEPENDENCIES

- **SHEAR TEST DATA*

  SHRINF

- **SHELL GENERAL SECTION*

  BENDING ONLY, COMPOSITE, CONTROLS, DENSITY, DEPENDENCIES, ELSET, LAYUP, MATERIAL, MEMBRANE ONLY, NODAL THICKNESS, OFFSET, ORIENTATION, POISSON, SHELL THICKNESS, SMEAR ALL LAYERS, STACK DIRECTION, SYMMETRIC, THICKNESS MODULUS, ZERO

- **SHELL SECTION*

  COMPOSITE, CONTROLS, DENSITY, ELSET, LAYUP, MATERIAL, NODAL THICKNESS, OFFSET, ORIENTATION, POISSON (All values except ELASTIC, MATERIAL), SECTION INTEGRATION, SHELL THICKNESS, STACK DIRECTION, SYMMETRIC, TEMPERATURE, THICKNESS MODULUS

- **SHELL TO SOLID COUPLING*

  CONSTRAINT NAME, INFLUENCE DISTANCE, POSITION TOLERANCE

- **SIMPEDANCE*

  NONREFLECTING, OP, PROPERTY

- **SIMPLE SHEAR TEST DATA*

  (No parameters)

- **SLIDE LINE*

Not supported

- **SLOAD*

  AMPLITUDE, OP
- **SOFT ROCK HARDENING*

  DEPENDENCIES, SR, TYPE
- **SOFT ROCK PLASTICITY*

  DEPENDENCIES, ECCENTRICITY, FLOW EVALUATION
- **SOFTENING REGULARIZATION*

  DEPENDENCIES
- **SOILS*

  ALLSDTOL, CETOL, CONSOLIDATION, CONTINUE, CREEP, END, FACTOR, STABILIZE, UTOL
- **SOLID ELECTROLYTE THEORY*

  Not supported
- **SOLID SECTION*

  COMPOSITE, CONTROLS, ELSET, LAYUP, MATERIAL, ORIENTATION, REF NODE, STACK DIRECTION, SYMMETRIC
- **SOLUBILITY*

  DEPENDENCIES
- **SOLUTION TECHNIQUE*

  REFORM KERNEL, TYPE
- **SOLVER CONTROLS*

  Not supported
- **SORPTION*

  LAW, TYPE
- **SPECIFIC HEAT*

  DEPENDENCIES
- **SPECTRUM*

  ABSOLUTE, AMPLITUDE, CREATE, DAMPING GENERATE, EVENT TYPE, G, NAME, RELATIVE, TIME INCREMENT, TYPE
- **SPH SURFACE BEHAVIOR*

  Not supported
- **SPRING*

  ELSET, ORIENTATION, RTOL
- **SRADIATE*

  AMPLITUDE, OP
- **STATIC*

  ADIABATIC, ALLSDTOL, CONTINUE, DIRECT, FACTOR, FULLY PLASTIC, LONG TERM, RIKS, STABILIZE

- **STEADY STATE CRITERIA*

  Not supported
- **STEADY STATE DETECTION*

  Not supported
- **STEADY STATE DYNAMICS*

  DAMPING CHANGE, DIRECT, FREQUENCY SCALE, FRICTION DAMPING, INTERVAL, REAL ONLY, STIFFNESS CHANGE, SUBSPACE PROJECTION
- **STEADY STATE TRANSPORT*

  Not supported
- **STEP*

  AMPLITUDE, CONVERT SDI, EXTRAPOLATION, INC, NAME, NLGEOM, PERTURBATION, SOLVER, UNSYMM
- **STEP CONTROL*

  ACTION, DT REFINEMENT, NAME
- **STEP CYCLING*

  Not supported
- **STEP CYCLING CONTROL*

  Not supported
- **SUBCYCLING*

  Not supported
- **SUBMODEL*

  ABSOLUTE EXTERIOR TOLERANCE, EXTERIOR TOLERANCE, GLOBAL ELSET, SHELL THICKNESS, SHELL TO SOLID, TYPE
- **SUBMODEL CONDITIONS*

  Not supported
- **SUBMODEL CUT*

  Not supported
- **SUBSTRUCTURE CHANGE*

  Not supported
- **SUBSTRUCTURE DAMPING*

  Not supported
- **SUBSTRUCTURE DAMPING CONTROLS*

  Not supported
- **SUBSTRUCTURE GENERATE*

  ELSET, GRAVITY LOAD, MASS MATRIX, NSET, OVERWRITE, PROPERTY EVALUATION, RECOVERY MATRIX, STRUCTURAL DAMPING MATRIX, TYPE, VISCOUS DAMPING MATRIX
- **SUBSTRUCTURE LOAD CASE*

  NAME
- **SUBSTRUCTURE MATRIX OUTPUT*

Not supported

- **SUBSTRUCTURE MODAL DAMPING*

  Not supported
- **SUBSTRUCTURE OUTPUT*

  OP4, SIMNot supported
- **SUBSTRUCTURE PATH*

  ENTER ELEMENT, LEAVE
- **SUBSTRUCTURE PROPERTY*

  ELSET
- **SUPERELASTIC*

  NONASSOCIATED
- **SUPERELASTIC HARDENING*

  Not supported
- **SUPERELASTIC HARDENING MODIFICATIONS*

  PROPERTIES, USER
- **SURFACE*

  COMBINE, CROP, FILLET RADIUS, INTERNAL, NAME, PROPERTY, TYPE (All values except USER)
- **SURFACE BEHAVIOR*

  AUGMENTED LAGRANGE, DIRECT, NO SEPARATION, PENALTY, PRESSURE-OVERCLOSURE
- **SURFACE FLAW*

  Not supported
- **SURFACE INTERACTION*

  DEPVAR, NAME, PAD THICKNESS, PROPERTIES, TRACKING THICKNESS, UNSYMM, USER
- **SURFACE PROPERTY*

  NAME
- **SURFACE PROPERTY ASSIGNMENT*

  PROPERTY (All values except GEOMETRIC CORRECTION)
- **SURFACE SECTION*

  DENSITY, ELSET
- **SURFACE SMOOTHING*

  Not supported
- **SURFACE TENSION*

  Not supported
- **SURFACE TENSION CONTACT ANGLE*

  Not supported
- **SWELLING*

  DEPENDENCIES, LAW

- **SYMMETRIC MODEL GENERATION*

  Not supported
- **SYMMETRIC RESULTS TRANSFER*

  Not supported
- **SYSTEM*

  (No parameters)
- **TABLE COLLECTION*

  Not supported
- **TEMPERATURE*

  ABSOLUTE EXTERIOR TOLERANCE, AMPLITUDE, BINC, BSTEP, EINC, ESTEP, EXTERIOR TOLERANCE, FILE, INPUT, INTERPOLATE, MIDSIDE, OP, USER
- **TENSILE FAILURE*

  DEPENDENCIES, ELEMENT DELETION, PRESSURE, SHEAR
- **TENSION CUTOFF*

  DEPENDENCIES
- **TENSION STIFFENING*

  DEPENDENCIES, TYPE
- **THERMAL EXPANSION*

  Not supported
- **TIE*

  ADJUST, CONSTRAINT RATIO, CYCLIC SYMMETRY, NAME, NO ROTATION, NO THICKNESS, POSITION TOLERANCE, TYPE
- **TIME POINTS*

  GENERATE, NAME
- **TORQUE*

  Not supported
- **TORQUE PRINT*

  Not supported
- **TRACER PARTICLE*

  Not supported
- **TRANSFORM*

  NSET, TYPE
- **TRANSPORT VELOCITY*

  Not supported
- **TRANSVERSE SHEAR*

  Not supported
- **TRANSVERSE SHEAR STIFFNESS*

  (No parameters)

- **TRIAXIAL TEST DATA*

  A, B, PT
- **TRS*

  DEFINITION
- **UEL PROPERTY*

  Not supported
- **UNDEX CHARGE PROPERTY*

  (No parameters)
- **UNIAXIAL*

  Not supported
- **UNIAXIAL TEST DATA*

  DEPENDENCIES, DIRECTION, SMOOTH
- **UNIFORM*

  Not supported
- **UNIT SYSTEM*

  Not supported
- **UNLOADING DATA*

  Not supported
- **USER DEFINED FIELD*

  SPECIFIED, USER (No parameters)
- **USER ELEMENT*

  Not supported
- **USER MATERIAL*

  CONSTANTS, EFFMOD, HYBRID FORMULATION, TYPE, UNSYMM
- **USER OUTPUT VARIABLES*

  (No parameters)
- **VARIABLE MASS SCALING*

  CROSS SECTION NODES, DT, ELSET, EXTRUDED LENGTH, FEED RATE, FREQUENCY, NUMBER INTERVAL, TYPE
- **VIEW FACTOR OUTPUT*

  Not supported
- **VISCO*

  ALLSDTOL, CETOL, CONTINUE, CREEP, FACTOR, STABILIZE
- **VISCOELASTIC*

  ERRTOL, FREQUENCY, NMAX, PRELOAD, TIME, TYPE
- **VISCOSITY*

DEFINITION (All values except CARREAU-YASUDA, CROSS, ELLIS-METER, HERSCHEL-BULKLEY, POWELL-EYRING, POWER LAW, TABULAR), DEPENDENCIES

- **VISCOUS*

  DEPENDENCIES, LAW, TIME

- **VOID NUCLEATION*

  DEPENDENCIES

- **VOLUMETRIC TEST DATA*

  DEPENDENCIES, SMOOTH, VOLINF

- **WAVE*

  Not supported

- **WEAR SURFACE PROPERTIES*

  DEPENDENCIES, FRIC COEF DEPENDENT, REFERENCE STRESS, UNITLESS WEAR COEF

- **WIND*

  Not supported

Special element types

Abaqus uses special element types to designate model features such as analytical rigid surfaces that are not part of the mesh. These element types appear in the output database. You may see them in the Visualization module while querying the model or creating display groups based on element type. *Table 1* lists the element types generated by Abaqus.

Table 1: Special element types in the output database.

ARSC	Analytic rigid surface (cylindrical)
ARSR	Analytic rigid surface (rotational)
ARSSE	Analytic rigid surface (extruded)
ARSSS	Analytic rigid surface (segmented)
LSURF1	2-node line on two-dimensional surface
LSURF2	2-node line on axisymmetric surface
LSURF3	3-node line on three-dimensional surface
LSURF4	3-node line on two-dimensional surface
LSURF5	3-node line on axisymmetric surface
LSURF6	Quadratic triangular surface
PSURF1	Node (point) on two-dimensional surface
PSURF2	Node on axisymmetric surface
PSURF3	Node on three-dimensional surface
QSURF1	4-node facet on three-dimensional surface
QSURF2	8-node facet on three-dimensional surface
QSURF3	9-node facet on three-dimensional surface
RNODE2D	Reference node (two-dimensional)
RNODE3D	Reference node (three-dimensional)
TSURF1	3-node facet on three-dimensional surface
TSURF1	6-node facet on three-dimensional surface

Special graphical symbols

Abaqus/CAE uses special graphical symbols to represent prescribed conditions; interactions; constraints; connectors; special engineering features; and spring and dashpot elements, reference nodes, and tracer particles in the Visualization module.

In this section:

- *Symbols used to represent prescribed conditions*
- *Symbols used to represent interactions, constraints, and connectors*
- *Symbols used to represent special engineering features*
- *Symbols used in the Visualization module*

Symbols used to represent prescribed conditions

Table 1, *Table 2*, and *Table 3* present the special graphical symbols used by Abaqus/CAE to represent loads, boundary conditions, and predefined fields, respectively. For information about symbol size and location, see *Understanding symbols that represent prescribed conditions*. For information on controlling the display of these symbols, see *Controlling the display of attributes*.

Table 1: Load symbol types and color.

Load category	Load type	Symbol	Color
Mechanical	Concentrated force	Arrow	Yellow
	Moment	Arrow	Violet
	Pressure	Arrow	Pink
	Shell edge traction	Arrow	Pink
	Shell edge moment	Arrow	Violet
	Surface traction	Arrow	Pink
	Pipe pressure	Arrow	Pink
	Body force	Arrow	Yellow
	Line	Arrow	Yellow
	Gravity	Arrow	Yellow
	Bolt load	Arrow	Yellow
	Generalized plane strain	Arrow	Yellow
	Rotational body force	Arrow	Green
	Connector force	Arrow	Yellow
	Connector moment	Arrow	Violet
	Inertia relief	Sphere with arrow	Green
Thermal	Surface heat flux	Arrow	Green
	Body heat flux	Square	Yellow
	Concentrated heat flux	Square	Yellow
Acoustic	Inward volume acceleration	Square	Yellow
Fluid	Concentrated pore fluid flow	Square	Yellow
	Surface pore fluid flow	Arrows	Blue
Electrical/Magnetic	Concentrated current	Square	Yellow
	Surface current	Square	Green
	Body current	Circle	Green
	Surface current density	Arrow	Yellow
	Body current density	Arrow	Green
	Concentrated charge	Square	Yellow
	Surface charge	Square	Green
	Body charge	Circle	Green
Mass diffusion	Concentrated concentration flux	Square	White

Load category	Load type	Symbol	Color
	Surface concentration flux	Arrow	Yellow
	Body concentration flux	Circle	Yellow

Table 2: Boundary condition symbol types and color.

Boundary condition category	Boundary condition type	Symbol	Color
Mechanical	Symmetry/Antisymmetry/ Encastre	Arrow	Orange for components applied to degrees of freedom 1–3.
			Blue for components applied to degrees of freedom 4–6.
	Displacement/Rotation	Arrow	Orange for components applied to degrees of freedom 1–3.
			Blue for components applied to degrees of freedom 4–6.
	Velocity/Angular velocity	Arrow	Sandy brown for components applied to degrees of freedom 1–3.
			Magenta for components applied to degrees of freedom 4–6.
	Acceleration/Angular acceleration	Arrow	Yellow for components applied to degrees of freedom 1–3.
			Blue for components applied to degrees of freedom 4–6.
	Connector displacement	Arrow	Orange for components applied to degrees of freedom 1–3.
			Blue for components applied to degrees of freedom 4–6.
	Connector velocity	Arrow	Sandy brown for components applied to degrees of freedom 1–3.
			Magenta for components applied to degrees of freedom 4–6.
	Retained nodal DOFs (while being defined in **Substructure generation** step)	Arrow	Blue
	Retained nodal DOFs (during substructure usage)	Cross	Blue
Electrical/Magnetic	Electrical potential	Square	Yellow
	Magnetic vector potential	Arrow	Green
Other	Temperature	Square	Yellow
	Pore pressure	Square	Yellow
	Fluid cavity pressure	Circle	Yellow
	Normalized concentration	Square	Yellow
	Acoustic pressure	Square	Yellow
	Connector material flow	Square	Yellow
	Eulerian boundary	Arrow	Green

Boundary condition category	Boundary condition type	Symbol	Color
	Eulerian mesh motion	Square	Yellow
	Submodel boundary condition	Square	Yellow
	Submodel load	Circle	Yellow

Table 3: Predefined field symbol types and color.

Predefined field category	Predefined field type	Symbol	Color
Mechanical	Translational/rotational velocity	Arrow	Sandy brown
	Stress/Geostatic stress	Circle	Blue
	Hardening	Circle	Magenta
Other	Temperature	Square	Yellow
	Field	Square	Yellow
	Material assignment	Circle	Sandy brown
	Initial state	Circle	Yellow
	Saturation	Square	Green
	Void ratio	Square	Sandy brown
	Pore pressure	Square	Magenta
	Fluid cavity pressure	Diamond	Green

Symbols used to represent interactions, constraints, and connectors

Table 1, *Table 2*, and *Table 3* present the special graphical symbols used by Abaqus/CAE to represent interactions, constraints, and connectors, respectively. For information about symbol size and location, see *Understanding symbols that represent interactions, constraints, and connectors*. For information on controlling the display of these symbols, see *Controlling the display of attributes*.

Table 1: Interaction symbol types and color.

Interactions	Symbol	Color
Surface-to-surface contact	Square	Yellow
Self-contact	Square	Yellow
Fluid cavity	Filled circle (cavity point)	Green
	Square (cavity surface)	Green
Fluid exchange	Circle (large)	Yellow
Fluid inflator	Circle (large) (cavity point)	Yellow
Model change	Square	Yellow
Cyclic symmetry	Square	Yellow
Elastic foundation	Square	Yellow
Cavity radiation	Square	Yellow
Actuation/sensor	Square	Yellow
Standard-Explicit co-simulation	Filled circle	Magenta
Incident wave	Square	Yellow
	Sphere (source point)	Yellow
	Filled circle (standoff point)	Yellow
Acoustic impedance	Square	Yellow
Surface film condition	Square	Yellow
Concentrated film condition	Square	Yellow
Surface radiation	Square	Yellow
Concentrated radiation to ambient	Square	Yellow
Pressure penetration (2D)	Arrow	Green
	Square (penetration region)	Green
Pressure penetration (3D)	Filled triangle	Green
	Square (penetration region)	Green

Table 2: Constraint symbol types and color.

Constraints	Symbol	Color
Tie	Circle	Yellow
Rigid body	Circle	Yellow
Coupling	Circle	Pink
	Lines	Gray
MPC constraint	Circle	Green

Constraints	Symbol	Color
	Lines	Green
Shell-to-solid coupling	Circle	Yellow
Embedded region	Circle	Yellow
Equation	Circle	Yellow

Table 3: Connector symbols and color.

Connector symbols	Color
Square (first points of wires)	Orange
Triangle (second points of wires)	Yellow
Orientation triad, if not the global coordinate system	Orange
Connection type label	Orange
Connector section assignment tag	Orange

Symbols used to represent special engineering features

Special graphical symbols are used by Abaqus/CAE to represent special engineering features.

Table 1: Engineering feature symbol types and color.

Engineering feature	Type	Symbol	Color
Crack	Crack front	Cross	Green
Fastener	Assembled (attachment points only)	Square	Light green
	Point based (positioning point only)	Square	Green
	Discrete	Circle	Green
		Lines	
Inertia	Point mass/inertia	Filled square	Green
	Heat capacitance	Filled square	Green
	Nonstructural mass	Filled square	Green
Springs/Dashpots	Connect two points	Spheres with connecting line	Magenta
		Label (`K`, `C`, or `K+C`)	
		Orientation triad, if not the global coordinate system	
	Connect points to ground	Sphere	Magenta
		Label (`K`, `C`, or `K+C`)	
		Orientation triad, if not the global coordinate system	
Topology and shape optimization	Design area	Diamond	Green
	Geometric restriction	Circle	Magenta
	Stop condition	Square	Yellow

Symbols used in the Visualization module

Special graphical symbols are used by Abaqus/CAE in the Visualization module.

Table 1 presents the special graphical symbols used by the Visualization module. These special symbols appear when you produce an undeformed, deformed, contour, or symbol plot of a model containing any of these components. For information on customizing the color and size of these symbols, see *Coloring all geometry in the Visualization module* and *Controlling the display of model entities*, respectively.

Table 1: Special graphical symbols used by the Visualization module.

	**Two-node dashpot elements** The Visualization module displays two-node dashpot elements using the symbol shown. You can choose to display a straight line in place of this symbol by selecting **Extra coarse** refinement for all curved edges and faces in your model. For more information, see *Refining curved edges and faces*.
	**Two-node spring elements** The Visualization module displays two-node spring elements using the symbol shown. You can customize the appearance of the curves in this symbol using the options for refining curved edges and faces. The Visualization module displays a two-node spring element as a straight line when you select **Extra coarse** refinement for all curved edges and faces in your model. For more information, see *Refining curved edges and faces*.
	**Reference nodes** The Visualization module displays the reference node for a rigid surface as a cross. You cannot customize the shape of this symbol.
	**Tracer particles** The Visualization module displays a tracer particle as a cross within a square. You cannot customize the shape of this symbol.

In addition to the special symbols the Visualization module uses to display the model components listed above, you can choose to display symbols representing results, model entities (boundary conditions, connectors, coordinate systems, and point elements), and nodal locations. For more information, see the following sections:

- *Plotting analysis results as symbols*
- *Controlling the display of model entities*
- *Controlling the display of constraints in the Visualization module*
- *Customizing node symbols*

Element and output variable support

Most Abaqus elements are supported in the Visualization module, with the following limitations:

- User-defined normals are not supported for gasket elements.
- Slide line elements applied to asymmetric-axisymmetric models are always displayed in the θ=0° plane, regardless of the specified θ for the element.
- Swept contour plots of CAXA elements may include colors that do not represent the current state at the nodes. Use of two Fourier modes improves the appearance, but for best results print the integration point values using **Report->Field Output**.

See *What kinds of elements must be generated outside the Mesh module?*, for a description of element support in the Mesh module of Abaqus/CAE.

Table 1 and *Table 2* list the output variables that are not supported by the Visualization module or the output database.

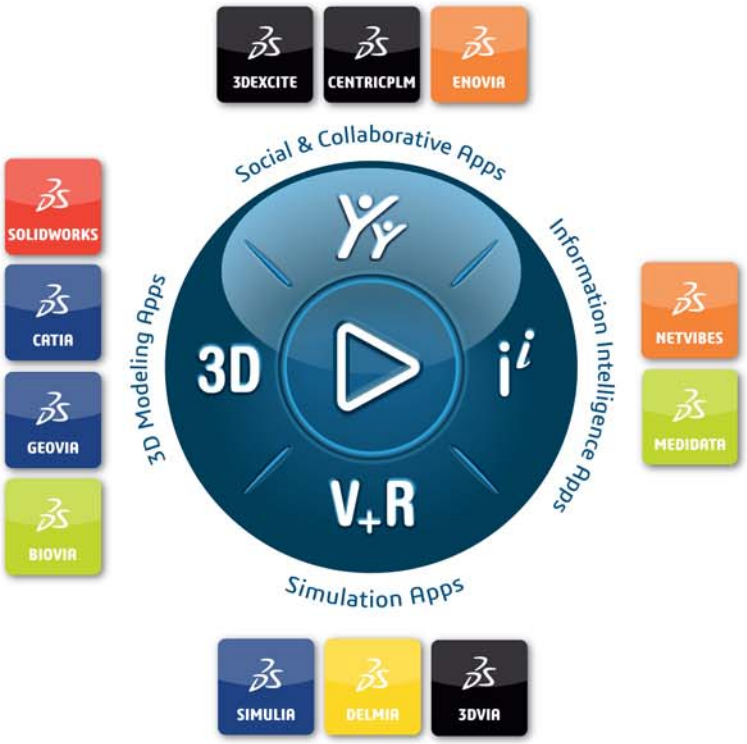
Table 1: Unsupported Abaqus/Standard variables.

CHRGS	Current values of distributed electrical charges.
CONF	Number of cracks at a concrete material point.
CRACK	Unit normal to cracks in concrete.
CS11	Average contact pressure for link and three-dimensional line gasket elements.
DG, DG*ij*	Deformation gradient.
DGP, DGP*n*	Principal stretches.
ECURS	Current values of distributed electrical currents.
FILM	Current values of **FILM* conditions.
FLUXS	Current values of distributed (heat or concentration) fluxes.
FOUND	Current values of foundation pressures.
GPU, GPU*n*	Phase angle of generalized displacements for modal analysis.
GPV, GPV*n*	Phase angle of generalized velocities for modal analysis.
GPA, GPA*n*	Phase angle of generalized accelerations for modal analysis.
LOADS	Current values of distributed loads.
MAXSS	Maximum axial stress on a section.
PHCHG	Reactive charge magnitude and phase angle for steady-state dynamic analysis.
PHE, PHE*ij*	Strain magnitude and phase angle for steady-state dynamic analysis.
PHEPG, PHEPG*n*	Electrical potential gradient vector magnitude and phase angle for steady-state dynamic analysis.
PHEFL, PHEFL*n*	Electrical flux vector magnitude and phase angle for steady-state dynamic analysis.
PHMFL	Mass flow rate magnitude and phase angle for steady-state dynamic analysis.
PHMFT	Total mass flow magnitude and phase angle for steady-state dynamic analysis.
PHPOT	Electrical potential magnitude and phase angle for steady-state dynamic analysis.
PHS, PHS*ij*	Stress magnitude and phase angle for steady-state dynamic analysis.
PPOR	Fluid, pore, or acoustic pressure magnitude and phase angle for steady-state dynamic analysis.
PRF, PRF*n*, PRM*n*	Reaction force and moment magnitudes and phase angles for steady-state dynamic analysis.

PTU, PTU*n*, PTUR*n*	Total displacement and rotation magnitudes and phase angles for mode-based steady-state dynamic analysis.
PU, PU*n*, PUR*n*	Displacement and rotation magnitudes and phase angles for steady-state dynamic analysis.
RAD	Current values of **RADIATE*.
SJP	Strain jumps at nodes. (Use the contour averaging options to obtain this variable.)
SOAREA	Area of the defined section.
SOCF	Center of the total force in the section.
SOD	Total mass flow across the section.
SOE	Total current across the section.
SOF	Total force in the section.
SOH	Total heat flux across the section.
SOM	Total moment in the section.
SOP	Total pore fluid volume flux across the section.
SS, SS*n*	Substresses for ITS elements.
SSAVG*n*	Average shell section stress component *n*.
TPFL	Total pore fluid volume flux leaving the secondary surface.
TPTL	Time integrated TPFL.
Connector output for steady-state dynamic analysis.	
Element output for user elements.	

Table 2: Unsupported Abaqus/Explicit variables.

BONDSTAT	Spot weld bond status for nodes.
BONDLOAD	Spot weld bond load for nodes.
CRACK	Crack orientations.
EIHEDEN	Internal heat energy density.
ESEDEN	Total elastic strain energy density.
SSAVG*n*	Average membrane or transverse shear stress component *n*.



Our **3D**EXPERIENCE® platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes is a catalyst for human progress. We provide business and people with collaborative virtual environments to imagine sustainable innovations. By creating virtual twin experiences of the real world with our **3DEXPERIENCE** platform and applications, our customers can redefine the creation, production and life-cycle-management processes of their offer and thus have a meaningful impact to make the world more sustainable. The beauty of the Experience Economy is that it is a human-centered economy for the benefit of all – consumers, patients and citizens.

Dassault Systèmes brings value to more than 300,000 customers of all sizes, in all industries, in more than 150 countries. For more information, visit **www.3ds.com**.

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes
17F, Foxconn Building,
No. 1366, Lujiazui Ring Road
Pilot Free Trade Zone, Shanghai 200120
China

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA

DASSAULT SYSTEMES

©2025 Dassault Systèmes. All rights reserved. **3DEXPERIENCE**, the 3DS logo, the Compass icon, IFWE, 3DEXCITE, 3DVIA, BIOVIA, CATIA, CENTRIC PLM, DELMIA, ENOVIA, GEOVIA, MEDIDATA, NETVIBES, OUTSCALE, SIMULIA and SOLIDWORKS are commercial trademarks or registered trademarks of Dassault Systèmes, a European company (Societas Europaea) incorporated under French law, and registered with the Versailles trade and companies registry under number 322 306 440, or its subsidiaries in the United States and/or other countries.