SIMULIA

3DEXPERIENCE

ABAQUS 2026 FD04
RELEASE NOTES

DASSAULT SYSTEMES

Contents

Abaqus Release Notes....2
Introduction & Spatial Modeling 3
Abaqus/CAE....4
Analysis 13
Benchmarks 86
Constraints 88
Coupled Simulation Interface 91
Elements 92
Example Problems 99
Execution 105
Getting Started with Abaqus/CAE....109
GUI Toolkit 110
Interactions 111
Keywords 129
Materials 155
Output 179
Prescribed Conditions 193
Scripting 207
Scripting Reference 209
Theory 216
User Subroutines 217
Verification 224

Trademarks and Legal Notices

Trademarks

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

Legal Notices

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

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

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

© Dassault Systèmes Americas Corp., 2026.

For a full list of the third-party software contained in this release, please go to the Legal Notices in the Abaqus 2026 HTML documentation, which can be obtained from a documentation installation, or in the SIMULIA Established Products 2026 Program Directory, which is available on *www.3ds.com*.

Abaqus Release Notes

The Abaqus Release Notes contain brief descriptions of the new features available in the Abaqus product line.

Documentation updates after the GA (General Availability; the first release on a new level) release are delivered in FD (Functional Delivery) releases. If new functionality is added in an FD release with no documentation delivery, the information is included in the related Program Directory.

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

Introduction & Spatial Modeling

This page describes recent changes in Introduction & Spatial Modeling.

2026 FD04

Modeling Surface Tension Loads with Eulerian Material Surfaces

You can now specify surface tension loads on Eulerian material surfaces in Abaqus/Explicit.

Benefits: You can model large deformation under surface tension loads.

For more information, see *Eulerian Surface Definition*, *Surface Tension Loads*, and **DSLOAD*.

2025 FD01

PDF Guides Available

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

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

For more information, see *Abaqus Introduction & Spatial Modeling Guide*.

2024 FD01

Generating a Node Set from a Surface

You can specify the name of a previously defined surface, in which case the nodes included in the surface will be assigned to the specified node set. This method can be used only to define sorted node sets.

Benefits: This surface-based node set feature makes it easier to create a node set that includes all the nodes on a predefined surface.

For more information, see *About Surfaces* and *Generating a Node Set from a Surface*.

Abaqus/CAE

This page describes recent changes in Abaqus/CAE.

2026 FD02

Creo Parametric

The Pro/ENGINEER Associative Interface is renamed to Abaqus Associative Interface for Creo Parametric. In addition, a new option, `PART_PARAMS`, supported in the `proe_abaqus.properties` file allows the **PartScope** parameters defined in Creo Parametric to be written to the parameter file (`*.par_abq`).

Benefits: This feature expands the capabilities of Abaqus/CAE.

For more information, see *Creating a CAD connection* and *Updating geometry parameters in an imported model*.

2026 FD01

Datum Creation Methods

The datum toolkit in Abaqus/CAE offers new methods for creating datum points, datum axes, and datum coordinate systems.

Benefits: You have more options for creating datums, simplifying model creation by requiring fewer intermediate datums.
You can now create a datum point by:

- Entering coordinates in a local coordinate system
- Intersecting an axis and a plane
- Offsetting from a point in a given direction

You can now create a datum axis by:

- Using the cross product of two lines
- Projecting a line onto a plane

You can now create a datum coordinate system by rotating another coordinate system by a given angle around an axis.

Background Grid SPH Particle Element Conversion

In Abaqus/CAE you can now specify section controls for smoothed particle hydrodynamics (SPH) parameters in Abaqus/Explicit analyses involving the conversion of continuum elements to SPH particles based on a uniform background grid. In earlier releases, you could specify parameters only for per-element particle conversion.

Benefits: This support allows greater flexibility when running SPH analyses in Abaqus/CAE without having to use the keywords editor.

For more information, see *Section Controls*.

2026 GA

Fluid Electric Potential Predefined Field

Abaqus/CAE now supports the definition of the fluid electric potential predefined field.

Benefits: This support allows greater flexibility to define fluid electric potential as a predefined field in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining a fluid electric potential field*.

Solid Electric Potential Predefined Field

Abaqus/CAE now supports the definition of the solid electric potential predefined field.

Benefits: This support allows greater flexibility to define solid electric potential as a predefined field in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining a solid electric potential field*.

Ion Concentration Predefined Field

Abaqus/CAE now supports the definition of the ion concentration predefined field.

Benefits: This support allows greater flexibility to define ion concentration as a predefined field in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining an ion concentration field*.

Electric Potential Electrolyte Boundary Condition

Abaqus/CAE now supports the definition of the electric potential electrolyte boundary condition.

Benefits: This support allows greater flexibility to define the electric potential electrolyte as a boundary condition in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining an electric potential electrolyte boundary condition*.

Ion Concentration Boundary Condition

Abaqus/CAE now supports the definition of the ion concentration boundary condition.

Benefits: This support allows greater flexibility to define ion concentration as a boundary condition in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining an ion concentration boundary condition*.

Distributed Ion Concentration Body Flux

Abaqus/CAE now supports the definition of distributed ion concentration body flux load.

Benefits: This support allows greater flexibility to define ion concentration as a boundary condition in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining a distributed ion concentration body flux*.

Ion Concentration Flux

Abaqus/CAE now supports the definition of ion concentration flux load.

Benefits: This support allows greater flexibility to define ion concentration as a boundary condition in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining an ion concentration flux*.

Fluid Surface Current

Abaqus/CAE now supports the definition of fluid current densities over a surface in a coupled thermal-electrochemical analysis.

Benefits: This support allows greater flexibility for modelling fluid surface current as a load in Abaqus/CAE without having to use the keywords editor.

For more information, see *Defining a fluid surface current*.

Support for Additional Degrees of Freedom in Tie Constraints

You can specify tie constraints on additional degrees of freedom, including temperature, pore pressure, electric potential in solid, electric potential in fluid, and ion concentration in fluid.

Benefits: This support allows you to specify more degrees of freedom for constraining the surfaces.

For more information, see *Defining tie constraints*.

Support for Uniform Coupling

You can use Uniform as the coupling type while defining the coupling constraints.

Benefits: This support allows you to specify a Uniform coupling type to couple nondisplacement degrees of freedom.

For more information, see *Defining coupling constraints*.

Support for Defining a Coupled Thermal-Electrochemical Procedure in Abaqus/CAE

Abaqus/CAE now supports the coupled thermal-electrochemical procedure in the GUI.

Benefits: This support allows you to define and modify a coupled thermal-electrochemical procedure interactively in Abaqus/CAE.

For more information, see *Configuring a coupled, thermal electrochemical procedure*.

Support for Defining a Fully Coupled Thermal-Electrochemical Structural Procedure in Abaqus/CAE

Abaqus/CAE now supports the coupled thermal-electrochemical structural procedure in the GUI.

Benefits: This support allows you to define and modify a coupled thermal-electrochemical structural procedure interactively in Abaqus/CAE.

For more information, see *Configuring a fully coupled, thermal electrochemical structural procedure*.

Support for Defining a Fully Coupled Thermal-Electrochemical Structural Pore Pressure Procedure in Abaqus/CAE

Abaqus/CAE now supports the coupled thermal-electrochemical procedure in the GUI.

Benefits: This support allows you to define and modify a coupled thermal-electrochemical structural pore pressure procedure interactively in Abaqus/CAE.

For more information, see *Configuring a fully coupled, thermal electrochemical structural pore pressure procedure*.

Support for Defining a Surface Film Condition on the Evolving Face of the Element

Abaqus/CAE now supports the surface film condition on the evolving faces of an element.

Benefits: This support provides more flexibility to define the surface film condition on the evolving faces of elements.

For more information, see *Defining a surface film condition interaction*.

Support for Defining a Surface Radiative Interaction on the Evolving Face of the Element

Abaqus/CAE now supports the surface radiative interaction on the evolving faces of an element.

Benefits: This support provides more flexibility to define the surface radiative interaction on the evolving face of elements.

For more information, see *Defining a surface radiative interaction*.

Support for Gap Diffusivity Contact Interaction Property in Abaqus/CAE

Abaqus/CAE now supports the gap diffusivity contact interaction property in the GUI.

Benefits: This support allows you to define and modify a gap diffusivity contact interaction property interactively in Abaqus/CAE.

For more information, see *Specifying gap diffusivity for electrical contact property options*.

Extended Support for Gap Electrical Conductance Contact Interaction Property in Abaqus/CAE

Abaqus/CAE now supports additional options for the gap electrical conductance contact interaction property in the GUI.

Benefits: You can now specify the additional options of `Type= solid/liquid` and other properties for these types. For liquid electrolyte, additional ion concentration dependent data is also supported.

For more information, see *Specifying gap conductance for electrical contact property options*.

Extended Support for Gap Heat Generation Contact Interaction Property in Abaqus/CAE

Abaqus/CAE now supports additional options for the gap heat generation contact interaction property in the GUI.

Benefits: You can now can specify the following options:

- `mechanical Interaction Fraction`
- `mechanical Interaction Weighting Factor`
- `electrical Conduction Solids Fraction`
- `electrical Conduction Solids Weighting Factor`
- `electrical Conduction Liquids Fraction`
- `electrical Conduction Liquids Weighting Factor`

For more information, see *Specifying heat generation for thermal contact property options*.

Support for Defining Step Control in Abaqus/CAE

Abaqus/CAE now supports step controls in the GUI.

Benefits: You can now define and modify step controls interactively in Abaqus/CAE.

For more information, see *Understanding step control*.

Support for Follower Parameter of Pre-tension Sections

Abaqus/CAE now extends support for the follower parameter in pre-tension sections.

Benefits: You can now define a varying normal for a pre-tension section in Abaqus/CAE, indicating that the normal is updated when performing a large-displacement analysis.

For more information, see *Creating and editing bolt loads*.

View Cut along a Path Plug-in

Abaqus/CAE now supports moving a cut plane through the model along a user-defined point-list or node-list path in the Visualization module, via an included plug-in.

Benefits: You can now move a cut plane along a user-defined path, dynamically modifying the view cut to different positions along the path. In addition, you can further rotate the view cut at the given position with orthogonal axes to arrive at a desired view cut.

For more information, see *View cut along a path*.

2025 FD01

Animation File Formats

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

Benefits: New file formats offer improved compatibility with modern media players and editing software, as well as HTML5 and common applications for producing presentations. The file size can be significantly smaller, with no appreciable loss of quality.

You can save MP4 animations with the MPEG-4 codec. If you have an installation of the FFmpeg program that supports H.264 encoding on your system, you can save MP4 animations using the H.264 codec.

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

PDF Guides Available

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

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

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

2025 GA

AutoCAD DXF Sketch Import

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

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

For more information, see *Importing sketches*.

Wear Modeling in Abaqus/CAE

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

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

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

Rotordynamic Loads in Abaqus/CAE

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

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

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

Step Dependence for Fluid Inflator and Fluid Exchange

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

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

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

Modeling Cracks and Seams

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

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

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

2024 FD03

State Space Solver for Transient Modal Dynamic Analysis

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

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

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

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

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

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

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

2024 GA

Python 3

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

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

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

Render Styles in Plots for Continuum Particle Elements

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

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

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

Step-Dependent General Contact

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

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

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

Contact Mass Scaling

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

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

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

Channel and Hat Section Profiles

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

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

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

Beam Section Offset for *BEAM SECTION

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

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

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

Rotational Couplings

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

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

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

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

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

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

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

Analysis

This page describes recent changes in Abaqus Analysis.

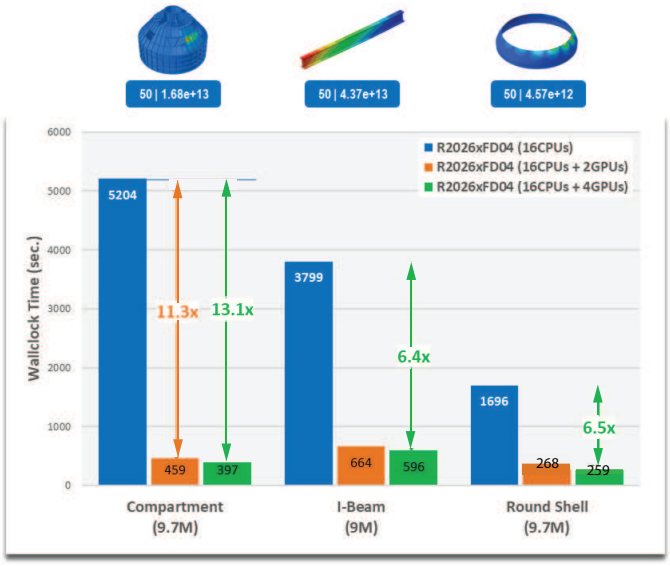
2026 FD04

GPU-Based Subspace Iteration Eigensolver

Performance of the subspace iteration eigensolver is improved significantly on systems with supported GPGPUs.

Benefits: You can reduce the run time of linear buckling analyses by leveraging GPGPU acceleration. When running on a single compute system using fewer than 32 CPU cores, one or more GPGPU cards can be used to accelerate the subspace iteration eigensolver for substantial speedups.

For example, the figure below highlights the performance gains achieved by the new subspace-based iteration solver across three representative buckling benchmark models, where each model has over 9M DOFs requiring over 1×10^{13} floating-point operations to factorize the global stiffness matrix. The performance data was obtained on a machine with Intel Xeon 6442Y processors (Sapphire Rapids, 2.6 GHz, 48 cores), 1700 GB RAM, and 4x NVIDIA H100 (96 GB/GPU).



The blue bars show the run time of Abaqus/Standard using the subspace iteration solver in shared memory parallel mode with 16 CPU cores only, whereas the orange and green bars correspond to the GPU-based subspace iteration solver with 2 GPUs and 4 GPUs, respectively. The results demonstrate that the GPU-based subspace iteration solver shows exceptional scaling up to 4 GPUs—with 13x speedup in the aerospace compartment model. Moreover, the GPU speedup increases as the number of subspace iterations increases, demonstrating that the new subspace iteration solver is especially efficient for problems requiring many iterations to converge.

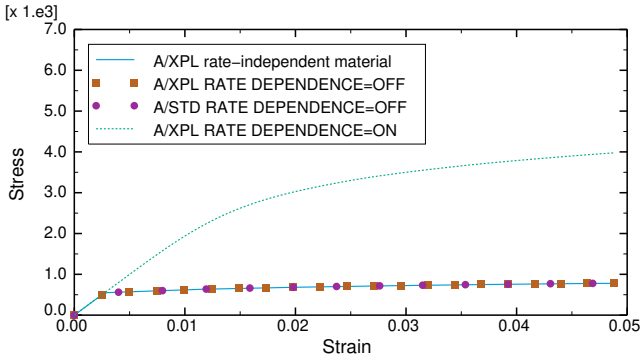
For more information, see *Parallel Execution in Abaqus/Standard*.

Controlling Material Rate Dependence in Explicit Dynamic Simulations

You can now control whether or not to consider material rate–dependence in an explicit dynamic step or coupled dynamic thermal-stress step.

Benefits: You can more easily run explicit dynamic simulations with rate-independent or long-term material response.
In earlier releases, you needed to remove rate-dependent properties from the material definitions to ignore material rate-dependence in explicit dynamic steps. Now, you can set the RATE DEPENDENCE parameter on the *DYNAMIC or *COUPLED TEMPERATURE DISPLACEMENT options equal to ON or OFF to include or ignore these effects. This can be helpful for explicit dynamic procedures simulating long-duration quasi-static processes or for applying long-term preloads preceding sudden dynamic events.

The figure below illustrates the agreement of the simulations ignoring material rate-dependent behaviors with reference simulations where the corresponding material definitions do not contain rate-dependent properties.



For more information, see *Explicit Dynamic Analysis* and *Fully Coupled Thermal-Stress Analysis*.

Supporting Multiple Linear Load Cases in a Submodel Analysis

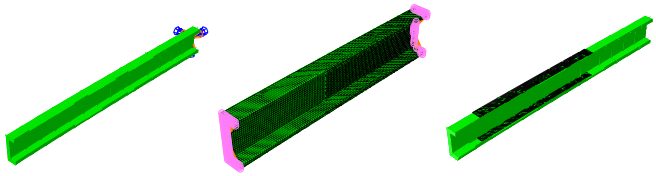
A step within a submodel can now refer to a step of a global model containing multiple linear load cases.

Benefits: You no longer need to create a separate perturbation step for each linear load case in the global model and the submodel, which reduces memory and execution time.

A multiple linear load case analysis is equivalent conceptually to a multiple-step analysis in which the load case definitions are mapped to consecutive perturbation steps. However, a multiple linear load case analysis might be more efficient than an equivalent multiple-step analysis.

Submodeling is used to study a local region in detail using results from a previously run larger model to drive conditions at submodel cuts. When a global model analysis contains multiple linear load cases, you can now use equivalent multiple linear load cases in the submodel analysis to study the behavior in the submodel region.

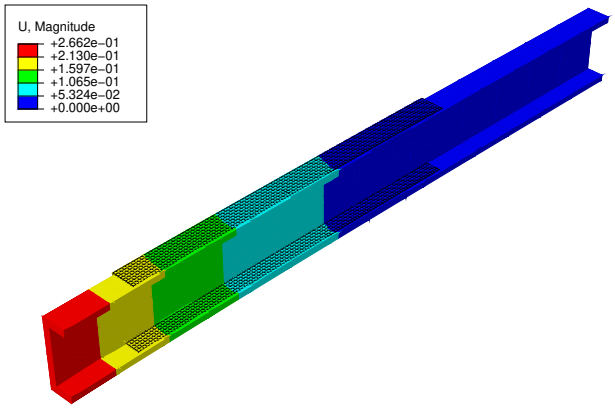
The figure below shows a global model in which a cantilever beam is modeled with beam elements (left), a submodel meshed with solid elements (center), and an overlay of the global model and submodel (right).



The general submodeling capability is used such that displacement and rotation results from the global model drive displacements and rotations via internally generated distributed couplings. The global and submodel simulations each have a three-step sequence as follows:

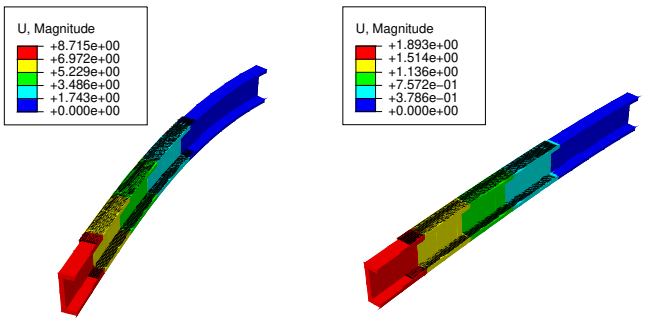
1. A linear perturbation step with two transverse load cases.
2. A general step with tensile loading and geometric nonlinearity considered.
3. Another linear perturbation step with the same two transverse load cases as in Step 1.

Step 2 is unaffected by Step 1 because Step 1 is a perturbation Step, but it causes Step 1 and Step 3 to have different base states. Step 2 applies axial tensile loading in a general step with nonlinear geometry, which causes stretching as shown in the figure below. Here, the displacement results from the global model and submodel are overlaid and in good agreement.

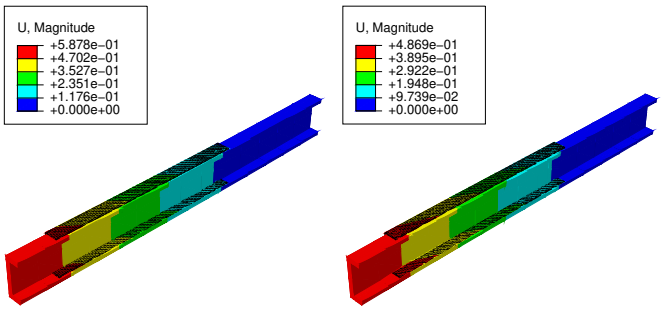


Step 1 involves a linear load case with transverse loading in the flange direction and another linear load case in the web direction. Step 3 has loading identical to that in Step 1, but it has a different base state than Step 1 due to inclusion of the load from nonlinear Step 2.

Below we compare perturbation solutions for the first and second load cases of Step 1 and Step 3, with results from the global model and submodel overlaid. The following figure shows displacement plots for the first load case of perturbation Step 1 (left) and Step 3 (right), causing bending of the beam:



The following figure shows displacement plots for the second load case of perturbation Step 1 (left) and Step 2 (right), causing bending of the beam.



The results for Step 1 and Step 3 differ significantly, even though the perturbation loads match, due to different base states. The base state for Step 1 is stress free. The base state for Step 3 includes the axial tensile stress resulting from Step 2, which leads to stress stiffening that reduces the amount of bending compared to the stress-free base state.

For more information, see *Using Multiple Linear Load Cases in a Submodel*.

Time Synchronization Between the Global Model and Submodel

You can now control the default behavior of time synchronization between the submodel analysis and the global model analysis.

Benefits: You can more easily understand the submodel setup.

By default, Abaqus now does the following:

- Associates the *i*th general step of a submodel with the *i*th general step of the global model for general submodeling.

  Abaqus issues an error message if the submodel has more general steps than the global model.

- Associates the *j*th perturbation step after the *i*th general step with the *j*th perturbation step after the *i*th general step of the global model.

  Abaqus issues an error message if the submodel has more perturbations steps than the global model after the corresponding general steps.

For more information, see *Referring to the Step and Increment of the Global Model*.

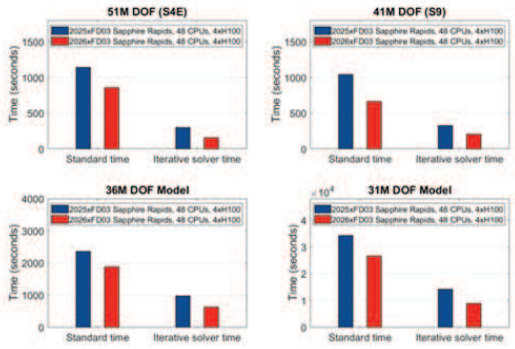
2026 FD03

GPU AMG Iterative Solver

Performance is improved significantly for the multi-GPU iterative solver, depending on model details and hardware configuration.

Benefits: Analysis run time is reduced.

Benchmark testing was conducted across four representative models to compare the performance of the GPU iterative solver in 2025 FD03 and 2026 FD03. All simulations were executed on a system equipped with 48-core Sapphire Rapids processors and four H100 GPUs. The results demonstrate a substantial reduction in total analysis time with the new implementation.



AMS GPU Acceleration for Frequency Extraction

AMS eigensolver performance for frequency extraction analysis is improved on systems equipped with high CPU core counts and NVIDIA GPUs with more than 80GB of memory.

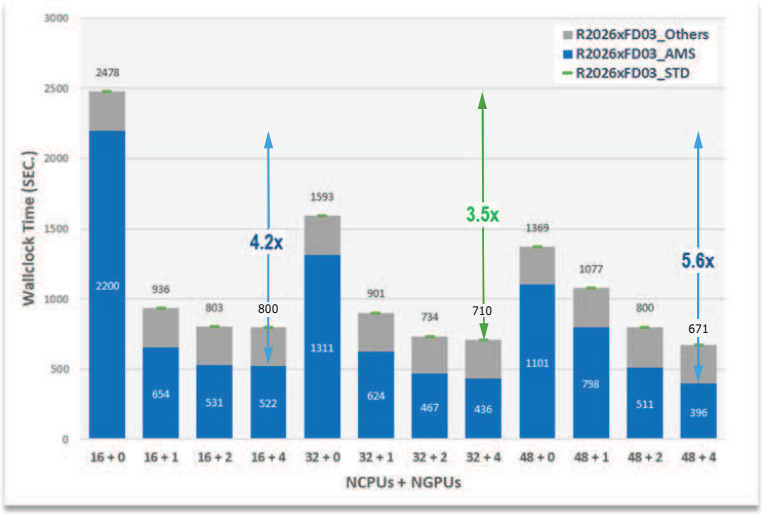
Benefits: The AMS eigensolver delivers significant speedups for frequency extraction analysis for large-scale bulky models and vehicle body models requiring over 20,000 eigenmodes.

With the advent of the high-performance NVIDIA GPUs featuring more than 80 GB memory, the GPU acceleration strategy for the AMS eigensolver has been redesigned. All three AMS phases now more effectively utilize large-memory GPUs, significantly improving overall GPU acceleration performance.

For the AMS reduction phase, we optimize the dynamic scheduling strategy to better saturate GPUs with larger workloads. In earlier releases, GPU acceleration performance often peaked at 16 CPUs when using multiple GPUs due to scheduling overhead costs. Now, an enhanced dynamic scheduling strategy efficiently distributes

more computational tasks to the large-memory GPUs, particularly on systems with higher CPU core counts (> 32 CPUs).

As an example, the figure below illustrates the GPU acceleration performance in 2026 FD03 for a 7.6M DOF stator model requiring 2.3e+15 floating-point operations to factorize four system operators (stiffness, mass, viscous, and structural damping matrices). The performance data was obtained on a machine equipped with Intel Xeon 6442Y processors (Sapphire Rapids, 2.6 GHz, 48 cores), 1700 GB RAM, and 4x NVIDIA H100 (96 GB) GPUs.

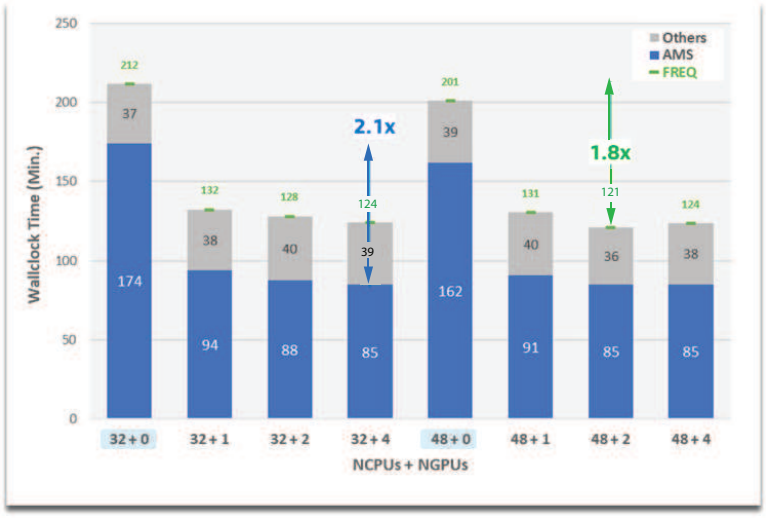


The blue columns represent the AMS run time, the gray columns represent the run time for the other phases (excluding the AMS run time from the total Abaqus/Standard time), and the small green dashes represent the total Abaqus/Standard run time. Abaqus/Standard using the AMS eigensolver demonstrates strong GPU-acceleration scaling up to 48 CPUs. Peak performance for this model is achieved using a configuration of 48 CPUs and 4 GPUs.

In addition, the GPU acceleration performance of the AMS eigensolver is further improved for a vehicle body model extracting more than 20,000 eigenmodes; this workload was previously constrained by GPUs with less than 80 GB of memory.

A new GPU dense eigensolver within the AMS reduced eigensolution phase can process large dense eigenproblems with significantly lower memory requirements. In addition, a new GPU acceleration for modal matrix projection including structural and viscous damping matrices and right-hand-side vectors is introduced.

The figure below shows the GPU acceleration performance for a 56M DOF vehicle trim body model. The same machine is used as for the stator model. The AMS eigensolution achieves a 2.1x speedup using 32 CPUs and 4 GPUs compared to a 32 CPU-only run.



For more information, see *Parallel Execution in Abaqus/Standard*.

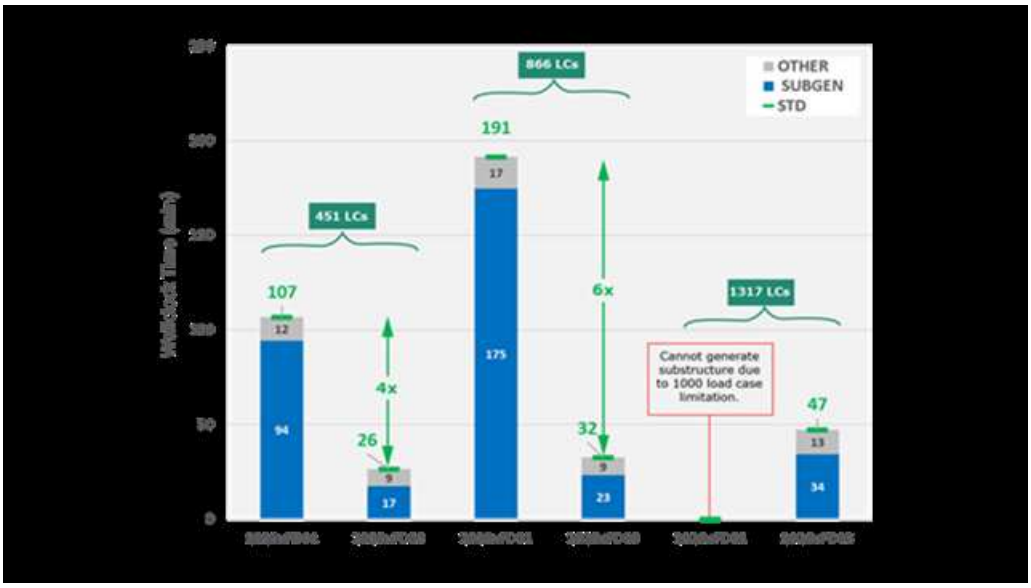
Substructure Load Case Generation

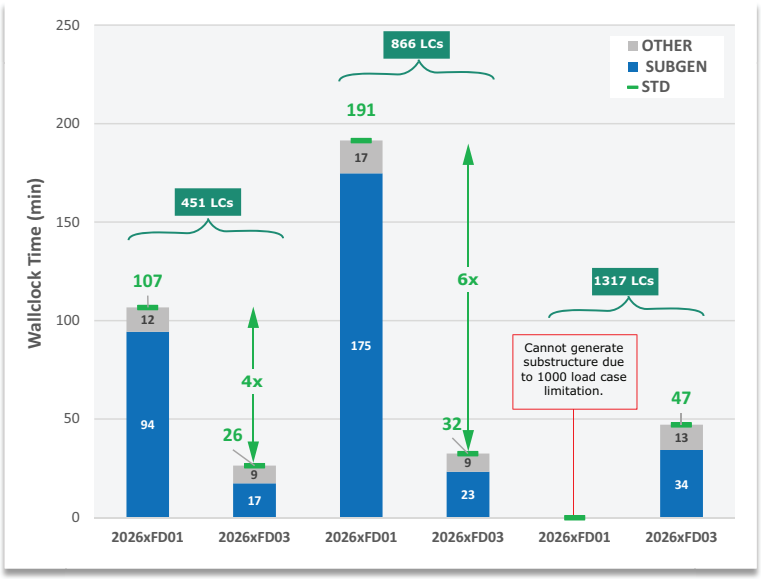
You can now generate substructures with more than 1000 load cases.

Benefits: Substructure generation analyses run faster, and you can generate all substructure load cases required in workflows using substructures.

In earlier releases, you could generate no more than 1000 load cases in a single substructure generation step.

The benchmark engine model demonstrates the enhancements. The finite element mesh has 11 M degrees of freedom. Generation of a substructure with 78 retained nodal degrees of freedom and 227 retained eigenmodes is performed for three cases with increasing numbers of substructure load cases. The analysis is run on a machine with a 64-core Emerald Rapids processor and 512 GB memory. In the figure below, STD is the total analysis time, SUBGEN (blue portions of the bars) is the time substructure generation time, and OTHER time (gray portions of the bars) includes eigenvalue extraction using the AMS eigensolver and other analysis phases.





For more information, see *Generating Substructures*.

Specifying Time-Dependent Tool Orientations and Bead Dimensions in Trajectory-Based Eigenstrain Additive Manufacturing Simulations

In a trajectory-based eigenstrain additive manufacturing simulation, you can now specify a time-dependent tool orientation and a time-dependent height and width of the deposition bead.

Benefits: You can more accurately model additive manufacturing simulations.

You specify the tool orientation and the bead dimensions as part of the event series definition used to describe the tool trajectory during the additive manufacturing simulation.

For more information, see *Eigenstrain-Based Analysis of Additive Manufacturing Processes*.

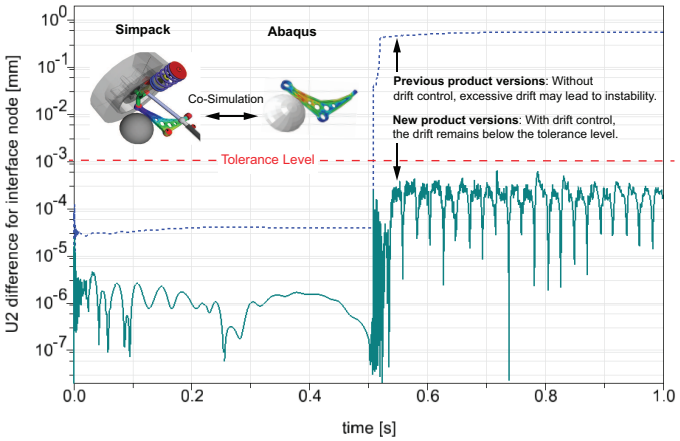
Abaqus-to-Simpack Co-Simulation

An Abaqus-to-Simpack co-simulation now computes initial interface forces to establish acceleration compliance of the transient FE and MBD subsystems. Algorithmic enhancements also minimize displacement drift.

Benefits: Setting up an Abaqus-to-Simpack co-simulation is easier, particularly when it starts from a nonresting state. You can also obtain more accurate and robust solutions.

Abaqus-to-Simpack co-simulation uses the enhanced subcycling method that ensures velocity compatibility at the interface. However, this method can lead to a displacement drift when the co-simulation uses a large coupling step size. Now, the co-simulation monitors the displacement drift during the solution phase and employs a Baumgarten type of approach to compensate for the displacement drift. In some cases, the coupling step size is also reduced automatically to keep the drift small. This leads to a more accurate solution and more robust coupling of both transient subsystems.

You must use Simpack 2027 to benefit from these enhancements.



Coupled Simulation Guide

A new Coupled Simulation Guide describes the SIMULIA Co-Simulation Services. It provides details about the SIMULIA Co-Simulation Services API and Co-Simulation Configuration file and examples of how to embed the services into your application.

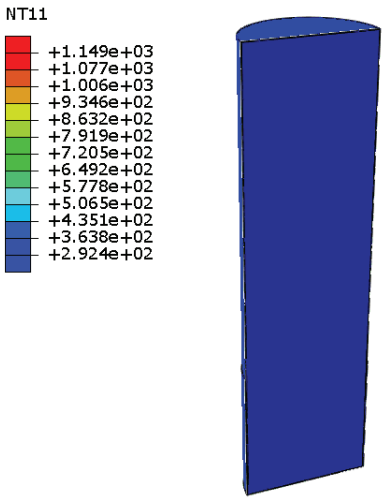
Benefits: You can more easily embed the SIMULIA Co-Simulation Services into your application.
The SIMULIA Co-Simulation Services expands the simulation capabilities of Abaqus by allowing you to perform coupled multiphysics and multiscale simulations between your application and Dassault Systèmes physics solvers and partner applications for importing fields in sequential coupled simulations, solving complex multiphysics co-simulations (for example, fluid-structure interaction), and multidomain coupling.

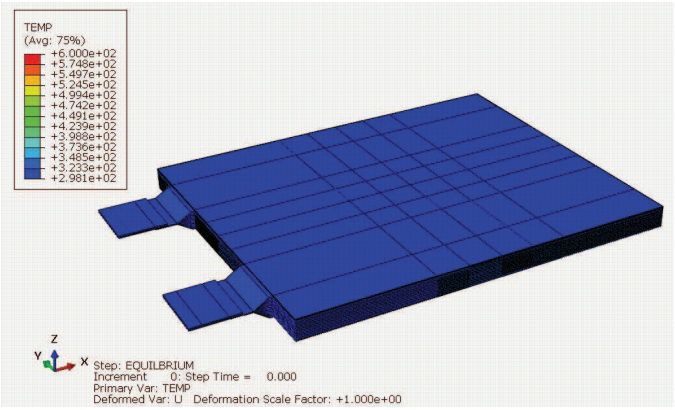
This is an advanced capability. You must have sufficient programming skills, in-depth knowledge of the Dassault Systèmes application being coupled, and familiarity with the limitations of the segregated solution methods for co-simulation. We encourage contacting customer support to discuss your project before embarking on any programming efforts.

Modeling Thermal Runaway in Lithium Ion Batteries

You can use coupled thermal-electrochemical analysis and fully coupled thermal-electrochemical-structural analysis to model thermal runaway in lithium ion batteries.

Benefits: You can now use Abaqus to more effectively assess battery safety by considering thermal runaway. Thermal runaway is a very rapid, self-feeding, uncontrolled event that releases large amounts of heat due to the exothermic reactions of the various battery components, such as the electrolyte, SEI layer, and decomposition of the lithium at the electrodes. The animations below show the rapid rise in temperature in cylindrical and pouch cells during a thermal runaway event.





For more information, see *Modeling Thermal Runaway in Batteries*.

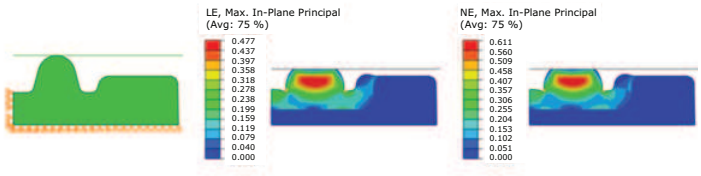
2026 FD02

Adjoint Sensitivity Analyses

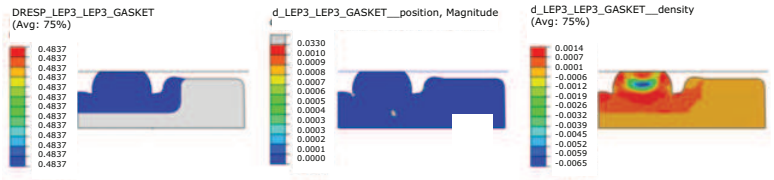
You can now specify strain, logarithmic strain, and nominal strain as design responses in adjoint design sensitivity analyses, and you can use hybrid elements in transient dynamic adjoint shape sensitivity analyses.

Benefits: You have more flexibility in design optimization workflows.
You can now design structures based on optimizing different strain measures. You can also include incompressible material response (for example, rubberlike materials) in the design domain for transient dynamic shape optimization.

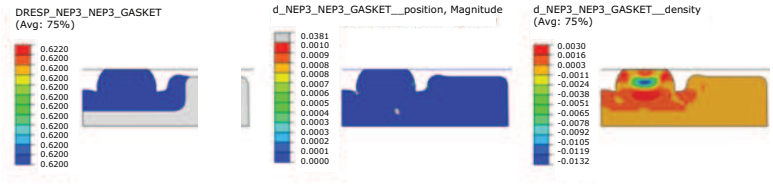
The figure below shows the loading, boundary conditions, and logarithmic and nominal principal strains for a rubber gasket sealing simulation using the plane strain assumption and hybrid elements due to nearly incompressible material response.



The figure below shows the maximum LEP3 as a design response, along with the corresponding sensitivities with respect to the nodal shape design variables and elemental topology design variables, respectively.



The figure below shows the corresponding results with the maximum NEP3 as the design response.



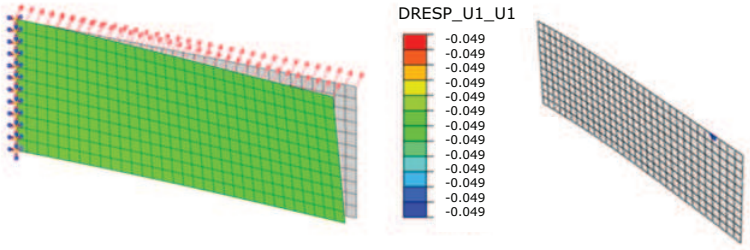
For more information, see *Specifying Design Responses in a Static or an Eigenfrequency Analysis*, *Specifying Design Responses in a Transient Dynamic Analysis*, and *Adjoint Design Sensitivity Analysis Overview*.

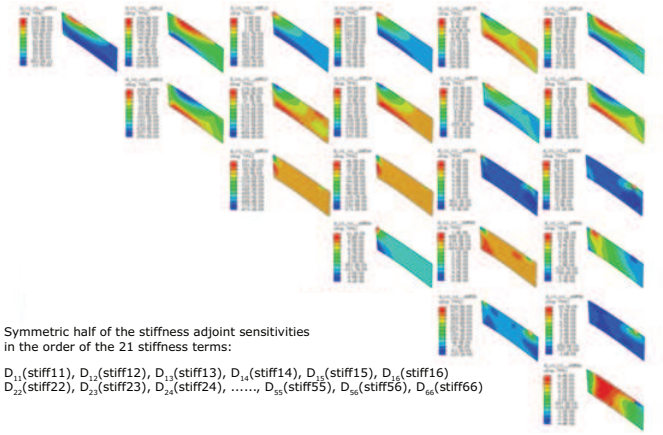
ABD Stiffness Sensitivity for Shell Elements Defined Using a General Shell Section

In Abaqus/Standard you can now specify the components of the section stiffness (ABD) matrix for shell general sections as design variables in an adjoint design sensitivity analysis.

Benefits: You can use the ABD stiffness sensitivities to optimize the design of composite shell structures. Adjoint design sensitivities with respect to the components of the section stiffness (ABD) matrix allow you to carry out generative design for the best possible overall stiffness properties for a composite shell structure.

The figure below shows the setup for an adjoint ABD design sensitivity analysis for a shell structure subjected to a single static load case and with nodal displacement, U1, of one of the nodes as the design response. Each of the 21 stiffness components of the ADB matrix for each element is considered to be a design variable. The second figure shows the adjoint sensitivities of the design response with respect to each of the 21 design variables.



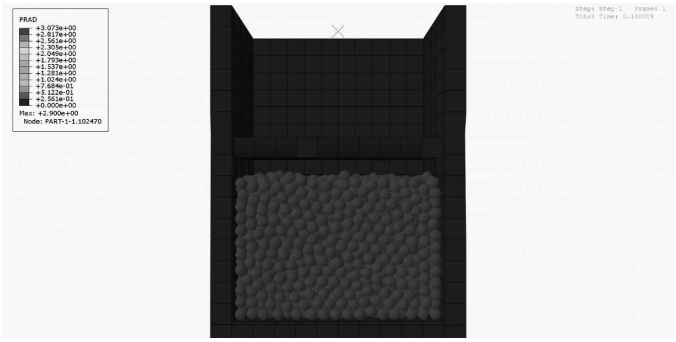


Particle Swelling for Discrete Element Analysis

Abaqus/Explicit now supports swelling of discrete particles.

Benefits: You can better understand the changes in media properties in granular media undergoing swelling. Granular media properties such as porosity depend on particle size. When the particle size changes due to swelling, media properties such as porosity also change.

The animation below shows the granular media behavior with reversible swelling that is modeled using discrete element method particles with user-defined swelling behavior.



For more information, see *Discrete Element Method*.

2026 FD01

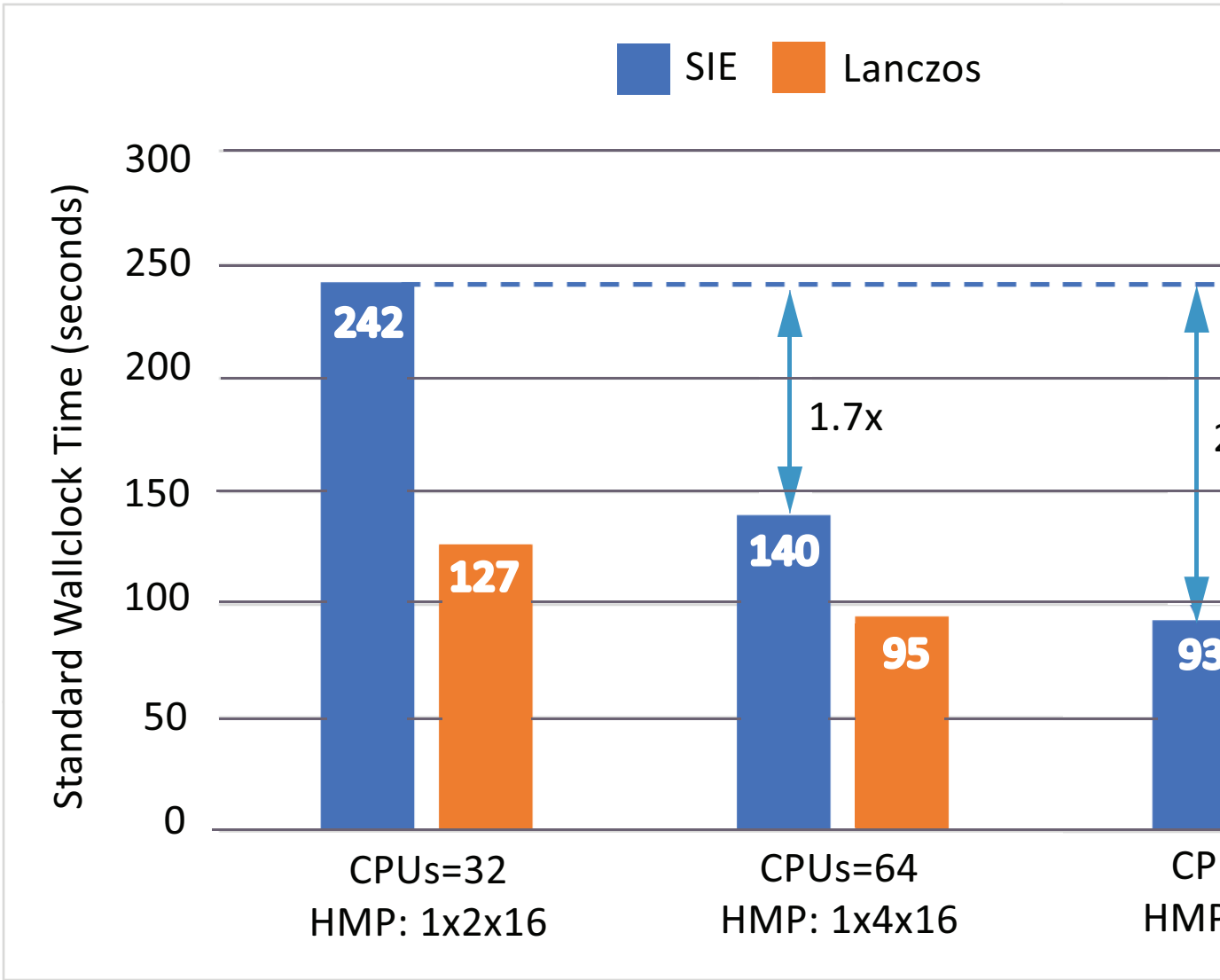
Subspace Iteration Eigensolver Functionality Enhancements

You can now perform frequency extraction analysis using MPI-based parallel execution of the subspace iteration eigensolver.

Benefits: You can see significant performance improvements for the frequency extraction of coupled structural-acoustic problems.

Frequency extraction analysis using the subspace iteration eigensolver can run in hybrid memory parallel mode for coupled structural-acoustic problems and support residual model computation and damping projection for subsequent modal analysis.

The figure below illustrates the enhanced performance of the subspace iteration eigensolver using a coupled structural-acoustic analysis with ACOUSTIC COUPLING=PROJECTION for a Neon model with 2.5 million degrees of freedom.



The analysis is run in hybrid memory parallel (HMP) mode using up to two compute nodes, each of which has an Intel Xeon 3.5 GHz Emerald Rapids processor with 2 x 32 cores and 512 GB memory.

In each of these runs, we request 50 modes up to a maximum frequency of 300 Hz, specify an acoustic range factor of 2.0, request residual modes, and turn on damping projection.

We compute a total of 112 modes (50 in the acoustic domain + 50 in the structural domain + 12 residual modes) with both the Lanczos and subspace iteration eigensolver.

The subspace iteration eigensolver exhibits a 1.7x and 2.6x speedup as we go from 32 cores (1 x 2 x 16 host configuration) to 64 cores (1 x 4 x 16 host configuration) and 128 cores (2 x 4 x 16 host configuration), respectively. In addition, the subspace iteration eigensolver is comparable to the Lanczos eigensolver on two compute nodes.

Subspace iteration also delivers good accuracy in subsequent modal analyses.

For more information, see *Natural Frequency Extraction* and **FREQUENCY*.

Performance Improvement from Alternate Contact Convergence Criteria

Abaqus/Standard now modifies contact-related convergence checks automatically when you specify an alternate convergence check.

Benefits: The solution time might be reduced.

The alternate convergence checks were introduced in an earlier release; however, in earlier releases they did not affect contact-specific convergence thresholds. Now, contact convergence is influenced when you set the CONVERGENCE CHECK parameter equal to MODERATE, RELAXED, or CONCEPT DESIGN also influence on the **CONTROLS* option.

The table below shows the effect on the total number of iterations on three contact models due to modified contact convergence checks. Each row in the table corresponds to a different setting of the convergence checks.

Model	Convergence Check Setting	Total # of iterations in earlier releases	Total # of iterations in 2026x FD01	Reduction in total # of iterations
1	STRICT (default)	857	857	No change
	MODERATE	760	732	3.7%
	RELAXED	801	766	4.5%
	CONCEPT DESIGN	784	749	4.5%
2	STRICT (default)	25,361	25,361	No change
	MODERATE	24,991	12,760	49.0%
	RELAXED	24,852	14,953	39.9%
	CONCEPT DESIGN	24,943	14,393	42.3%
3	STRICT (default)	556	556	No change
	MODERATE	494	494	0%
	RELAXED	380	380	0%
	CONCEPT DESIGN	380	380	0%

No changes to the convergence tolerance occur across versions with the default STRICT setting in effect; therefore, the total number of iterations remains the same for all models that use this setting. The impact for models with the MODERATE, RELAXED, and CONCEPT DESIGN settings is as follows:

- Small percentage reductions in the number of iterations for Model 1, with the DESIGN setting requiring the fewest iterations.
- Larger percentage reductions in the number of iterations for Model 2, with the MODERATE setting requiring the fewest iterations. This shows that further relaxation of convergence thresholds does not always lead to convergence improvement, which is somewhat counterintuitive. You might occasionally observe the STRICT setting leading to better convergence than the other settings.
- No effect on the number of iterations for Model 3, indicating that contact-specific convergence checks do not control convergence for this model.

In terms of accuracy, Model 2, which exhibits the largest impact compared to earlier releases, was examined. There were no differences in peak Mises stresses or peak contact stresses due to the changes for each of the alternate convergence check settings. For this model, both in earlier releases and in this release, the peak contact stresses were within 3% and the peak Mises stresses were within 0.1% of each other for the different settings. In general, you must ensure that solution accuracy from these alternate checks is acceptable on its own or in comparison with the default STRICT setting.

For more information, see *Modifying the Default Convergence Checks* and **CONTROLS*.

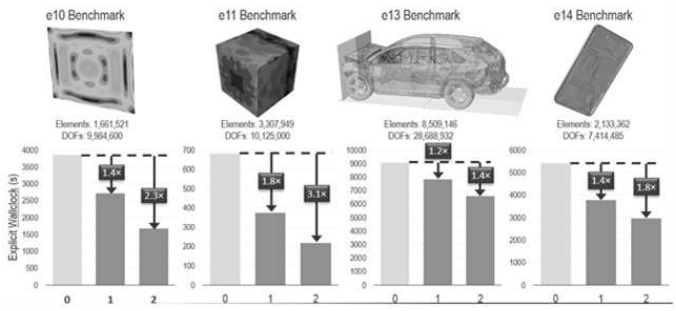
GPU Acceleration in Abaqus/Explicit

For NVIDIA GPUs on Linux systems, Abaqus/Explicit now supports GPU acceleration for a wide variety of dynamic and quasi-static structural simulations.

Benefits: Performance is significantly improved, particularly when running large models.
Abaqus/Explicit takes advantage of the compute acceleration on NVIDIA GPUs on Linux systems. The solver can be executed either on a single node with one or more GPU devices or on a multi-node cluster with multiple GPUs. The implementation emphasizes results accuracy compared to CPU-only execution.

GPU execution supports most features, with exceptions for user subroutines, CEL, ALE, and particle methods. The acceleration from GPU usage depends on the computational intensity of the model and data flow between the CPU cores and the GPU devices. Therefore, the performance benefit is workflow dependent. Typically, large models with hundreds of thousands of elements to multiple millions of elements can run significantly faster by using GPU acceleration.

The figure below shows several examples of Abaqus/Explicit GPU acceleration over a range of workflows with the addition of 1 and 2 GPUs.



The benchmarks e10 and e11 are dominated by element computation. The production level models e13 (a crash simulation) and e14_DropTest_v0 (a phone drop test) demonstrate a range of performance gains with the addition of 1 or 2 A100 GPUs to a 48-core Icelake node. On the challenging side, the crash model (whose compute cost is dominated by contact) shows a moderate performance gain, whereas the drop test (which still has a decent number of contacting parts) shows a significant gain. The other two examples show an excellent speed up that can be achieved for element computation-dominated workflows.

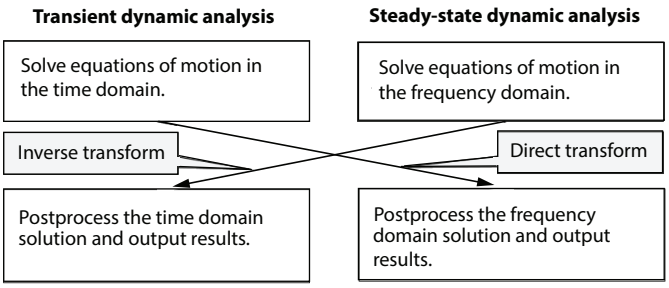
For more information, see *Parallel Execution in Abaqus/Explicit*.

Fourier Transform in Linear Dynamics Procedures

You can now use a Fourier transform in linear dynamics procedures to convert the mechanical response from the time domain into the frequency domain and vice versa.

Benefits: The new functionality improves the durability and NVH workflows.
You can now obtain a:

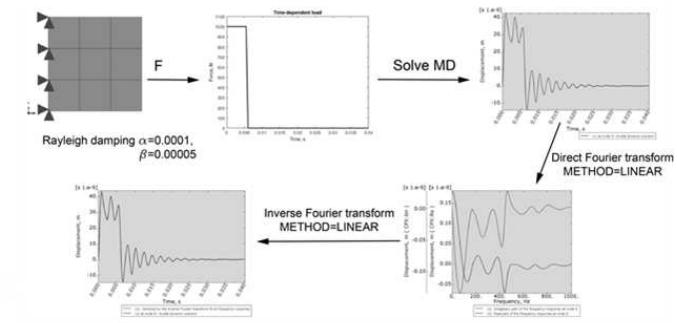
- Frequency response from a time-domain response using a direct Fourier transform.
- Time-domain response from a frequency response using an inverse Fourier transform.



Abaqus applies the Fourier transform on the modal level (to the generalized displacements). Therefore, the computational complexity of the Fourier transform is independent of the number of degrees of freedom in the model. Instead, the computational complexity of the implemented Fourier transform algorithms depends on the number of eigenmodes, time, and frequency points.

The example below illustrates validation of the direct and inverse Fourier transforms. First, a modal dynamics problem was solved in the time domain. Then, a direct Fourier transform was applied that results in the frequency

response (in the frequency domain). Finally, an inverse Fourier transform was applied that restores the original motion in the time domain. The plots shown demonstrate horizontal displacement of the point at which the force is applied.



For more information, see *About Dynamic Analysis Procedures*.

SPH Performance Improvements and Memory Usage Reduction When Modeling Liquids

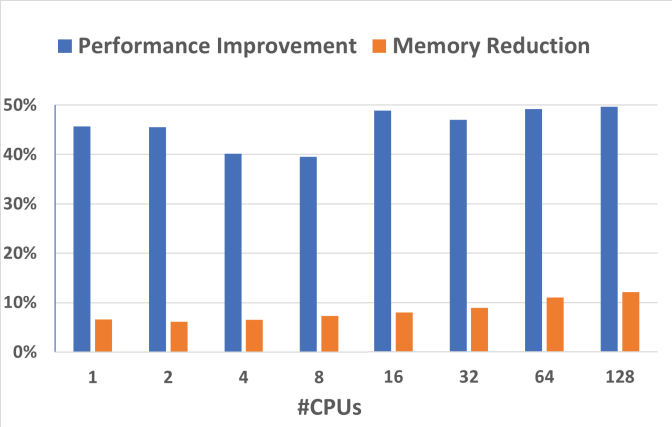
You can now conduct SPH simulations in Abaqus/Explicit with a new particle-to-particle pairwise interaction-based algorithm.

Benefits: Performance is improved and memory usage is reduced for SPH simulation in liquid modeling.

In earlier releases of Abaqus/Explicit, the SPH method used an algorithm based on a pseudo element type architecture: a SPH particle and its surrounding cloud particles within the smoothing length range formed a pseudo element. All element-level calculations were based on the particle interactions within this pseudo element. This code architecture, while inline with other Lagrangian element types in Abaqus/Explicit, does not take advantage of the particle-to-particle pairwise-interaction nature of the SPH method and results in suboptimal performance and excessive memory usage, particularly when modeling liquids.

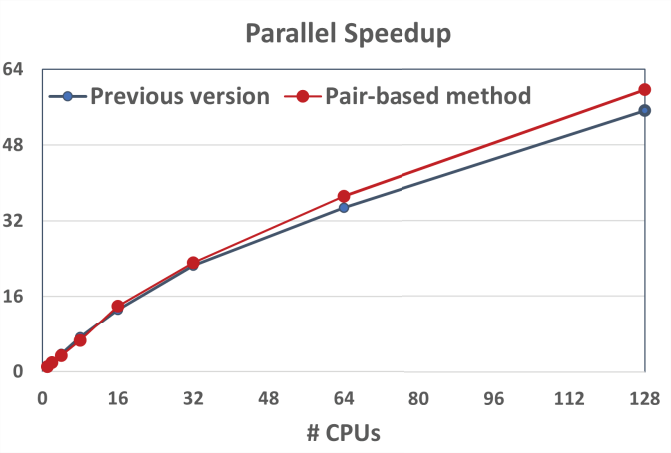
Now, you can use a new SPH element solving technology using a particle-to-particle pairwise-interaction based solution. The neighboring particle search and domain repartition algorithms are rearchitected to further improve performance. This new algorithm is adopted by default for liquid materials modeled with the Mie-Grüneisen equation of state (*EOS, TYPE=USUP) when the pressure is dependent of the internal energy (only the reference speed of sound and the slope of the UsUp curve are defined for the material).

The example below illustrates the performance improvements and memory reduction achieved using the particle pair-based solution technology for a tank sloshing analysis with different number of CPUs. The first figure shows the performance improvement and memory usage reduction compared to the previous pseudo-element-based approach across different numbers of CPUs.



The performance improvement ranges from ~39% to ~49%, while the memory usage reduction ranges from ~6% to ~12%. The performance improvement and memory usage reduction for any given SPH model might vary depending on the presence of other features such as contact, deformable elements, etc.

The second figure shows the parallel speedup improvement across different numbers of CPUs. The parallel speedups are noticeably better for the new particle pair–based SPH method when the number of CPUs is larger.



You cannot use this new feature in combination with the improved SPH boundary behavior (*SPH SURFACE BEHAVIOR) or when the *EOS, TYPE=USUP material definition includes tensile failure (*TENSILE FAILURE).

For more information, see *Smoothed Particle Hydrodynamics*.

Enhanced GPU Support for Modal Dynamic Solvers

Abaqus now delivers faster, more efficient GPU acceleration for mode-based steady-state and transient modal dynamic analyses.

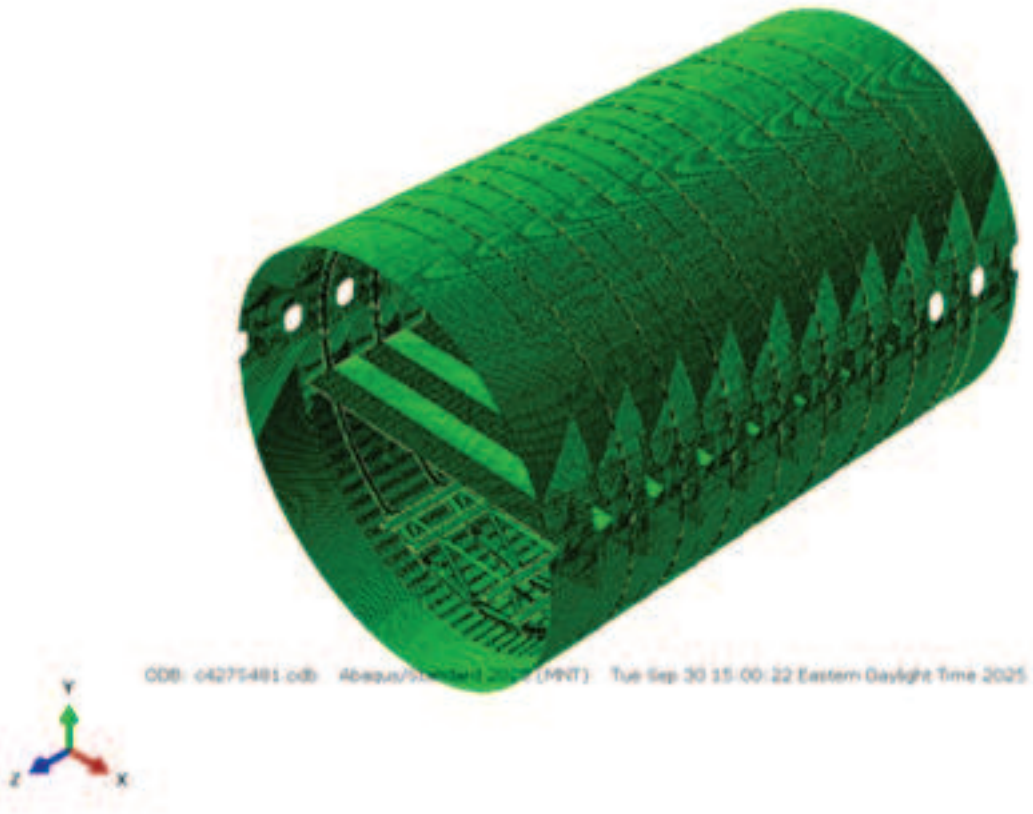
Benefits: These solver enhancements dramatically improve performance and scalability for linear dynamic simulations on GPU-enabled systems, providing faster, more efficient analysis of large-scale models and complex dynamic scenarios across steady-state and transient workflows

GPU support for modal analysis solvers in Abaqus has been significantly enhanced to deliver faster and more scalable solutions. Performance of the multi-GPU Conventional mode-based steady-state dynamics solver has been improved substantially. A newly implemented GPU-based Spectral solver extends GPU support for the mode-based steady-state dynamic analysis. In addition, a new GPU-based State Space solver enables efficient

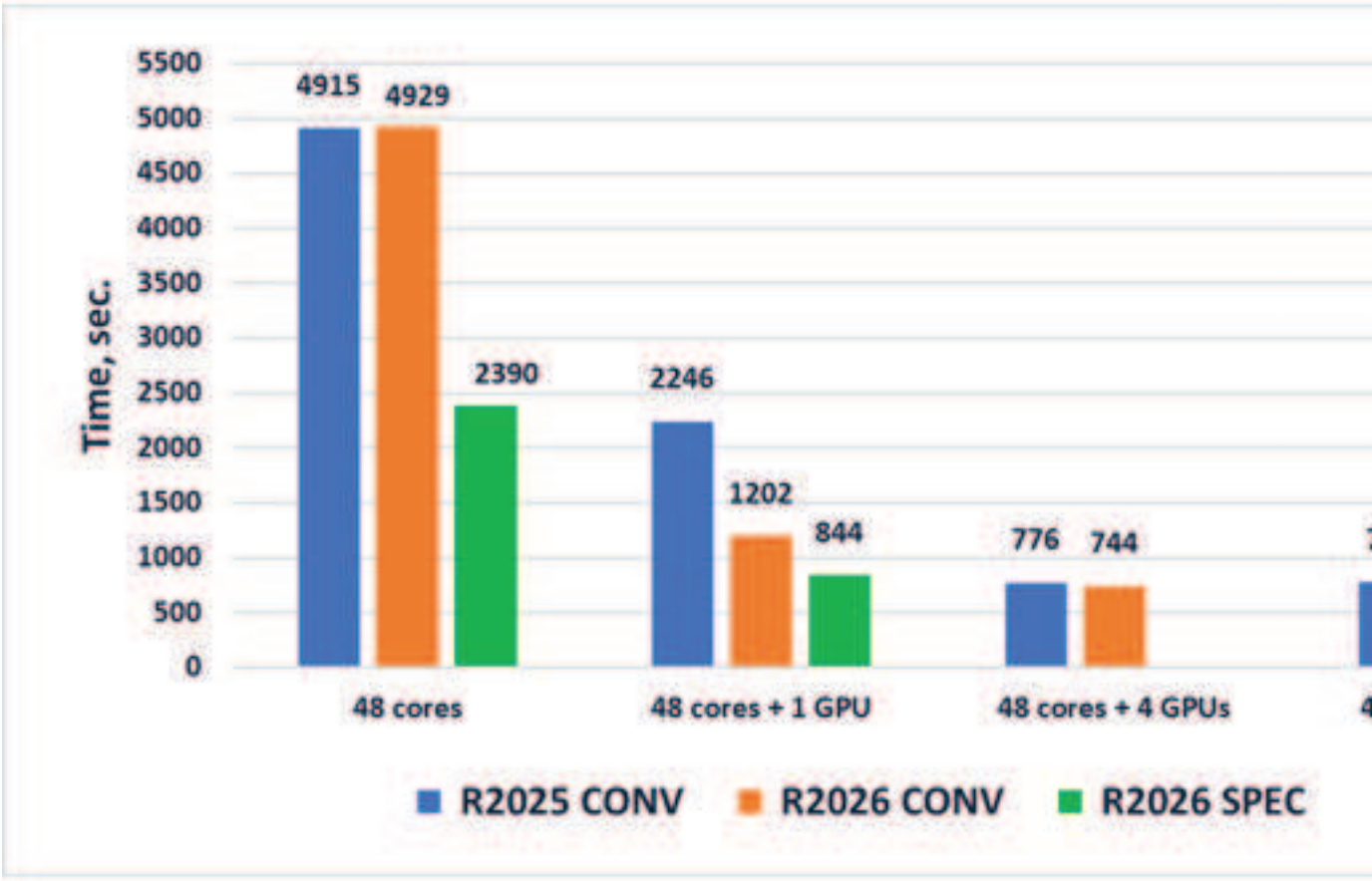
transient modal dynamic simulations. These improvements unlock substantial performance gains, especially for large-scale models and complex dynamic scenarios.

Example 1. Frequency Response Analysis – Fuselage Model

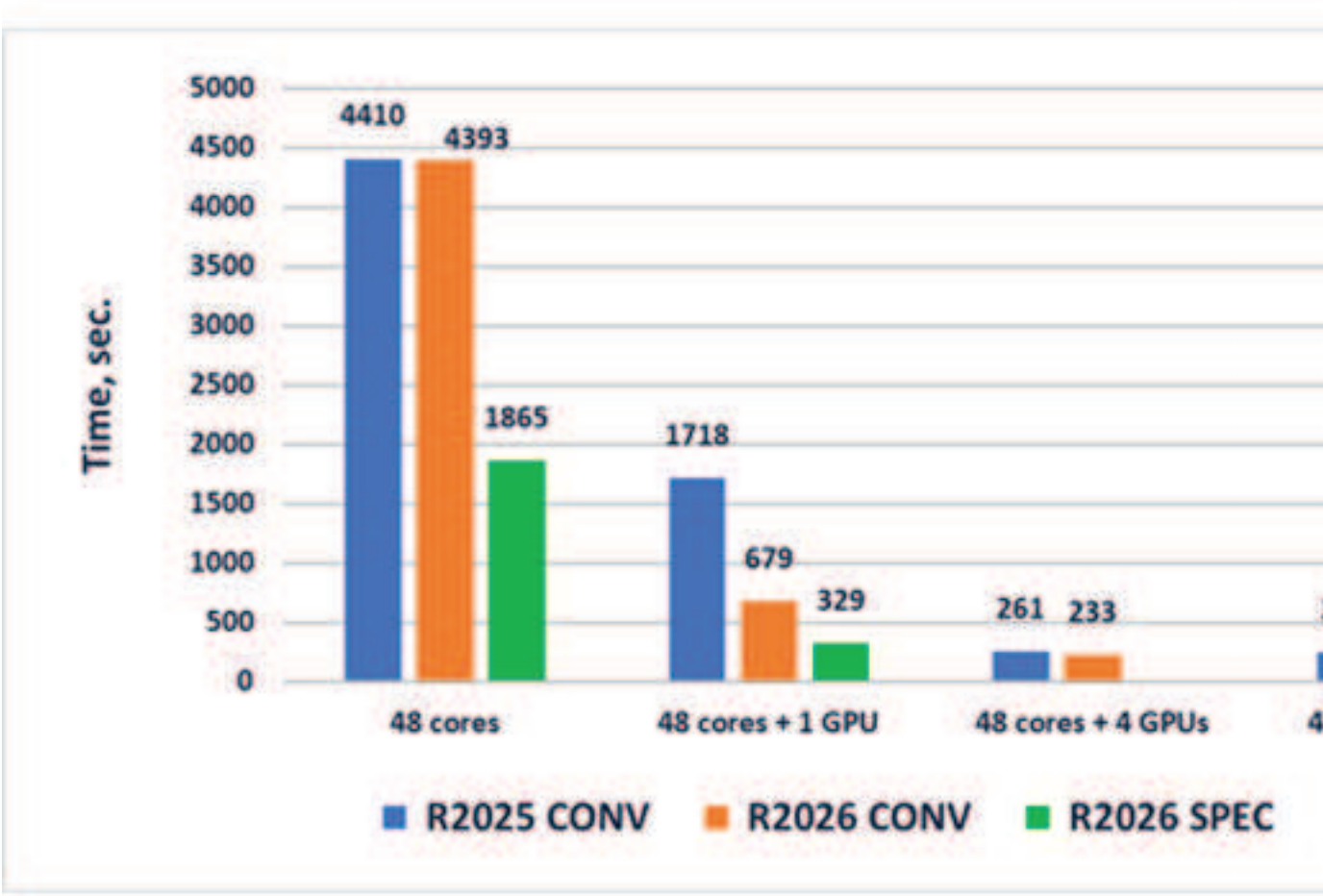
- 48.3 million degrees of freedom
- Structural material damping
- 1000 frequency points
- 120 load cases
- 21,327 eigenmodes



This example highlights the solver's ability to efficiently handle high-resolution modal steady-state dynamic (SSD) simulations. The figure below shows the total analysis time for the fuselage model.

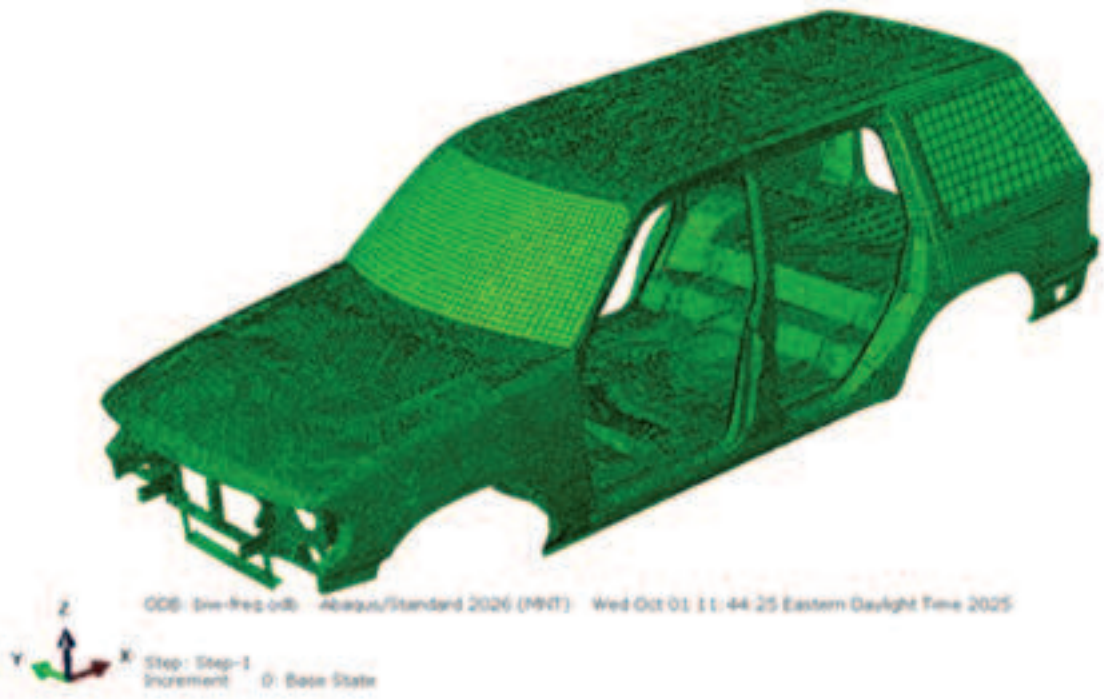


The figure below shows the modal SSD solver time for the fuselage model.



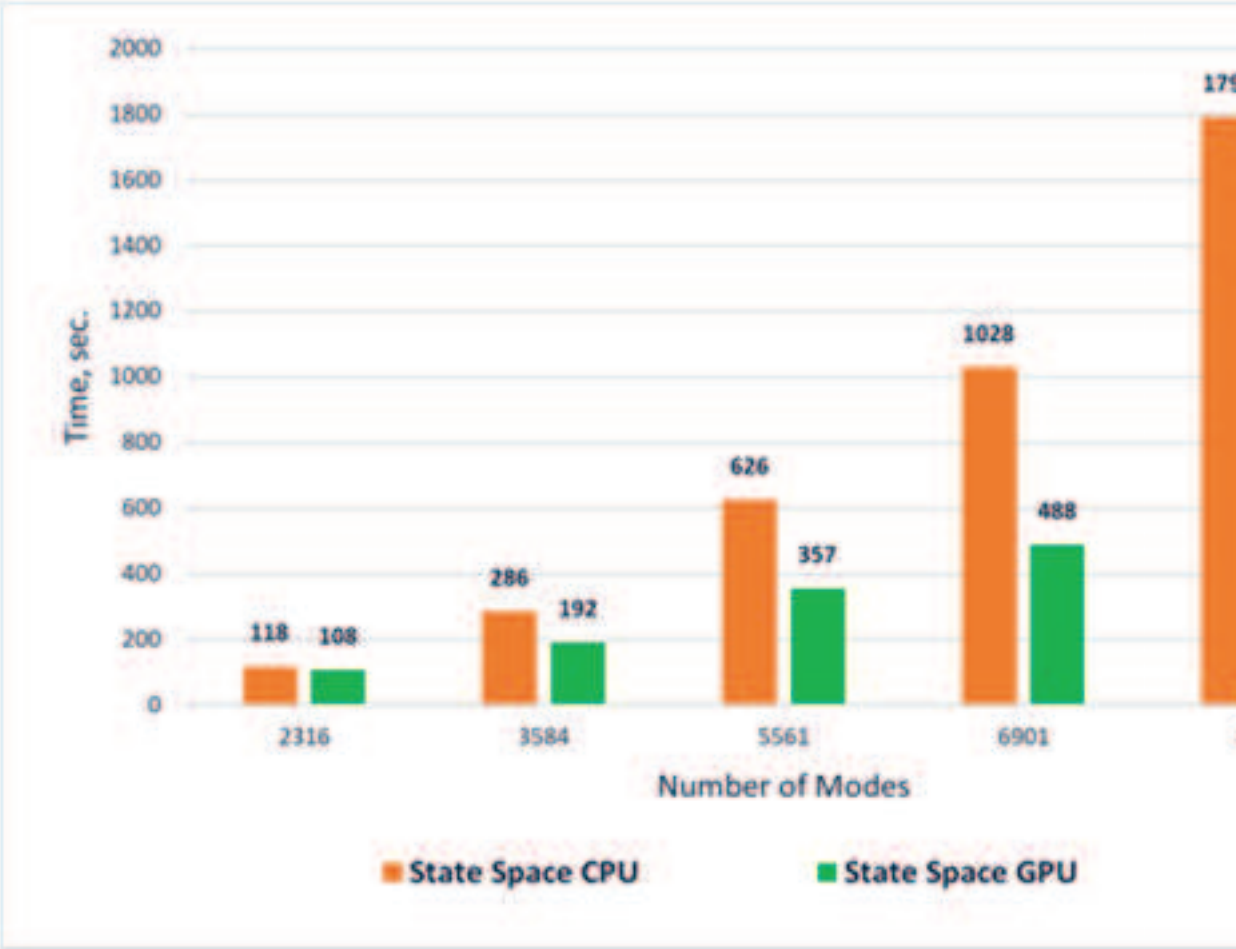
Example 2. Transient Modal Dynamic Analysis – Body-in-White Model

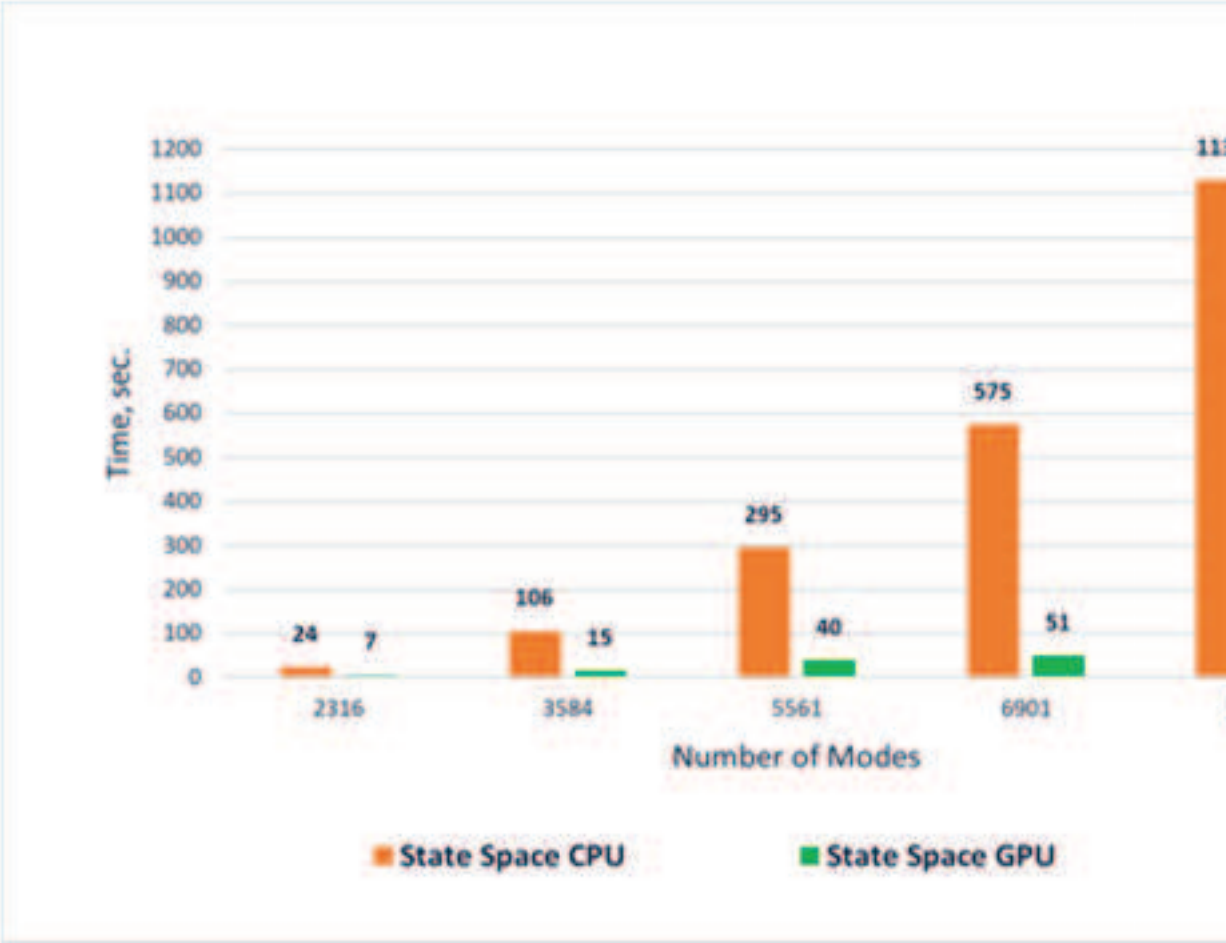
- 780,000 degrees of freedom
- Material viscous damping
- 10,000 time points



This example demonstrates the performance improvements achieved using the GPU-based State Space solver for transient modal dynamic analysis for several simulations with different number of eigenmodes used for the modal approximation of the solution.

The figure below shows the total analysis time for the body-in-white model.


2000
1800
1600
1400
1200
1000
800
600
400
200
0
Time, sec.
118
108
286
192
626
357
1028
488
2316
3584
5561
6901
Number of Modes
State Space CPU
State Space GPU



For more information, see *Parallel Execution in Abaqus/Standard*.

New Parallel Architecture for the SIMULIA Co-Simulation Services

The SIMULIA Co-Simulation Services (CSS) supports a new distributed memory parallel architecture to enhance performance, capacity, and scalability.

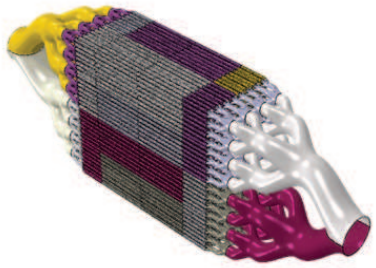
Benefits: The new CSS architecture supports multiple data communication channels and mapping units to maximize the computing resource utilization. You will see greatest performance benefits for large volumetric coupled simulation.

The parallel CSS architecture is a scalable architecture for high-performance computing. The parallel architecture utilizes multiple compute ranks for field transfer and mapping. The new architecture:

- supports large volumetric coupled simulations which the serial CSS architecture was not able to handle before; and
- provides parallel field transfer and mapping.

The figure below shows the structural model of the heat exchanger and the parallel mapping domains. The temperature field was computed by a conjugate heat transfer analysis using the SIMULIA Navier-Stokes solver

(available in the Fluid Scenario Creation, and subsequently imported into a Abaqus/Standard thermal-stress analysis.



The CFD model has close to 40 million cells and the structural model has more than 13 million elements. The table below provides the speedup of the mapper's neighborhood search and mapping during the field transfer for different number of RGrids. An RGrid is a geometric partition of the interface region on which a mapper unit operates on. Significant speedup is observed by engaging most of the client's processors to map. This speedup further reduces the partner client's CPU idle time during a co-simulation while waiting for updated results.

The CSS-API can be embedded in either serial or parallel; both serial and parallel implementations can cooperate. Furthermore, a serial client might delegate mapping to the parallel client, thus maximizing the use of compute resources.

This new capability requires that the clients use the latest CSS libraries. Please refer to *https://www.3ds.com/support/hardware-and-software/simulia-system-information/compatibility/co-simulation/* to find third-party compatibility with Abaqus 2026 FD01 or later.

Table 1: Mapper Speedup for Heat Exchanger Model

CPUs	RGrids	Search Time (s)	Search Speedup	Map Time (s)	Map Speedup
64	1	162.1	1	30.0	1
64	8	73.8	2	3.9	7.7
64	16	23.5	6.9	1.9	15.8
64	32	9.8	16.5	0.94	31.9
64	64	4.3	37.7	0.47	63.8

For more information, see *About Co-Simulation* and *General Capability for Importing External Fields*.

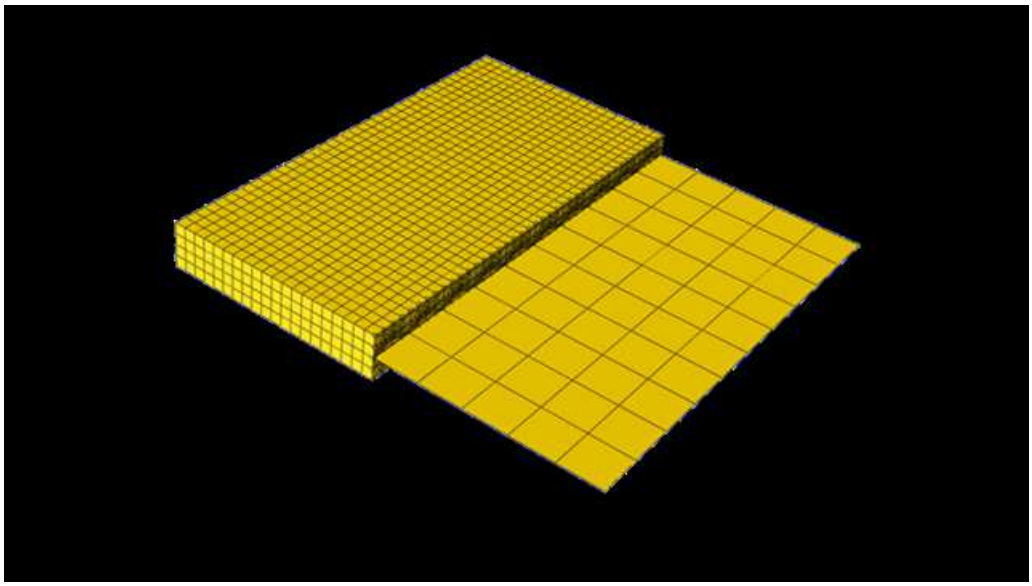
Beam-to-Beam and Solid-to-Solid Submodeling Using the General Submodeling Interface

General submodel now supports beam-to-beam and solid-to-solid submodeling in addition to beam-to-solid, beam-to-shell, shell-to-shell, and shell-to solid submodeling including temperature, pore pressure and acoustic pressure degrees of freedom.

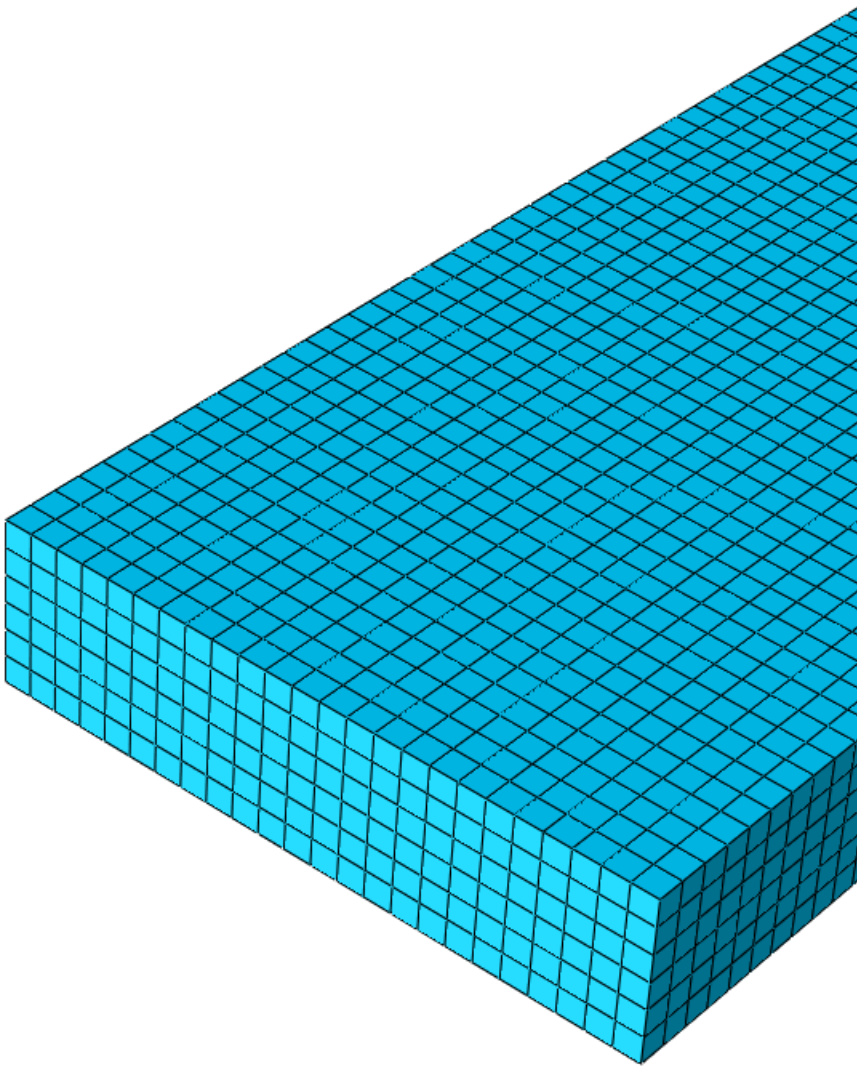
Benefits: You can now perform submodeling with a broader mix of element and analysis types between global and submodel analyses.

The example shown below demonstrates use of solid-to-solid and shell-to-solid submodeling together.

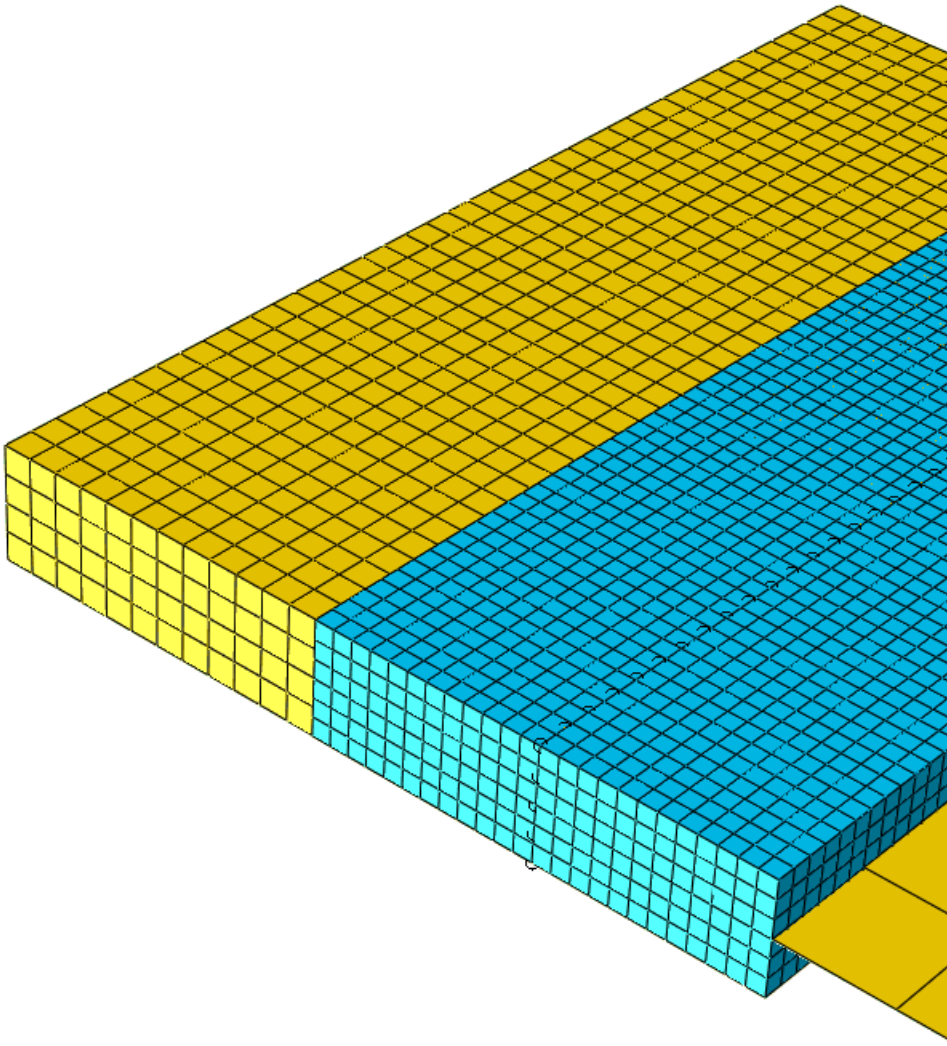
The figure below shows the global model of a plate modeled with solid elements in one region and shell elements in another region with shell-to-solid coupling connecting the two regions.



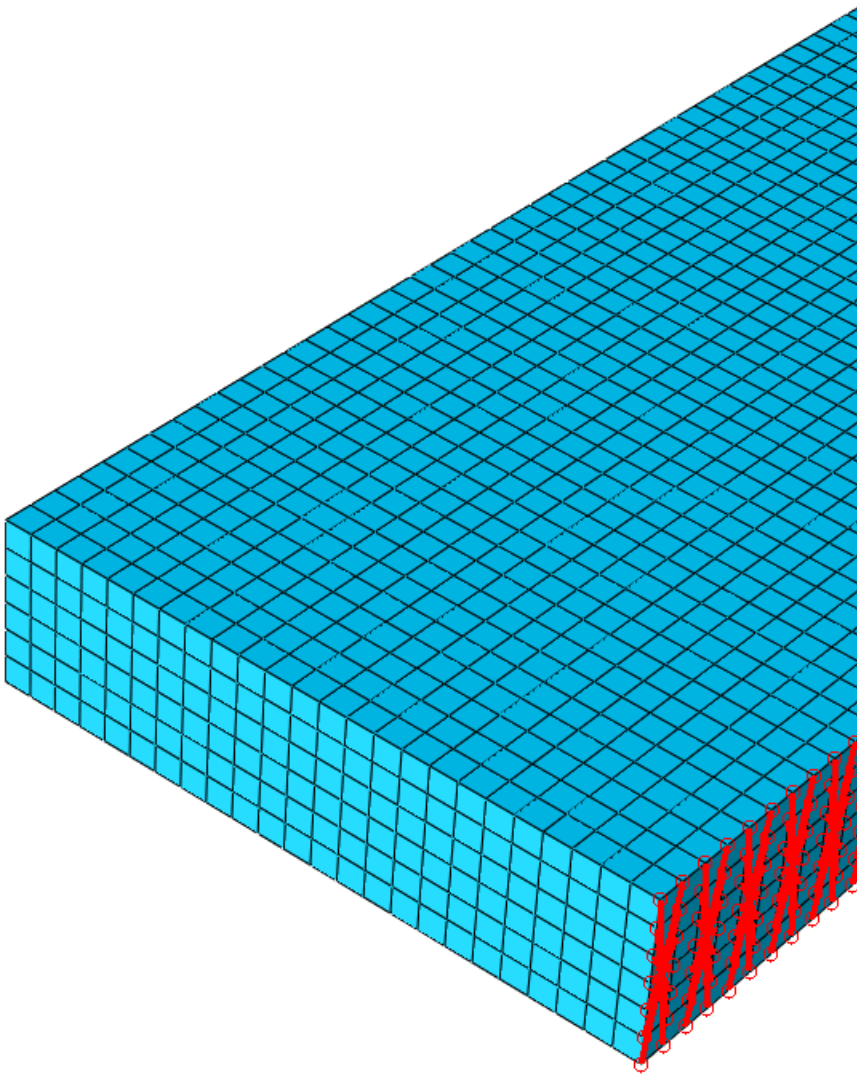
The figure below shows the submodel meshed with solid elements that occupies the middle portion of the global model as shown in the subsequent figure. The interface on one side is shell-to-solid, and the interface on the other side is solid-to-solid.



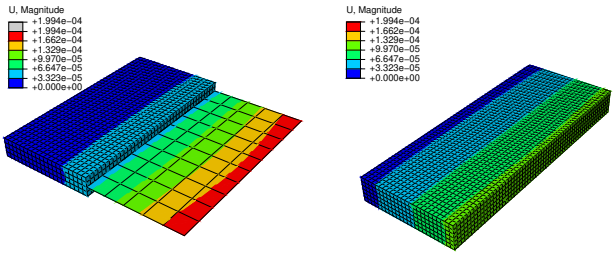
The figure below shows the global and submodel superimposed.



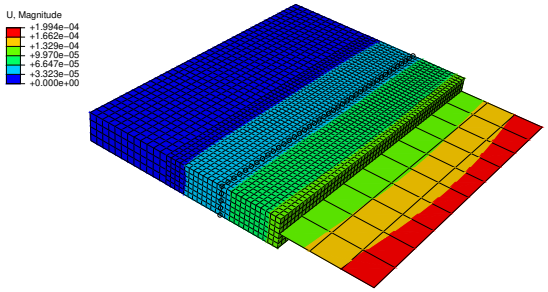
The figure below shows the 30 internally generated distributing couplings used to impose submodel conditions at the submodel cut surface that corresponds to the shell element region of the global model. Displacements and rotations at these internally generated reference nodes are imposed from the interpolated results within the shell elements of the global model. The other submodel cut surface corresponds to solid-to-solid submodeling and displacements at the nodes of that submodel cut surface are imposed based on interpolation of results within solid elements of the global model.



The figure below compares the magnitude of displacement solution for the global model (left side) and the submodel (right side).

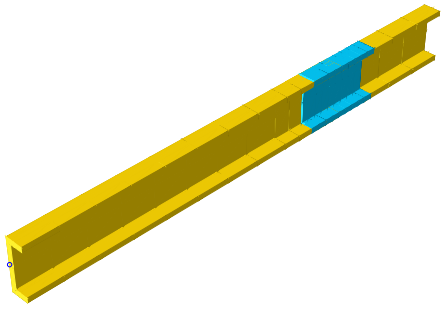


The figure below shows overlay displacement magnitude contour plots for the global and submodel of the example demonstrating shell-to-solid and solid-to-solid submodeling.

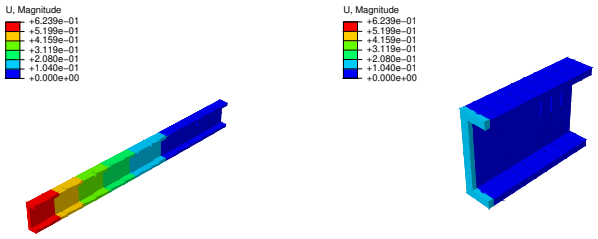


The next example demonstrates beam-to-beam submodeling for a small section near the root of a cantilever beam.

The figure below shows the global and submodel superimposed, with the global model colored gold and the submodel colored blue. Beam elements are used for both the models.



The figure below shows the plot of the displacement solution for the global model and the submodel. Beam cross section rendering is used for both these figures.



The keyword to define the submodel cut surface has been moved from history data to model data. You can no longer select the degrees of freedoms on which to apply the submodel conditions, you must apply submodel conditions to all the primary degrees of freedoms of the nodes on the submodel cut surface.

For more information, see *General Submodeling*.

2026 GA

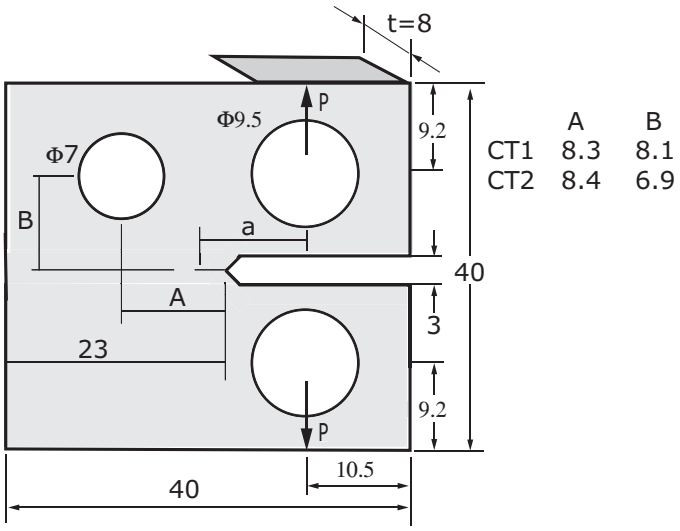
Fatigue Crack Growth with Adaptive Remeshing

You can now use fatigue crack growth with adaptive remeshing in Abaqus/Standard.

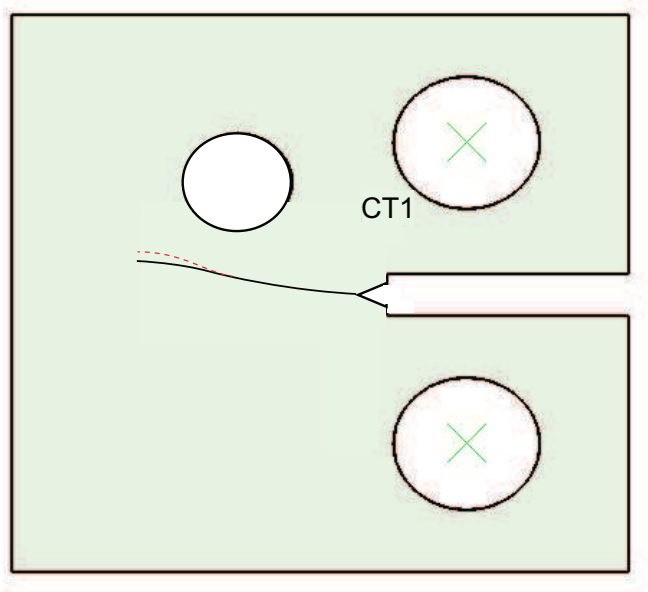
Benefits: This functionality combines the fracture/fatigue technology in Abaqus/Standard with a tetrahedral element-based adaptive remeshing technology to predict crack growth under cyclic loading conditions. The crack growth rate is based on the Paris Law, which is driven by the classical contour integral method for computing energy release rates or stress intensity factors at the crack front.

The overall workflow is semi-automated and involves automatic execution of a series of successive Abaqus analyses with progressively longer crack lengths, without any direct user intervention. The tetrahedral element–based adaptive remeshing technology is used to automatically create a new mesh after each increment of crack growth. The new mesh conforms with the updated crack front. The capability is applicable only for linear elastic fatigue crack growth analysis; therefore, state mapping is not necessary between any two successive increments of crack growth.

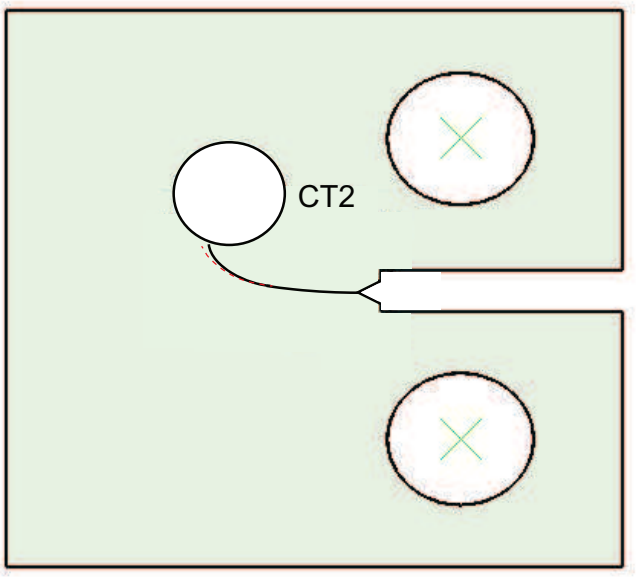
The figure below shows a modified compact tension specimen, following Miranda et al. (2003), with an additional hole that is machined to curve the crack propagation path. The dimensions are in mm. The crack either propagates into the hole or deflects away from it, depending on the hole's location.



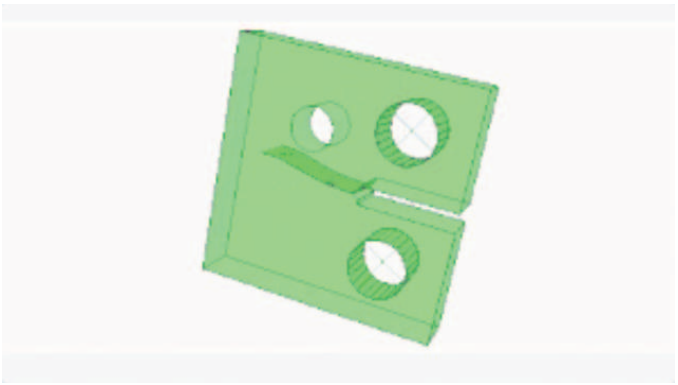
The figure below shows the crack paths predicted by Abaqus overlaid on the experimental measurements for the modified compact tension CT1 specimen.



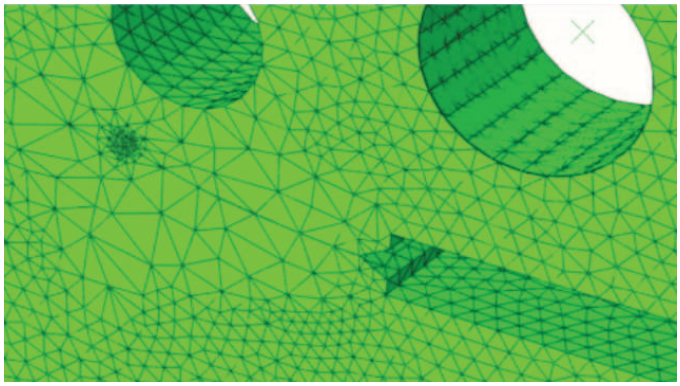
The figure below shows the crack paths predicted by Abaqus overlaid on the experimental measurements for the modified compact tension CT2 specimen.



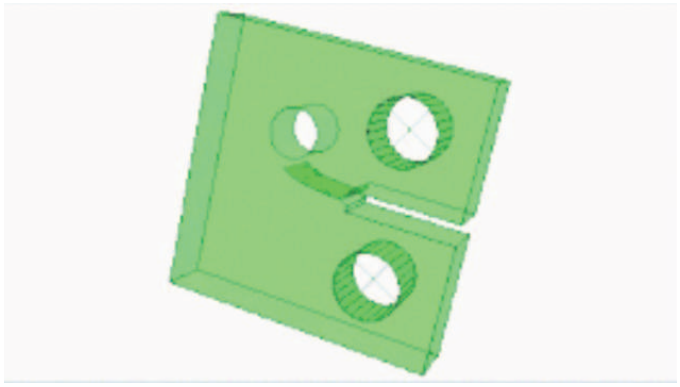
For the CT1 specimen, the animation below shows how the crack initially propagates toward the hole and then deflects away from it as it gets near to the hole.



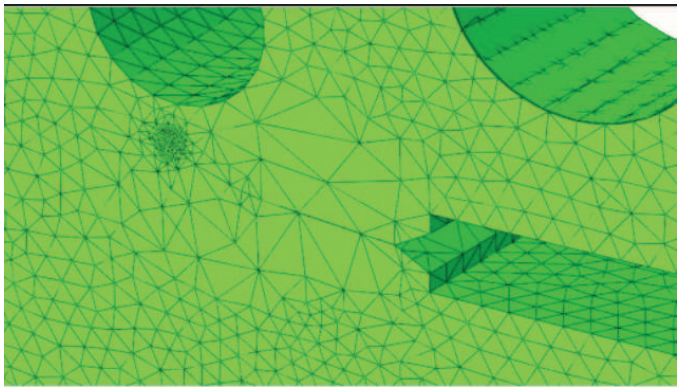
The following animation shows a local view:



For the CT2 specimen, the animation below shows how the crack propagates toward the hole and finally sinks into the hole.



The following animation shows a local view:



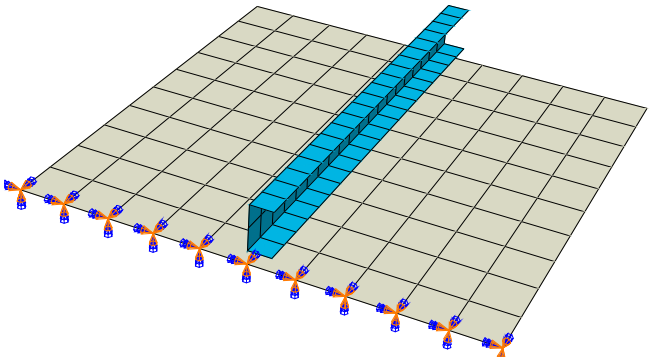
For more information, see *Fatigue Crack Propagation with Adaptive Remeshing*, **ADAPTIVE REMESH*, **ADAPTIVE REMESH RETAIN*, **CONTOUR INTEGRAL*, **FATIGUE*, and **FRACTURE CRITERION*.

Shell-to-Solid and Shell-to-Shell Submodeling Using the General Submodeling Interface

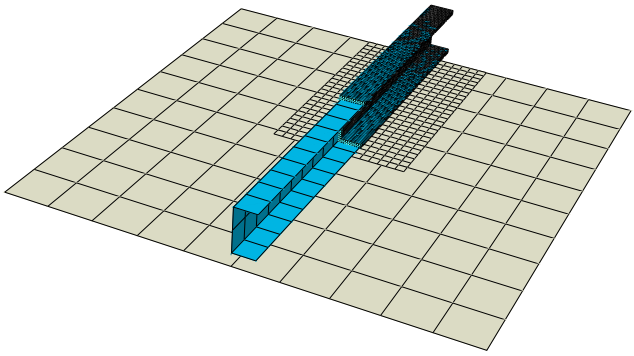
The general submodel interface now supports shell-to-shell and shell-to-solid submodeling in addition to the beam-to-solid and beam-to-shell submodeling cases supported in earlier releases.

Benefits: You can now perform submodeling analyses with a broader mix of element types between the global and submodel analyses.
The following example demonstrates shell-to-shell and shell-to-solid submodeling in a single submodel analysis.

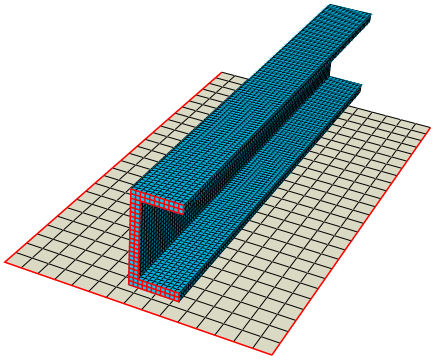
The figure below shows the global model of a stiffener tied to a plate, with the stiffener and plate meshed with shell elements.



The figure below shows a submodel superimposed with the global model.

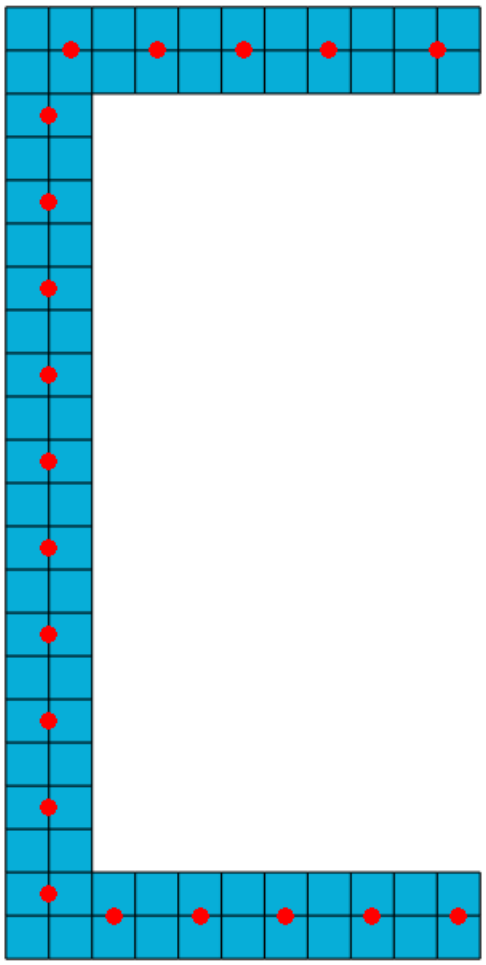


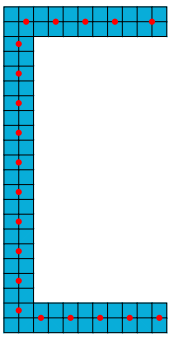
The figure below shows the submodel alone and highlights the submodel cut where results from the global model drive the submodel. The plate has a finer shell element mesh for the submodel than in the global model, and the stiffener is meshed with solid elements in the submodel and shell elements in the global model.



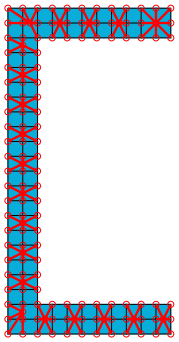
The method used to impose conditions from the global model on a local model at a shell-to-shell submodel cut involves interpolation from the global model results. The method used to enforce conditions from the global model on the local model at a shell-to-solid submodel cut is more complex but easy to use. In this case, Abaqus/Standard internally generates distributing couplings with cloud nodes of solid elements on the cut surface of the submodel. The number of reference nodes and their positions on the submodel interface are determined by an automated algorithm. Displacements and rotations at reference nodes of the distributing couplings are interpolated from results of shell nodes of the global model. Each of the driven solid nodes belongs to a single distributing coupling.

The figure below shows the positioning of the internally generated reference nodes in this example.

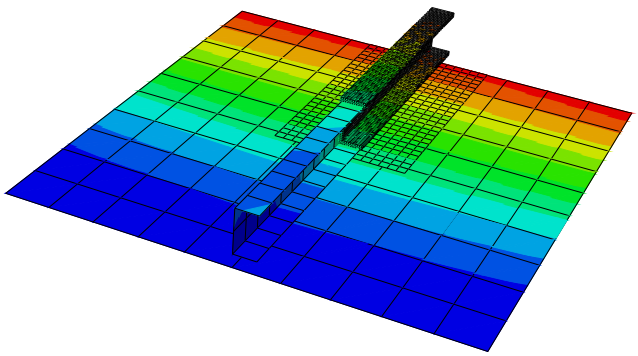




The figure below highlights the connectivity of the internally generated distributing couplings for the shell-to-solid submodel.



The figure below shows the superimposed displacement results for the global model and the submodel with good correlation of results for the global model and the submodel.



For more information, see *General Submodeling*.

2025 FD04

Internal Generation of Parameter Table Types, Property Table Types, and Event Series Types for Additive Manufacturing Simulations

For most additive manufacturing simulations, Abaqus now automatically includes the definitions for the "built-in" special-purpose parameter table types, property table types, and event series types in your model.

Benefits: It is much easier to input the data needed for additive manufacturing simulations.

The data model for the special-purpose additive manufacturing technique is based on using special table collections and event series. In earlier releases, you had to explicitly include the definitions for the "built-in" parameter table types, property table types, and event series types specific to the special-purpose additive manufacturing simulations in your input file. Now, Abaqus generates the information provided by these definitions internally.

The one exception is when you output your simulation in SIM format. In this case you must still provide the definitions explicitly in the input file.

For more information, see *Special-Purpose Techniques for Additive Manufacturing*.

2025 FD03

MPI-Based Parallel Execution of the Subspace Iteration and Lanczos Eigensolvers

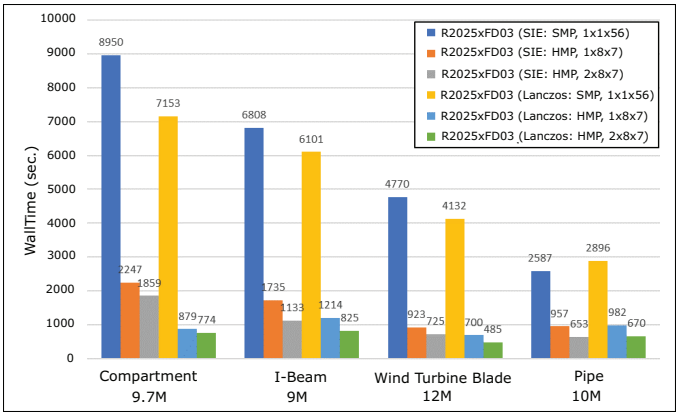
You can now use MPI-based parallel execution of the subspace iteration eigensolver for frequency extraction and buckling analyses and MPI-based parallel execution of the Lanczos eigensolver for buckling analysis.

Benefits: You can solve larger buckling and frequency extraction problems than previously possible and obtain better performance by leveraging multiple MPI ranks across multiple compute nodes.

You can now perform frequency extraction and buckling analysis using the subspace iteration eigensolver in hybrid-memory-parallel (HMP) mode. By utilizing both MPI and thread parallelism within a single compute node, as well as across multiple compute nodes on HPC hardware, the subspace iteration eigensolver significantly improves the performance of both types of analysis.

In addition, buckling analysis using the Lanczos eigensolver can be executed in HMP mode, delivering significantly enhanced performance on HPC hardware with many-core systems. Frequency extraction with the Lanczos eigensolver was introduced in earlier releases.

The figure below illustrates the performance improvement of Abaqus/Standard using the subspace iteration eigensolver and Lanczos eigensolver in 2025 FD03 for four benchmark models extracting 50 buckling eigenmodes. These four models were run in hybrid mode using up to two compute nodes, each of which has an Intel Xeon 2.6 GHz Ice Lake processor with 2 × 28 cores and 512 GB memory.



For the four benchmark models, both eigensolvers shows superior performance in HMP mode (1 × 8 × 7) compared to the shared-memory-parallel (SMP) mode configuration (1 × 1 × 56) and good parallel performance using two compute nodes (2 × 8 × 7). Abaqus/Standard automatically switches to HMP execution of supported procedures for analyses running with at least 32 CPU cores on Linux when no layout is specified.

For more information, see *Parallel Execution in Abaqus/Standard* and *Natural Frequency Extraction*.

GPU AMG Iterative Solver

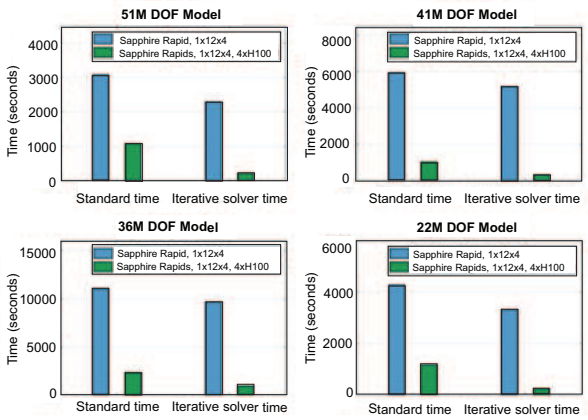
The iterative solver now supports execution across multiple NVIDIA GPUs. It can solve both symmetric and unsymmetric systems of equations without constraints involving Lagrange multipliers.

Benefits: You can now accelerate the iterative sparse solver with multiple GPUs enabling scalability that improves performance and removes model size limitations associated with the single-GPU version.

Unlike the GPU direct solver, which offloads compute-intensive operations to the GPU, the GPU iterative solver runs entirely within GPU memory.

Sufficient GPU memory is critical for successful execution. The memory requirement primarily depends on the number of nonzero entries in the matrix operator, which in turn is largely determined the element type and specific Abaqus/Standard options used (for example, **KINEMATIC COUPLING* and **EQUATION*). In general, when using linear 8-node brick elements, the peak GPU memory usage can be estimated approximately as (× 80 × 12 × *number-of-DOFs* × 1E-9 +) GB, where typically ranges from 2.5 to 4, depending on the AMG smoother used, and accounts for additional temporary workspace needed for handling multiple batch jobs during the factorization-based smoother setup.

Benchmark results comparing the CPU and GPU iterative solver performance across four models are presented below. The CPU solver was executed using a 1 × 12 × 4 configuration (Sapphire Rapids with 48 CPU cores in a single compute node running as 12 MPI ranks of 4 threads each), while the GPU solver added four H100 GPUs to these same CPU cores. The 51M DOF model is the "Medium Engine" version of the s4 engine block benchmark run with the unsymmetric iterative solver. The 41M DOF model is the s9 fuselage benchmark. The 36M DOF and 22M DOF models are static customer powertrain and vehicle frame benchmarks, respectively. Across all cases, the differences in “Iterative solver time” between the CPU and GPU solver are greater than the differences observed in overall “Standard time: because only the iterative solver computations are GPU accelerated. Several of these benchmark models cannot be run on a single GPU due to memory requirements.



For more information, see *Parallel Execution in Abaqus/Standard*.

New Parameter Table Type and Property Table Type References

You can use new parameter table type and property table type references for modeling electrochemical processes in a battery.

Benefits: You can now perform battery multiphysics simulations more easily in Abaqus/Standard.

ABQ_EChemPET_Electrode_Particle_IBVUser	Model the electrochemical process in a battery by specifying a user-defined Butler-Volmer current density at the microscale particle-electrolyte interface using user subroutine `UECHEMECDBV`.
ABQ_EChemPET_Electrode_Particle_IDissUser	Model the electrochemical process in a battery by specifying a user-defined electrode dissolution current density at the microscale particle-electrolyte interface using user subroutine `UECHEMECDDISS`.
ABQ_EChemPET_Electrode_Particle_ILPLUser	Model the electrochemical process in a battery by specifying a user-defined plated lithium current density at the microscale particle-electrolyte interface using user subroutine `UECHEMECDLPL`.
ABQ_EChemPET_Electrode_Particle_ISEIReformUser	Model the electrochemical process in a battery by specifying a user-defined Solid-Electrolyte Interface reformation current density at the microscale particle-electrolyte interface using user subroutine `UECHEMECDSEIREFORM`.
ABQ_EChemPET_Electrode_Particle_ISEIUser	Model the electrochemical process in a battery by specifying a user-defined Solid-Electrolyte Interface current density at the microscale particle-electrolyte interface using user subroutine `UECHEMECDSEIREFORM`.
ABQ_EChemPET_Electrolyte_ChargeDischarge	Model the electrochemical process in a liquid electrolyte battery by specifying a table of values that identify whether a given step in an analysis is a charging or discharging step.

For more information, see *ABQ_EChemPET_Electrode_Particle_IBVUser*, *ABQ_EChemPET_Electrode_Particle_IDissUser*, *ABQ_EChemPET_Electrode_Particle_ILPLUser*, *ABQ_EChemPET_Electrode_Particle_ISEIReformUser*, *ABQ_EChemPET_Electrode_Particle_ISEIUser*, and *ABQ_EChemPET_Electrolyte_ChargeDischarge*.

Importing all Element Sets from the Previous Analysis

You can now easily import all element sets in the previous analysis that have a section definition that is supported for import.

Benefits: You can now import all elements in the analysis model without explicitly listing the element sets to be imported.
The new, default input method imports elements once from all importable sections defined in a previous analysis. It also supports repositioning of the imported elements and importing from multiple previous analyses.

For more information, see *About Transferring Results between Abaqus Analyses* and **IMPORT*.

2025 FD02

Adjoint Fatigue Design Responses and Sensitivities

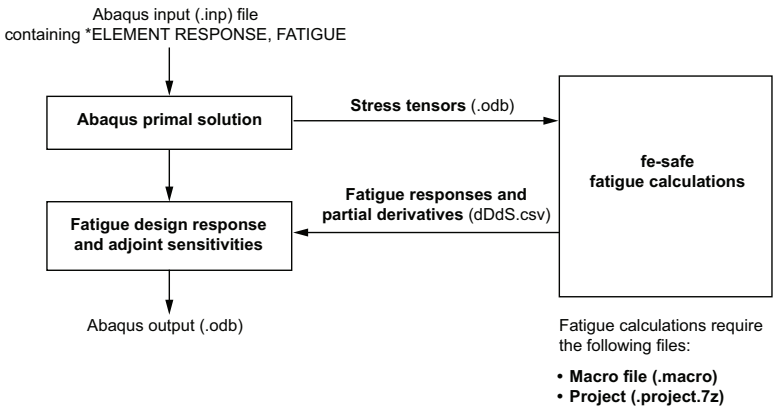
You can obtain adjoint fatigue design responses and corresponding sensitivities utilizing a combined Abaqus and fe-safe workflow.

Benefits: You can carry out Abaqus- and fe-safe-based design optimization accounting for fatigue design responses.
The results from an adjoint fatigue design sensitivity analysis for a model with shell and solid continuum elements are shown below. The analysis includes two static load cases where the shell elements are subjected to pressure loads. The stresses for these load cases are then carried over to drive the fe-safe fatigue calculations.

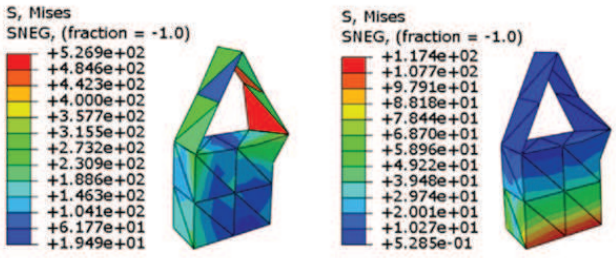
The design variables are mixed and include the following types:

- Shell element thickness (for sizing sensitivities)
- Nodal position (for shape/bead sensitivities)
- Relative material density (that scales the constitutive material properties and the stress relaxation for topology sensitivities)

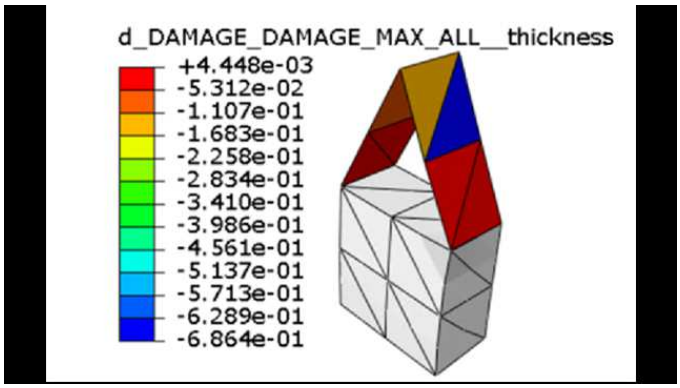
The figure below illustrates the overall combined Abaqus fe-safe workflow for fatigue design responses and corresponding adjoint sensitivities.



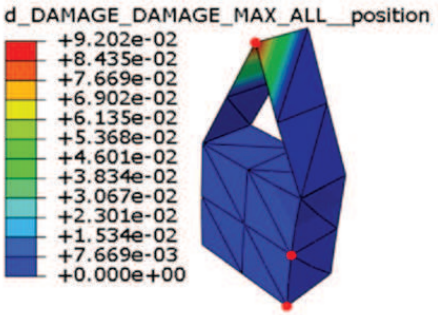
The figure below shows contour plots of the Mises equivalent stress corresponding to the two load cases in the Abaqus static analysis.



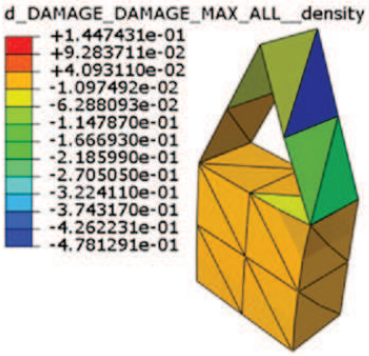
The figure below shows a contour plot of the adjoint design sensitivity of the DAMAGE design response with respect to the shell thickness.



The figure below shows a contour plot of the adjoint design sensitivity of the DAMAGE design response with respect to the nodal positions.



The figure below shows a contour plot of the adjoint design sensitivity of the DAMAGE design response with respect to relative material density.



For more information, see *Adjoint Design Sensitivity Analysis Overview*.

ABQ_EChemPET_Electrode_FluidVF_BrugExp Parameter Table Type Reference

You can use the new parameter table type reference **ABQ_EChemPET_Electrode_FluidVF_BrugExp** to specify the Bruggeman exponents to be used for computing the effective ionic diffusivity and ionic conductivity of the fluid phase in a battery.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see *ABQ_EChemPET_Electrode_FluidVF_BrugExp*.

ABQ_EChemPET_Electrode_SolidVF_BrugExp Parameter Table Type Reference

You can use the new parameter table type reference **ABQ_EChemPET_Electrode_SolidVF_BrugExp** to specify the Bruggeman exponents to be used for computing the effective electrical conductivity of the solid phase in a battery.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see *ABQ_EChemPET_Electrode_SolidVF_BrugExp*.

Eigenvalue Extraction Results Postprocessing

You can use a restart analysis to extract the nodal and element variables from the natural frequency extraction data.

Benefits: You can reduce the output from large natural frequency extractions and better control how results are structured by generating tailored output using postprocessing restart analyses.
For large models that need additional output from the natural frequency extraction step, using a restart analysis can improve the computational performance of the output phase and allows you to tailor output results by generating multiple, smaller files. You can select certain eigenmodes for the output or generate eigenstress, eigenstrain, or similar field output for a specific portion of the model (using an element or node set), which allows you to overcome hardware constraints (such as limited disk space on the disk). Abaqus/Standard extracts the eigenmodes from the original frequency extraction analysis database and calculates the values of the output variables without recomputing the eigenmodes and eigenfrequencies. However, you cannot use this capability to output energy and energy-related variables.

For more information, see *Natural Frequency Extraction*.

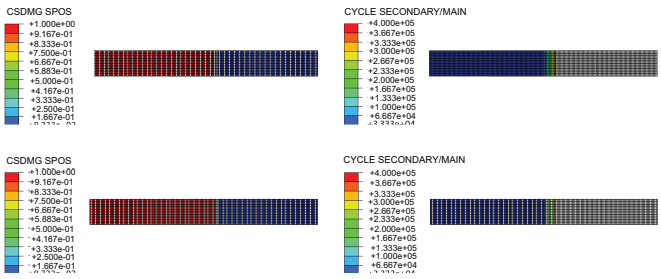
2025 FD01

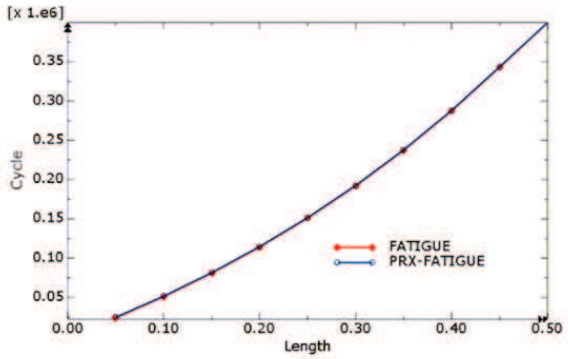
PRX-Fatigue Criterion

You can use the PRX-Fatigue criterion to model delamination propagation under cyclic loading in Abaqus/Standard.

Benefits: The PRX-Fatigue criterion offers smoother delamination fronts and faster solution times due to its better convergence characteristics.

The following figures show results from a linear elastic fatigue crack growth simulation in a single leg bend specimen, where the fracture energy release rates are computed using either the traditional-VCCT technique (top) or the PRX-VCCT technique (bottom). The results obtained by using the two different approaches agree very well with one another.





For more information, see *Crack Propagation Analysis*, *Fatigue Crack Growth Laws for Linear Elastic Materials*, **FRACTURE CRITERION*, and *Debonding behavior of a single leg bending specimen*.

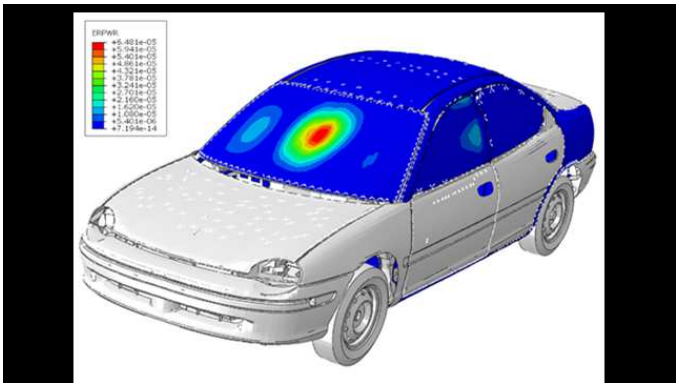
Equivalent Radiated Power Output in Steady-State Dynamic Procedures

You can request output for the equivalent radiated power, equivalent radiated power density, total equivalent radiated power, and the equivalent radiated acoustic pressure in all steady-state dynamic analysis procedures.

Benefits: The new output variables enable you to calculate these quantities when the model contains no acoustic elements or when you need to compute an equivalent radiated power on a structural element-based surface that has no contact with acoustic elements.

Output variable ERPWR represents the equivalent power that is radiated from the tributary area of a node on the element-based structural surface, while ERPWRDEN is an intensive representation of equivalent radiated power density (that is, it is normalized by the node's tributary area), which might yield more meaningful results for cases with nonuniform meshes. You can use output variable ALLERPWR to calculate the equivalent radiated power emitted by a panel (represented by a structural element-based surface). You can use output variable ERPAC to obtain the levels of the equivalent acoustic pressure on the radiating element-based structural surface without employing acoustic elements.

The image below shows a contour plot of the equivalent radiated power (PRPWR) for the Neon model. This model is used for illustrative purposes only and is courtesy of the Public Finite Element Model Archive of the National Crash Analysis Center at George Washington University.



For more information, see *Mode-Based Steady-State Dynamic Analysis* and **EQUIVALENT RADIATED SURFACE PROPERTIES*.

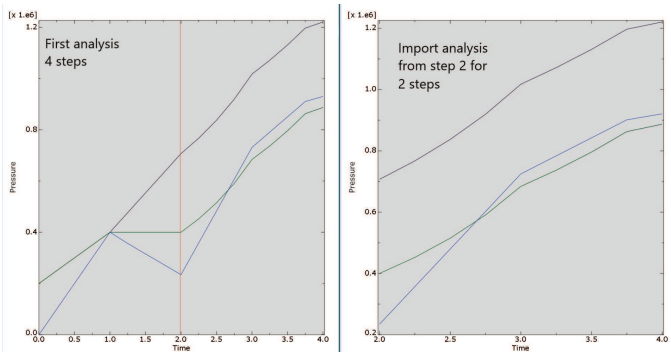
Importing the Fluid Cavity State

You can import the state of a fluid cavity bounded by a surface in any combination of Abaqus/Standard and Abaqus/Explicit import analyses.

Benefits: The transfer of the state of the reference node establishes initial equilibrium of the fluid cavity in the import analysis, and it provides consistent initial conditions for the continuation of the analysis.

The nodal state at the fluid cavity reference node (which represents the pressure inside the fluid cavity) is imported when you redefine the fluid cavity and its reference node, import or redefine the surface of the cavity, or specify import of the material state.

The example below illustrates import of pressure of several fluid cavities in an Abaqus/Explicit to Abaqus/Explicit import analysis. The left image shows the history of the fluid cavity pressures in a single analysis with four steps. The right image shows the results for the same model when you import the fluid cavity at the end of the second step.



For more information, see *About Transferring Results between Abaqus Analyses*.

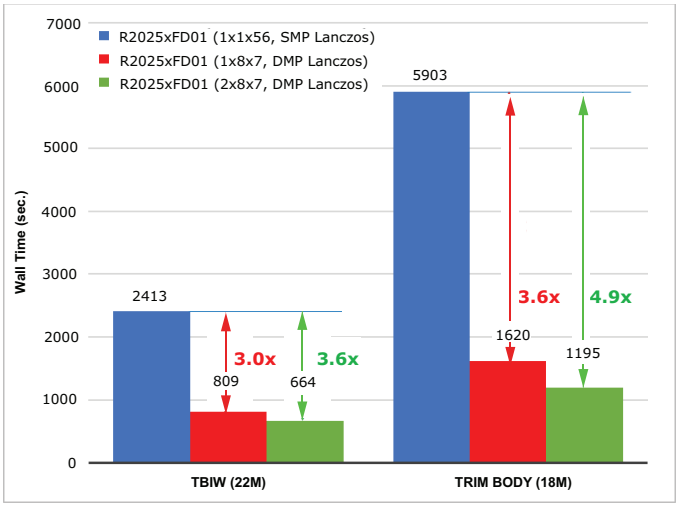
MPI-Based Parallel Execution of Lanczos Eigensolver for Coupled Structural-Acoustic Analysis

You can use MPI-based parallel execution of the Lanczos eigensolver for coupled structural-acoustic analyses.

Benefits: Frequency extraction analysis using the Lanczos eigensolver in a hybrid mode runs significantly faster and can solve large problems by utilizing multiple compute nodes for coupled structural-acoustic problems.

By leveraging both MPI and thread parallelism within a single compute node, as well as across multiple compute nodes on an HPC cluster, the improved Lanczos eigensolver significantly enhances performance and capacity for fully coupled structural-acoustic analysis workflows.

For example, the figure below illustrates the performance improvement of Abaqus/Standard using the Lanczos eigensolver in 2025 FD01 for two benchmark models with acoustic cavities.



The models are run in a hybrid mode using up to two compute nodes, each of which has an Intel Xeon 2.6 GHz Ice Lake processor with 2 × 28 cores and 512 GB memory. Running in hybrid mode yields improved performance as this is a more optimal job layout compared to SMP on these compute nodes. Additional benefits are demonstrated by the ability to leverage processor cores and memory of multiple compute nodes, which was not previously possible.

The first model is a Total Body in White (TBIW) model with 22 million degrees of freedom (DOFs) extracting 113 eigenmodes, and the second model is a trim body model with 18 million DOFs extracting 179 eigenmodes. The reference data is set to the SMP Lanczos run in 2025 FD01.

The job for the first model in a hybrid mode shows 3.0× speedup compared to the job in SMP mode on a single compute node and shows 3.6× speedup on two compute nodes. The job for the second model in a hybrid mode shows a 3.6× speedup compared to the SMP job on a single node and a 4.9× speedup on two compute nodes.

For more information, see *Parallel Execution in Abaqus/Standard* and *Natural Frequency Extraction*.

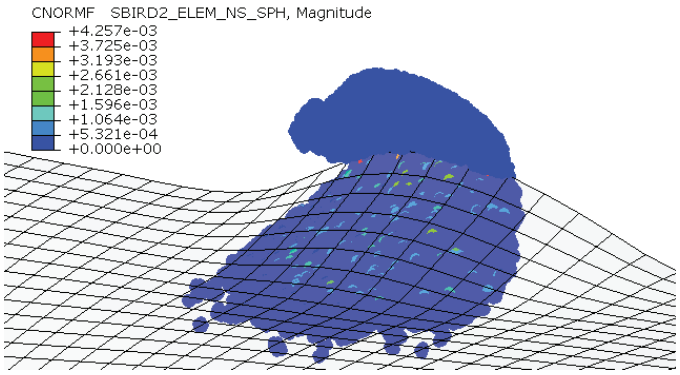
Contact Output for SPH Particles in Parallel Execution

You can request contact output for SPH particles in parallel analyses in Abaqus/Explicit.

Benefits: Supporting contact output for SPH particles in parallel analysis makes it easier to assess the simulation results.

In earlier releases, contact output for SPH particles was available only in serial execution, limiting the full assessment of SPH simulation results in parallel execution runs. You can now request contact output for SPH particles in analyses run with DMP execution.

The figure below shows the contact output CNORMF for SPH particles in a bird impact analysis for a parallel run using 8 CPUs.



For more information, see *Smoothed Particle Hydrodynamics*.

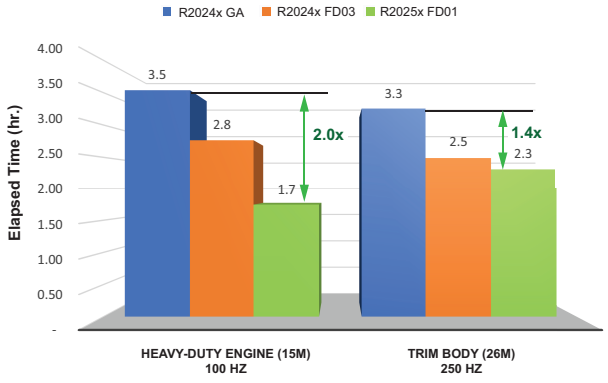
AMS Full Recovery Performance Improvement

Performance for the full recovery of eigenmodes and substructure modes for frequency extraction analysis and substructure generation analysis using the AMS eigensolver is improved for models with many MPC-type constraints.

Benefits: This enhancement reduces analysis run times significantly for common flexible body generation workflows.

The AMS full recovery performance is improved significantly for models with many MPC-type constraints by minimizing the SIM I/O operations. The performance improvement can be frequently demonstrated for a large substructure generation model with many retained nodes and a nondefault flexible body generation request that imposes the AMS full recovery. This improvement will be especially significant for the generation of large flexible bodies for particular multibody dynamics solvers with many retained nodes.

For example, the figure below illustrates the improved performance of Abaqus/Standard using the AMS eigensolver for two benchmark models run on a machine with Intel Xeon Gold 6248 (2.5 GHz Cascade Lake processor with 2 × 20 cores) and 1.5 TB memory.



The first model is a heavy-duty engine model with 15 million degrees of freedom (DOFs) extracting 8 eigenmodes and 5706 substructure modes, and the second model is a trim body model with 26 million DOFs extracting 2547 eigenmodes and 2831 substructure modes. The figure shows how the AMS full recovery workflows have been improved incrementally in 2024 FD03 and 2025 FD01.

The first model took 3.5 hours to run in 2024 GA. This run time was reduced to 2.8 hours in 2024 FD03, and it is further reduced to 1.7 hours in 2025 FD01 due to the improved MPC recovery workflow. As a result, the job in 2025 FD01 runs twice as fast as the one in 2024 GA. Similarly, the job for a 26M DOF trim body model in 2025 FD01 runs 1.4 times faster than in 2024 GA.

For more information, see *Natural Frequency Extraction*.

Additive Manufacturing: Scan Pattern Assignment Using an Element Set

In an Abaqus/Standard additive manufacturing process simulation employing the scan pattern approach, you can now assign a scan pattern directly to the elements contained in an element set.

Benefits: The enhancement facilitates the assignment of scan patterns in situations when different regions in the model cannot be easily separated by a box definition.

You can combine scan pattern assignments using the new element set–based approach with the preexisting box-based assignments. In addition, you can use the new approach for both eigenstrain-based and thermomechanical simulations of additive manufacturing processes.

For more information, see *Thermomechanical Analysis of Powder Bed–Type Additive Manufacturing Processes Using the Pattern-Based Method* and *Eigenstrain-Based Analysis of Additive Manufacturing Processes*.

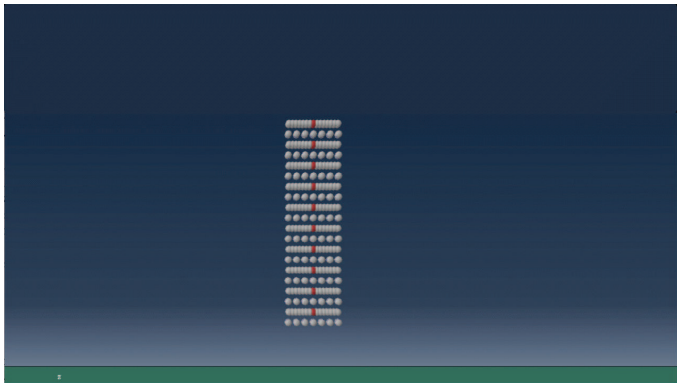
Particle Clustering for Discrete Element Analysis

You can now define arbitrary grain geometry in an Abaqus/Explicit discrete particle analysis by clustering particles together.

Benefits: The new capability allows you to define more realistic nonspherical grain geometry, which improves the fidelity of a discrete element method (DEM) analysis.

Abaqus/Explicit now supports the definition of arbitrary grain geometries for discrete particle analysis by allowing you to define clusters of particles that are rigidly held together to represent the geometry of the physical grains. Each cluster has one parent particle and an arbitrary number of child particles. The particles in a cluster might or might not overlap with each other. The cluster moves as a rigid body with its reference point at the center of the parent particle. (The cluster has three translational and three rotational degrees of freedom at the reference node.) The new DEM particle cluster functionality supports all existing contact interactions.

Only the DMP execution mode is supported for the new DEM particle cluster functionality. The animation below shows a DEM model of a beam stack under gravity loading, with individual beams modeled as clusters of particles. The parent particles have been color-coded in red and blue, while child particles are shown in gray.



For more information, see *Discrete Element Method* and *Particle cluster drop*.

Flexible Body Generation Performance

Flexible body generation performance is improved substantially.

Benefits: You can complete the flexible body generation analysis faster than in earlier releases.

Abaqus/Standard can generate a flexible body from a substructure. It supports several flexible body types for several external flexible body dynamics solvers. Generating the flexible body entities involves massive calculations and significantly affects the analysis time.

The enhancements improve performance for analyses generating flexible bodies for the Simpack multibody dynamics solver, for the AVL EXCITE™ flexible body dynamics solver from AVL LIST GmbH, and for the ADAMS™ flexible body dynamics solver from MSC.Software Corporation.

To demonstrate the performance improvements, two benchmark models are run on Intel Xeon Gold 6348 (2.6 GHz Ice Lake processor with 56 physical cores) with 2 NUMA nodes and 2 sockets.

Model 1. Car body-in-trim structure with 26 million finite element degrees of freedom, 384 substructure retained degrees of freedom (static modes), and 1130 substructure dynamic modes.

Model 2. Engine structure with 12 million finite element degrees of freedom, 2028 substructure retained degrees of freedom (static modes), and 1400 substructure dynamic modes.

The table below illustrates the improved performance of the flexible body generation.

Benchmark model	Flexible body type								
	ADAMS™			AVL EXCITE™			Simpack		
	R2024	R2025	Gain	R2024	R2025	Gain	R2024	R2025	Gain
Model 1									
Flex. Body Gen. time, sec	68950	1830	**37.7×**	7611	785	**9.7×**	14714	1645	**8.9×**
Abaqus/Standard time, sec	72258	5218	**13.8×**	10962	4252	**2.6×**	18053	4995	**3.6×**
Disk peak usage, GB	418	420	**~1×**	762	418	**1.8×**	763	418	**1.8×**

Benchmark model	Flexible body type								
	ADAMS™			AVL EXCITE™			Simpack		
	R2024	R2025	Gain	R2024	R2025	Gain	R2024	R2025	Gain
Model 2									
Flex. Body Gen. time, sec	14886	3931	**3.8×**	21779	1230	**17.7×**	50468	2789	**18.1×**
Abaqus/Standard time, sec	16382	5544	**3.0×**	23298	2774	**8.4×**	51975	4374	**11.9×**
Disk peak usage, GB	372	369	**~1×**	990	370	**2.7×**	990	369	**2.7×**

For more information, see *Generating Substructures*.

PDF Guides Available

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

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

For more information, see *Abaqus Analysis Guide*.

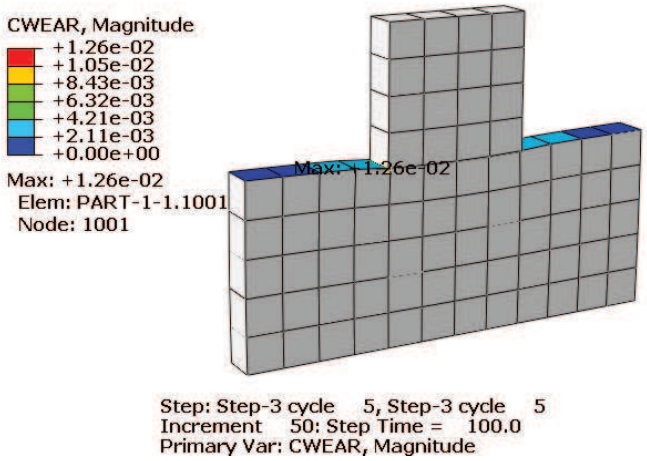
2025 GA

Step Cycling

The step cycling capability provides a simple way to repeat a step or a series of steps over a prescribed number of cycles.

Benefits: You can define cyclic repetition of a sequence of steps using the new step cycling capability instead of explicitly defining steps for every cycle, which provides simpler model setup, better performance, and the ability to end step cycling based on the analysis solution.
There are situations where the analysis requires the repetition of a step, or a series of steps, over a specified number of cycles. Examples of such workflows include aging in batteries, brake wear analysis, and fatigue. In earlier releases, the only approach to repeat a step, or a series of steps, in an analysis was through repeated definition of the same step or steps multiple times. This resulted in reduced performance during all stages of a model (preprocessing, analysis, and output).

The use of step cycling provides an efficient mechanism to loop over the sequence of steps with significant performance improvements. It also provides mechanisms to end a step sequence prematurely based on certain user-specified criteria. A step cycle sequence can be preceded or followed by steps and/or another cycle sequence.



The animation above shows wear distance accumulation on the slider and the bottom block as the slider repeatedly moves back and forth. In this example, the sample wear cycle is modeled using two steps. In “Step-2” the slider moves to the right and returns to the center, while in “Step-3” the slider moves to the left and returns to the center. You can find additional details on this example in *Step Cycling for Contact Wear Simulations*.

For more information, see *Step Cycling*, *Modeling Aging in Batteries*, and *Contact Wear*.

2024 FD03

Alternative Convergence Checks

A new set of alternative convergence controls are now available in Abaqus/Standard.

Benefits: In simulations that involve strong multiphysics coupling among a number of different fields, the alternative convergence controls, which are typically more relaxed compared to the default convergence controls, can result in better convergence characteristics and faster solution times with minimal effects on accuracy.

While the traditional convergence checks in Abaqus/Standard work well for structural simulations, they might be too restrictive for some multiphysics problems (such as battery electrochemistry simulations) that involve strong coupling among a number of different physical fields.

In earlier releases, the only way to modify the default convergence behavior in Abaqus/Standard was to specify nondefault tolerances on the maximum residuals and corrections during an increment. The new alternative convergence checks provide greater flexibility and can be used either globally (for all fields that are active in the analysis) or on an individual field basis. You can also modify the choice of the convergence checks at the step level.

Important: The use of this set of features requires some experience and testing to ensure that the alternative convergence measures produce results with an acceptable level of accuracy.

The table below shows a comparison of solver passes based on different convergence checks.

Model	Number of DOFs	Iteration Count		Gain
		Convergence Check = Strict	Convergence Check = Relaxed	
Model 1	792	1998	1662	1.20
Model 2	2232	673	357	1.88
Model 3	82692	156	128	1.22
Model 4	2380141	1417	958	1.48
Model 5	2764054	1998	1662	1.20
Model 6	4457894	545	395	1.38

For more information, see *Modifying the Default Convergence Checks*.

Adjoint Design Sensitivity Analysis

The adjoint design sensitivity functionality is enhanced significantly for eigenfrequency and direct steady-state dynamic analyses.

Benefits: You can perform the adjoint design sensitivity analysis faster, more capabilities are available, and the usability of the adjoint design sensitivity analysis is improved.
The following enhancements are available:

- Eigenfrequency adjoint design sensitivity analysis now supports the AMS eigensolver. In earlier releases, you could use only the Lanczos eigensolver for this analysis type. Now, you can run adjoint design sensitivity analyses for larger models and take advantage of parallel scaling in the AMS eigensolver.
- Direct steady-state dynamic adjoint design sensitivity analysis now supports topology and shape optimization. In earlier releases, you could use only sizing optimization for this analysis type.
- Direct steady-state dynamic adjoint design sensitivity analysis and eigenfrequency adjoint design sensitivity analysis now allow you to specify sizing and topology design variables for the same element.

For more information, see *Adjoint Design Sensitivity Analysis Overview*.

State Space Solver for Transient Modal Dynamic Analysis

You can use the new state space solver to perform transient modal dynamic analysis for models with nondiagonal modal damping operators.

Benefits: The new solver is unconditionally stable and offers improved performance for models with many time increments and a moderate number of modes. It also supports coupled structural-acoustic analysis.

In earlier releases, the only solver available for transient modal dynamic analysis with nondiagonal modal damping was the conventional solver. The conventional solver uses a time integration scheme based on the assumption that the nondiagonal entries of the modal damping operator vary linearly within each time increment together with the loads. It calculates the approximate modal solution, and the solution accuracy depends on the time step size. In addition, the time integration algorithm is not unconditionally stable, particularly for models with a high level of damping. For common engineering simulations, the accuracy and stability of the conventional solver is usually acceptable. However, obtaining an accurate and stable solution using the conventional solver

requires the use of very small time increments for specific cases. In addition, the conventional solver does not support coupled structural-acoustic analysis.

The new state space transient modal dynamic solver calculates the exact modal solution for systems with arbitrary modal damping for loads that vary piecewise linearly with time. This solver uses the complex eigenmodes of the unsymmetric modal state space operator to represent the real transient modal solution. The solution includes two phases: solving the unsymmetric eigenvalue problem for the state space operator at the "setup" phase and calculating the transient solution at the "apply" phase. The setup phase is the most computationally intensive part of the analysis. It can be costly for systems with many thousands of modes. However, the calculated solution with the state space solver is more accurate than for the conventional solver and the algorithm is unconditionally stable. In addition, the state space solver supports structural-acoustic transient modal analysis using the modal subspace built from the coupled structural-acoustic eigenmodes or from the uncoupled structural and acoustic eigenmodes.

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

Unit System Translation

The SIMULIA Co-Simulation Services now supports unit system translation.

Benefits: You now have the flexibility to use the unit system best suited to your industrial workflow.

To avoid mistakes, it is best to perform coupled multiphysics and multiscale simulation where both models use the same units. Abaqus has no units built into it and assumes self-consistent units. Some third-party solvers might represent their internal data in a unit system different from the one in which the user defined the model.

The SIMULIA Co-Simulation Services typically tabs into the solver and consequently exchanges field data in the solver's unit system. This forces the Abaqus model to be in the unit system used by the other solver's internal format and can lead to unconventional model specification. For example, consider a sequential analysis of a circuit board modeled in CST Studio Suite with results imported into Abaqus. You might employ the CGS system (centimeter-gram-seconds) to define the electromagnetic model. However, CST Studio Suite converts everything to SI (International System) units to perform calculations. This forces the Abaqus model to be in SI units and to use meters as the length dimension for model definition (geometry, material data, load, etc.), which is an unconventional choice for this particular workflow.

Translation between unit systems is available only if the following are true:

- the unit system is defined in the Abaqus model and
- the other solver supports unit system translation through the co-simulation service.

For more information, see *Defining a Unit System for Translating Data with Other Solvers* and **UNIT SYSTEM*.

DMP Direct Sparse Solver Enhancements

The backward pass phase in the distributed memory parallel (DMP) direct sparse solver is improved.

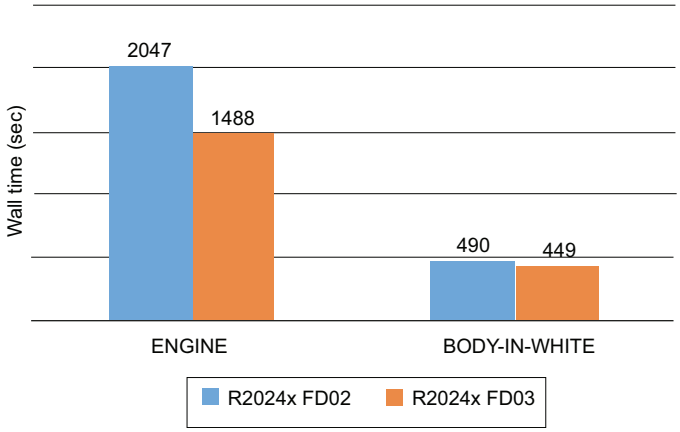
Benefits: The enhancements to the DMP direct sparse solver allow you to complete many analysis procedures faster.

Performance improvements are expected in workflows dominated by the backward pass; for example,

- static substructure generation using the conventional (not AMS-based) algorithm,
- linear perturbation static or steady-state dynamic analysis with multiple load cases, and
- eigenvalue extraction using the Lanczos eigensolver.

The improvements also eliminate cases where the solver could stop responding in certain DMP configurations. While this enhancement improves performance of the backward pass when the factorized matrix resides entirely in system memory, it might degrade performance for cases where the factorized matrix must be partially stored on disk.

To demonstrate performance improvements, two benchmark models are run on Intel Xeon Gold 6348 (2.6 GHz Ice Lake processor with 56 physical cores), with 2 NUMA nodes, and 2 sockets.



- **Model 1:** Static substructure generation for an engine model with 12M DOFs with 2028 retained DOFs run using a host configuration of 1 × 1 × 56 (1 node, 1 mpi rank, 56 threads). This model shows 1.4x speedup in 2024 FD03 compared to 2024 FD02.
- **Model 2:** Direct steady-state dynamic analysis for a body-in-white car model with 3.6M DOFs with 20 frequency points and 120 load cases run using a host configuration of 2 × 8 × 7 (2 nodes, 8 mpi ranks, 7 threads). This model shows 1.1x speedup in 2024 FD03 compared to 2024 FD02.

For more information, see *Direct Linear Equation Solver*.

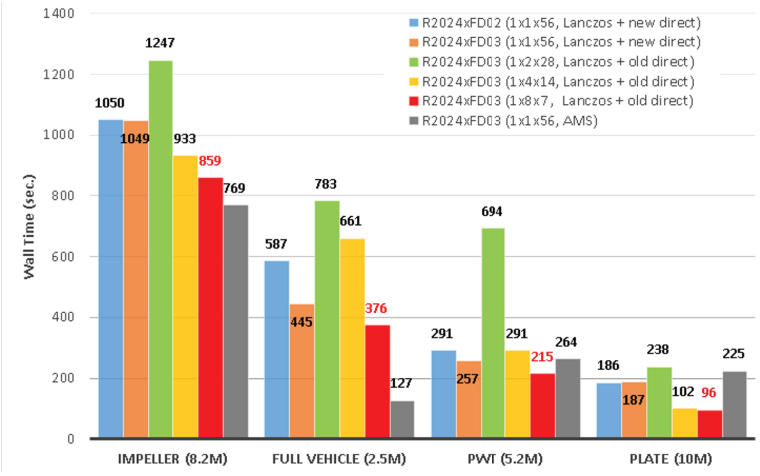
MPI-Based Parallel Execution of the Lanczos Eigensolver

MPI-based parallel execution of the Lanczos eigensolver is now available.

Benefits: Frequency extraction analysis using the Lanczos eigensolver in hybrid mode, which leverages both MPI and thread parallelism, runs significantly faster and can solve much larger problems by utilizing multiple compute nodes.

You can now run frequency extraction analyses using the Lanczos eigensolver in a hybrid mode that utilizes both MPI and thread parallelism. By leveraging both MPI and thread parallelism within a single compute node, as well as across multiple compute nodes on the HPC cluster, the new Lanczos eigensolver significantly enhances the performance of Lanczos eigensolver workflows.

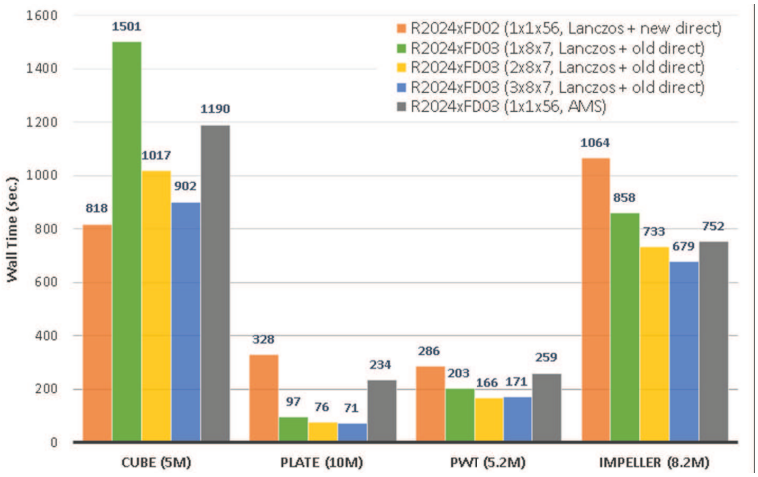
The figure below shows the performance improvement of Abaqus/Standard using the new Lanczos eigensolver in 2024 FD03 for four benchmark models. Each column represents an Abaqus/Standard run time in a different host configuration. Because of the new enhanced thread-based direct solver released in 2024 GA, the Lanczos eigensolver uses the new enhanced thread-based direct sparse solver in shared memory parallel (SMP) mode and the DMP direct sparse solver available in earlier releases in hybrid mode.



The figure illustrates the following:

- The Lanczos eigensolver in SMP mode in 2024 FD03 shows better performance than the previous version in 2024 FD02 for some of models due to the enhancements in the eigenmode transfer phase.
- The Lanczos eigensolver in hybrid mode in 2024 FD03 shows good parallel scaling using multiple MPI ranks on a single compute node. It also shows significantly improved performance compared to the eigensolver in 2024 FD02. For the full vehicle model with 2.5M DOFs, for example, Abaqus/Standard using a host configuration of 1 x 8 x 7 in 2024 FD03 shows 1.6 times speedup compared to 2024 FD02.
- For all four benchmark models, the Lanczos eigensolver in hybrid mode (1 x 8 x 7) runs faster than the Lanczos eigensolver in SMP mode. It also runs faster than the AMS eigensolver with full recovery for some models in 2024 FD03. Therefore, the new Lanczos eigensolver in hybrid mode surpasses the performance limitations of the eigensolvers in SMP mode.

The figure below shows the performance improvement of the new Lanczos eigensolver when utilizing multiple compute nodes on the HPC cluster.



The figure illustrates the following:

- The new Lanczos eigensolver shows good parallel scaling using multiple compute nodes.

- The new Lanczos eigensolver using multiple compute nodes (that is, using more CPU cores and memory) runs significantly faster than the Lanczos eigensolver and the AMS eigensolver in SMP mode for all four models.
- For the impeller model with 8.2M DOFs, for example, Abaqus/Standard using a host configuration of 3 x 8 x 7 in 2024 FD03 runs 1.6 times faster than the previous version in 2024 FD02. It is even faster than the AMS eigensolver in 2024 FD03. This demonstrates that the new Lanczos eigensolver in hybrid mode using multiple compute nodes surpasses the performance limitations of the eigensolvers in SMP mode.

For more information, see *Parallel Execution in Abaqus/Standard*.

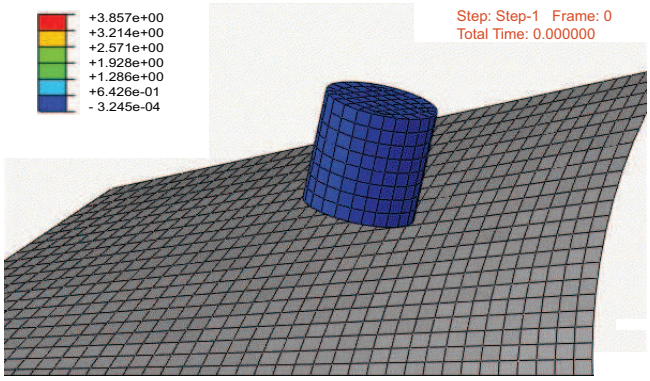
Parallel Execution for SPH Progressive Element Conversion

You can now run SPH progressive element conversion models in parallel.

Benefits: Parallel execution support can greatly improve the simulation performance of models that include SPH progressive element conversion.

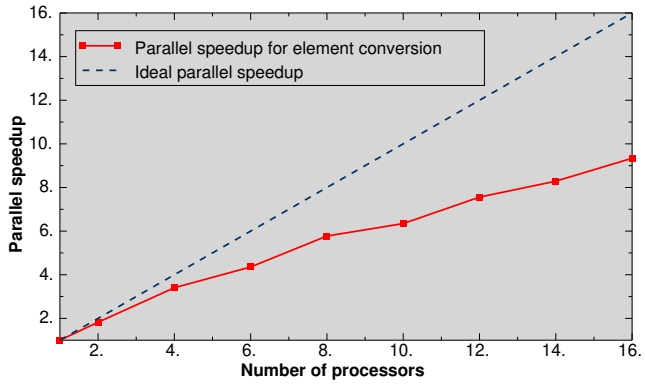
Earlier releases of Abaqus/Explicit supported SPH progressive element conversion only in serial execution mode, limiting the usability of this feature for broad applications. This release supports parallel execution for SPH progressive element conversion, which enables you to take full advantage of the intrinsic strengths of both the Lagrangian finite element and SPH methods.

The figure below shows a parallel run example for a model of a bird striking an airplane engine blade.



The bird structure consists of 480 C3D8R elements, with an EOS material definition that can be converted to around 100k SPH particles based on a strain conversion criterion. Once the strain for a certain C3D8R element meets the user-defined conversion criterion, the element is converted to SPH particles to sustain more severe deformation.

The graphs below show the parallel execution speedup as function of the number of processors, illustrating reasonably good scaling.



The parallel speedup might vary for different models depending on the uniformity of the SPH particle distributions. It is recommended to use more uniform-sized parent elements (preferably C3D8R) for the SPH progressive element conversion to achieve better accuracy and parallel scalability.

Fatigue Crack Growth Analysis Using Field Import-Based Submodeling

You can now import external fields in a linear elastic fatigue crack growth analysis.

Benefits: You can now predict fatigue crack growth in submodels through external field import by mapping fields from the global model to the boundary of the submodel. This allows you to use global models with multiple steps.
In the traditional submodeling approach, the submodel can be driven only by results from a single step in a global model.

For more information, see *General Capability for Importing External Fields*, *Linear Elastic Fatigue Crack Growth Analysis*, and *Node-Based Submodeling*.

2024 FD02

Element Design Responses for Transient Adjoint Design Sensitivity Analysis

Transient sensitivity analysis now supports element design responses based on stress and plastic strain as well as user-defined responses.

Benefits: Element design responses, such as stresses and various measures of plastic strain, allow design optimization for transient dynamic workflows to set constraints on the maximum von Mises stress and/or various measures of the plastic strain.
In earlier releases, Abaqus limited the list of available design responses in a transient sensitivity analysis primarily to nodal responses. The rate of internal work was the only available element-level response. The list of element design responses is now expanded significantly to include different stress and/or plastic strain measures.

In addition, you can use user subroutine *UELEMDRESP* to define a user-defined element design response, such as a failure criterion, in a transient optimization.

For more information, see *Adjoint Design Sensitivity Analysis Overview* and *UELEMDRESP*.

Specifying Design Responses in an Adjoint Design Sensitivity Analysis

You can now specify design responses in an adjoint design sensitivity analysis. In addition, you can prevent sensitivity calculations and output that might not be important for design optimization.

Benefits: You can manage the specification of design responses more effectively and prevent the output database from becoming too large.

The list of available operators for node and element design responses now includes the maximum absolute and the minimum absolute operators. These operators can be helpful in situations such as the following:

- In the context of node responses, the maximum absolute operator is useful when the nodal displacements for a node set that contains many nodes should be constrained to be less than a certain value. Without the maximum absolute operator, you must define one node response for each node in the node set, leading to many node responses (and corresponding adjoint sensitivities) in the output database. When you use the maximum absolute operator, you need to define only a single design response that uses this design objective or constraint.
- The minimum absolute operator is useful when the reaction force for a node set that contains many nodes must be larger than a certain value. Without the minimum absolute operator, you must define one node response for each node in the node set, leading to many node responses (and corresponding adjoint sensitivities) in the output database. When you use the minimum absolute operator, you need to define only a single design response that uses this design objective or constraint.
- You can now specify lower and upper bounds on the values of a design response outside of which Abaqus will not compute the adjoint sensitivities. By default, Abaqus writes the design response and the corresponding adjoint sensitivities to the output database for all design responses that do not have any prescribed bounds for the design response definition. In some situations, the sensitivities are necessary only if the value of a design response is within a certain range. In these cases, specifying the lower and upper bounds on the design response values provides a convenient way to limit the sensitivity calculations and the size of the output database.

For more information, see *Adjoint Design Sensitivity Analysis Overview*.

Structure-to-Structure Co-Simulation Using Abaqus/Standard and Abaqus/Explicit

The keyword interface has changed for supporting structure-to-structure co-simulation using Abaqus/Standard and Abaqus/Explicit.

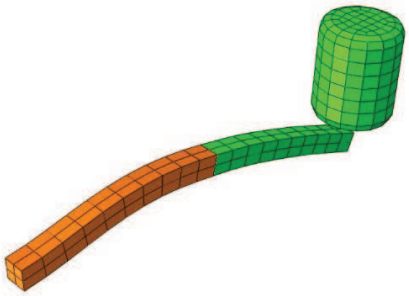
Benefits: The new keyword interface is aligned with other co-simulation products and supports improved interface computations that improve structure-to-structure co-simulations.

Co-simulation between Abaqus/Standard and Abaqus/Explicit exhibits strong physics coupling and requires special treatment at the co-simulation interface. The enhanced subcycling method (see *Enhanced Subcycling Method for Interface Calculations*), which improves solution robustness and is more computationally efficient, was introduced in an earlier release. The enhanced subcycling method is recommended over the original subcycling method (see *Original Subcycling Method for Interface Calculations*) and the lockstep method (see *Lockstep Method for Interface Calculations*).

The original subcycling method and the lockstep method used the **CO-SIMULATION*, PROGRAM=ABAQUS option. Since these are no longer the recommended interface treatments, the PROGRAM=ABAQUS parameter is being deprecated. In addition, the *CO-SIMULATION CONTROLS option, which selects between the original subcycling and lockstep algorithms, is also being deprecated.

You can still select any of the above interface treatment methods by specifying the fields exchanged explicitly and by using the appropriate co-simulation service configuration file, as described in *Structural-to-Structural Co-Simulation*.

The image below shows a cantilever beam impacted by an object. The beam is split into two domains: the orange portion is modeled in Abaqus/Standard, and the green portion is modeled in Abaqus/Explicit.



For more information, see *Structural-to-Structural Co-Simulation*.

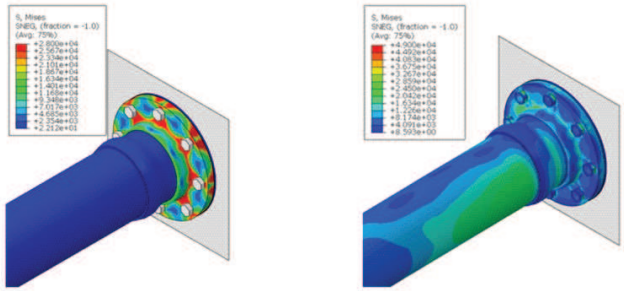
Full-Model Input for Abaqus Import

You can use the full-model input format to specify both new and imported model data in an Abaqus/Standard or Abaqus/Explicit import analysis.

Benefits: The full-model input format allows flexibility in import feature management; for example, partial import of element sets of a section, import of previously rigid element sets as deformable, and import analysis using the SIM results format.

When you use the full-model input format, you give imported model data (such as nodes and elements, node sets and element sets, and sections and materials) in the input. This provides complete model information used in the import analysis.

By using the full-model input format, you can perform certain import workflows (such as importing a previously rigid element set as deformable) simply by omitting the rigid body definition of the element set in the import input. The following example illustrates this approach. The bolts are defined as rigid in the first analysis (left), and they are imported as deformable in the import analysis (right).



For more information, see *About Transferring Results between Abaqus Analyses*.

Undrained Pore Fluid Flow and Stress Analysis

You can now perform an undrained pore fluid flow and stress analysis in Abaqus/Explicit.

Benefits: You can account for the presence of a permeating pore fluid and its effect on the mechanical response of a saturated granular solid skeleton under dynamic loading conditions.

A porous medium typically consists of a solid skeleton of numerous solid grains that are “loosely packed” and a permeating pore fluid that occupies a system of interconnected pores among these solid grains. You can ignore the effect of fluid flow in situations when the loading rate is significantly faster than the rate at which pore fluid can drain from the porous medium. This corresponds to the undrained conditions.

The undrained pore fluid flow and stress analysis functionality in Abaqus/Explicit is intended for the simulation of fully saturated porous media under undrained conditions. The functionality enables modeling the increase in stiffness of the soil in applications such as trenching, gouging, and pile driving. In these applications, you can ignore fluid diffusion terms when the rate of loading is fast compared to the characteristic time of the diffusion process.

For more information, see *Undrained pore fluid flow and stress analysis*.

2024 FD01

Iterative Solver Performance Improvements

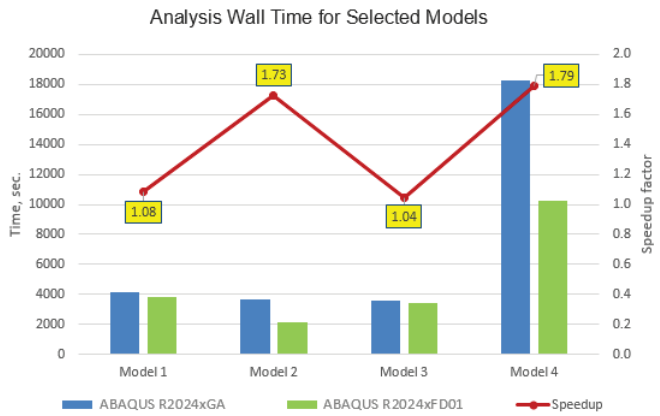
A new domain decomposition algorithm improves iterative solver convergence rates and shortens the overall solution time.

Benefits: The iterative solver is now able to converge on models on which it could not converge in earlier releases (for example, automotive battery structural simulations) or on models for which convergence was prohibitively slow (for example, electric engine analyses).

The new domain decomposition algorithm takes into account the connectivity of the finite element model, the element formulation details, and the material properties. It creates a better clustering of elements with similar properties and separates out clusters with dissimilar properties. This makes it easier for the iterative solver to find a smooth solution in individual aggregates, which results in improved convergence rates overall. The algorithm optimizes the load balance between domains for large models in parallel runs by accounting for the actual computational costs of different elements. The solution time reduction due to the improved domain decomposition varies from model to model. The improvement can be moderate for some finite element models, but it can be significant for large-scale models with complicated geometry using higher-order element formulations.

The figure below illustrates the improved performance of the iterative solver for four selected finite element models:

- Model 1: airplane fuselage structure; shell-dominated finite element model with 48.3 million degrees of freedom.
- Model 2: powertrain model with 23.6 million degrees of freedom meshed with linear continuum elements.
- Model 3: powertrain model with 11.2 million degrees of freedom meshed with quadratic continuum elements.
- Model 4: battery model with 46.0 million degrees of freedom, meshed with linear continuum, shell, and connector elements.



For more information, see *Iterative Linear Equation Solver*.

Generating Reduced Bases for Use with Nonlinear Flexible Multi-Body Systems

A new analysis technique in Abaqus/Standard reads operator data from upstream analyses and generates reduced bases for use in downstream nonlinear flexible multi-body system (MBS) simulations with Simpack.

Benefits: This streamlines the process for preparing MBS simulations that capture large deformation effects with high accuracy and fast run times.

A reduced basis is a set of basis vectors that can represent solutions from preceding analysis cases when they are superimposed with according scaling factors. You can now generate reduced bases for use in downstream nonlinear flexible MBS simulations with Simpack as follows:

1. Define a nonlinear static analysis case that ramps up a large deformation of a flexible component in a shape that is characteristic for its behavior in downstream system simulations. For example, apply a distributed load that acts on the surface of a wind turbine rotor blade in the flap-wise direction, or define a spring support for a twist beam axle and apply a twisting load at the wheel attachment points.
2. Output the operator data at various deformation levels.
3. Add natural frequency extraction steps at various deformation levels.
4. Add the reduced basis generation step.

The reduced basis generation step computes the reduced basis from the solution data and computes the auxiliary operator data from all the preceding analysis steps. The reduced basis and projected operator data are written to a SIM (`.sim`) file. Optionally, you can write a Simpack Flexible Body Interface (`.fbi`) file with a complete Simpack nonlinear flexible body definition that you can assign directly to a nonlinear flexible body in Simpack.

For more information, see *Reduced Basis Generation* and **REDUCED BASIS GENERATE*.

Random Response Analysis Performance Improvements

Random response analyses now have improved performance for the element results computation and output.

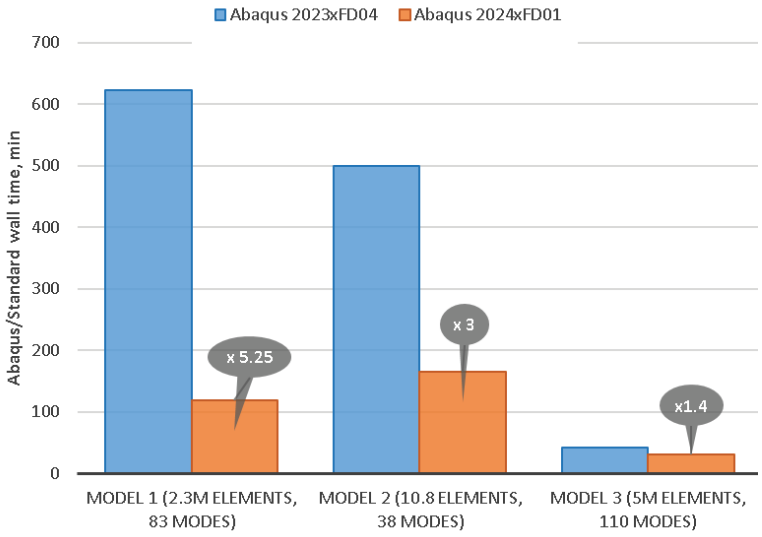
Benefits: The improvements reduce the Abaqus/Standard analysis wall time for random response analyses, particularly for large models with extensive element field output requests.

The enhancement substantially improves performance of the following phases of the random response analysis:

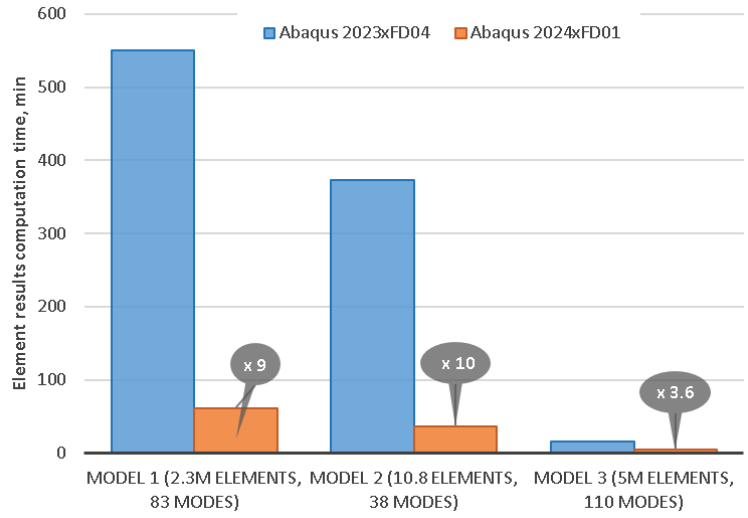
- Reading the eigenfrequency extraction stresses.
- Computing the element results. New scalable algorithms perform the element output computations.

In addition, you can now use the frequency split technique to run a random response analysis on models that do not require calculations of the root-mean-square variables.

The figure below illustrates the wall time improvements for running a random response analysis in Abaqus/Standard for three industrial models.



The figure below illustrates the reduction in the wall time for the element results computation phase for the same three models.



The models and results are as follows:

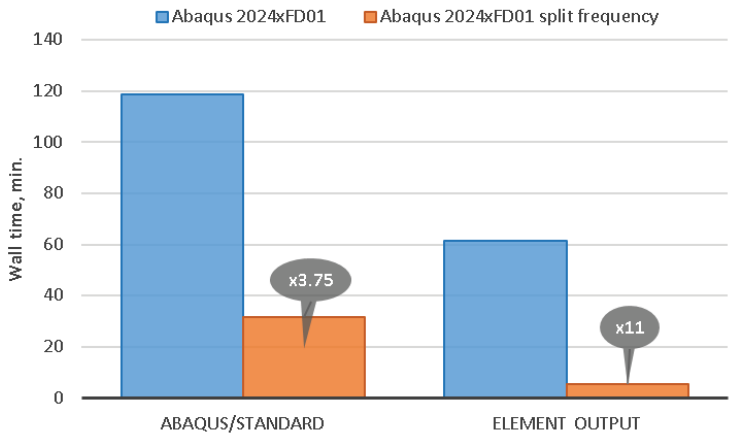
- Model 1 is comprised of 2.3 million elements containing approximately 9 million degrees of freedom. The random response analysis was performed using 83 eigenmodes and 840 frequency points. Stress field output was requested in all increments.

  The Abaqus/Standard wall time was reduced approximately 5.25 times, while the element results computation phase was sped up approximately 9 times.
- Model 2 is comprised of 10.8 million elements containing approximately 53 million degrees of freedom. The random response analysis was performed using 38 eigenmodes and 20 frequency points. Stress and strain field output was requested in all increments.

  The Abaqus/Standard wall time was reduced approximately 3 times, while the element results computation phase was sped up approximately 10 times.
- Model 3 is comprised of 5 million elements containing approximately 25 million degrees of freedom. The random response analysis was performed using 110 eigenmodes and 38 frequency points. Stress and Mises stress field output was requested in 0.6 million elements at all frequency points.

  The Abaqus/Standard wall time was reduced approximately 1.4 times, while the element results computation phase was sped up approximately 3.6 times.

The figure below shows the result of applying the frequency split technique to Model 1. The 840 frequency points used for the random response analysis were split into 16 batches. The total Abaqus/Standard wall time (that includes the wall time of a frequency extraction step and the largest wall time of the random response analyses) was reduced approximately 3.75 times, while the element results computation phase was reduced approximately 11 times.



For more information, see *Random Response Analysis*.

AMS Eigensolver Performance Improvement

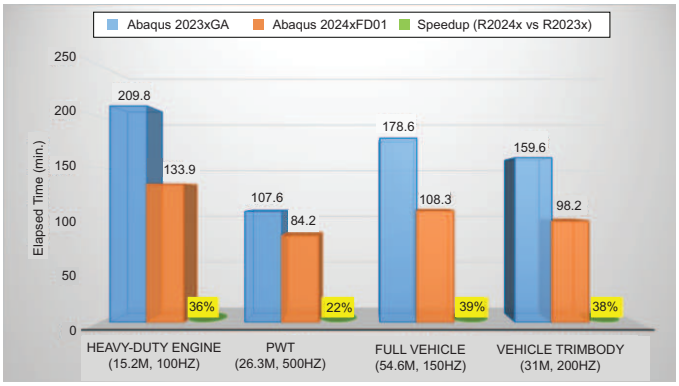
The performance of the frequency extraction analysis using the AMS eigensolver has been improved for models that require full eigenmodes and substructure modes.

Benefits: The AMS full recovery performance improvement significantly reduces the analysis run times for an AMS frequency extraction analysis and for an AMS-based substructure generation analysis requesting full eigenmodes.

The performance of the following phases related to full eigenmodes recovery is significantly improved for the AMS eigensolver workflow including the AMS-based substructure generation procedure:

- AMS recovery phase
- Constraint transformation and recovery phases
- Substructure recovery matrix output phase

The figure below illustrates the performance improvement of Abaqus/Standard in Abaqus2024 FD01 for four industrial models. In this figure, the blue columns represent the Abaqus2023 GA run time, and the orange columns represent the Abaqus2024 FD01 run time.



The models and results are as follows:

- Model 1 is a heavy-duty engine model with 15.2 million degrees of freedom, extracting 10 eigenmodes below 100Hz and 5,704 substructure modes for 5,704 retained degrees of freedom.

  Abaqus2024 FD01 shows a 36% performance improvement over Abaqus2023 GA.
- Model 2 is a powertrain model with 26.3 million degrees of freedom, extracting 6 eigenmodes below 500Hz and 1,512 substructure modes.

  Abaqus2024 FD01 shows a 22% performance improvement over Abaqus2023 GA.
- Model 3 is a full vehicle model with 54.6 million degrees of freedom, extracting 1,630 eigenmodes below 150Hz and 1,984 substructure modes.

  Abaqus2024 FD01 shows a 39% performance improvement over Abaqus2023 GA.
- Model 4 is a vehicle trim body model with 31 million degrees of freedom, extracting 2,598 eigenmodes below 200Hz.

  Abaqus2024 FD01 shows a 38% performance improvement over Abaqus2023 GA.

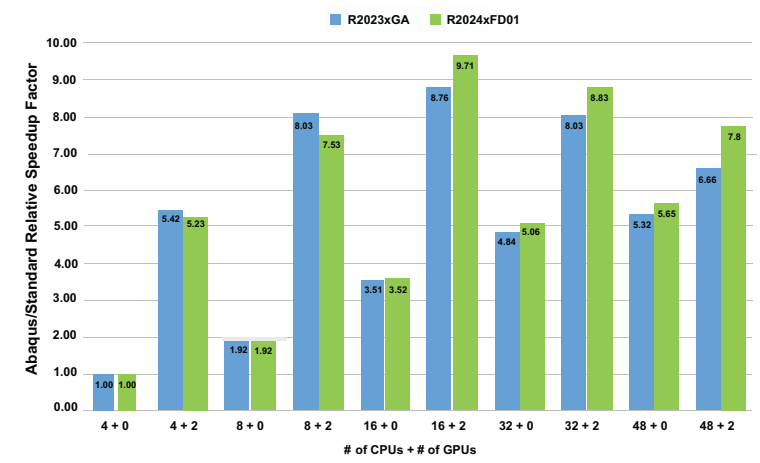
For more information, see *Natural Frequency Extraction*.

AMS Eigensolver GPU Acceleration

Robustness and performance of the AMS eigensolver GPU acceleration is improved for frequency extraction analyses with new NVIDIA GPU cards. New GPU acceleration functionality speeds up modal damping projection in the AMS recovery phase.

Benefits: Analyses using the AMS eigensolver run faster using multiple GPUs.

The figure below shows the GPU acceleration performance of Abaqus/Standard for a 7.6 million degree of freedom stator model that requires 2.3×10^{15} floating-point operations to factorize four system operators (including viscous and structural damping matrices).



The blue columns represent the Abaqus2023 GA run time, and the green columns represent the Abaqus2024 FD01 run time. The performance data was obtained on a machine equipped with Intel Xeon Gold 6336Y (Ice Lake, 2.4GHz, 48 cores) processors, 512 GB physical memory, and 2x NVIDIA A100 (40 GB memory) cards.

Abaqus/Standard using the AMS eigensolver shows good parallel scaling up to 48 CPUs. However, using 16 CPUs and 2 GPUs results in the best performance of the GPU acceleration for this model.

For more information, see *Parallel Execution in Abaqus/Standard*.

Scalable Spectral Solver for Mode-Based Steady-State Dynamics Analysis

A new scalable spectral solver is available for common cases of mode-based steady-state dynamics analysis.

Benefits: Using the new solver can significantly reduce the analysis time for damped finite element models with general structural damping and specific viscous damping properties.

You can use the new spectral solver for the special cases of mode-based steady-state dynamic simulations. For all cases, you can apply any kind of structural damping in the model. The special cases are defined by the complementary viscous damping properties.

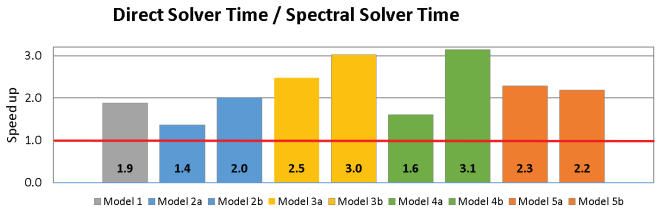
Case 1	No viscous damping	No material, global, or modal viscous damping
		No dashpot elements, connector damping, or substructures with viscous damping
		No matrix input data for the viscous damping

		No other possible sources of the viscous damping
Case 2	Low rank viscous damping operator	Viscous damping is defined locally in the model, for example, using dashpot elements, connector damping, or substructures with viscous damping
		Structural-acoustic coupled steady-state dynamic analysis using uncoupled eigenmodes when the number of acoustic modes is significantly smaller than the number of the structural modes
		Modal viscous damping defined for selected modes

You can use the spectral solver for mechanical, acoustic, and structural-acoustic analyses. Using the spectral solver can be beneficial for models with the mentioned special damping properties when many eigenmodes and many points in the frequency domain are used for the frequency response calculation.

The spectral solver algorithm consists of two analysis phases: the Setup Solver Phase and the Apply Solver Phase. The computational cost of the Setup Solver Phase depends on the number of the eigenmodes, and it can be significant when that number is high. The computational cost of the Apply Solver Phase is proportional to the number of the frequency points specified for the frequency response calculation, and the solution calculation at a frequency point is relatively cheap. Therefore, the spectral solver can be less effective for the analyses with a small number of frequency points (when the setup phase dominates), but it can be very effective for the same model when more frequency points are used.

The figure below shows the performance benefits from using the new spectral modal solver over the conventional (default) solver that solves the system of equations with a dense complex matrix and multiple right hand sides at every frequency point.



It compares the wall-time ratios for a steady-state dynamic analysis using the conventional and the new spectral solvers for the following models:

Model		DOF	Load Cases	Eigenmodes	Frequency Points	Damping Properties	Cores
1	Body-in-white	3.6 million	120	12200	1000	Material structural damping; no viscous damping	40

Model		DOF	Load Cases	Eigenmodes		Frequency Points	Damping Properties	Cores
2	Full vehicle	9.7 million	165	8800 uncoupled eigenmodes including 1000 acoustic modes		1000 2000	Material structural damping; viscous damping — 6 dashpots	40
3	Body-in-white	35.6 million	21	33800		1000	Material structural damping; viscous damping —70 dashpots	40 56
4	Body-in-trim	2.6 million	110	11000+	Uncoupled structural and acoustic eigenmodes Fully coupled structural-acoustic eigenmodes	1000	Material structural damping; viscous damping — 6 dashpots	
5	Fuselage model	48.3 million	120	21300		1000	Material structural damping; no viscous damping	40 56

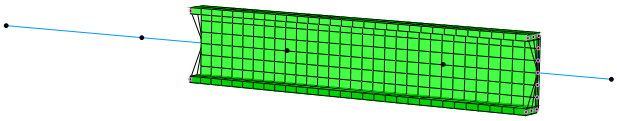
Beam-Based Submodeling from Global Beam Elements to Shell Elements

You can now perform beam-to- shell submodeling.

Benefits: Using shell elements in a submodel allows you to gain more detailed insight on results of interest for a region of a thin-walled beam that is modeled with beam elements in a global model.

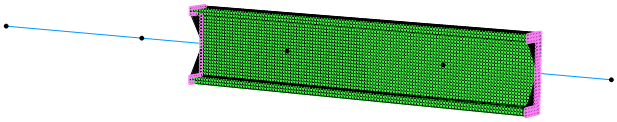
Beam-to-shell submodeling is often a good strategy to study details in one or more local regions if a global model uses beam elements.

The figure below shows a local shell-element model superimposed with beam elements from a global model. The shell thickness is rendered in the figure.



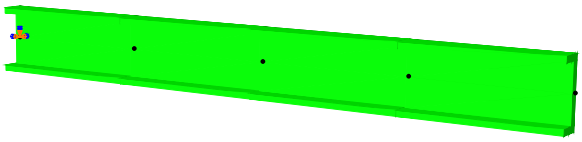
This simple example is provided to demonstrate the concept; typically, the global model would be larger and more complex than shown here.

Beam-to-shell submodeling is similar to the beam-to-solid submodeling capability introduced in Abaqus 2023 FD03; in both cases distributing couplings are used to impose conditions from the global model at submodel cuts. For comparison, the figure below shows a local solid-element model superimposed with beam elements from a global model.

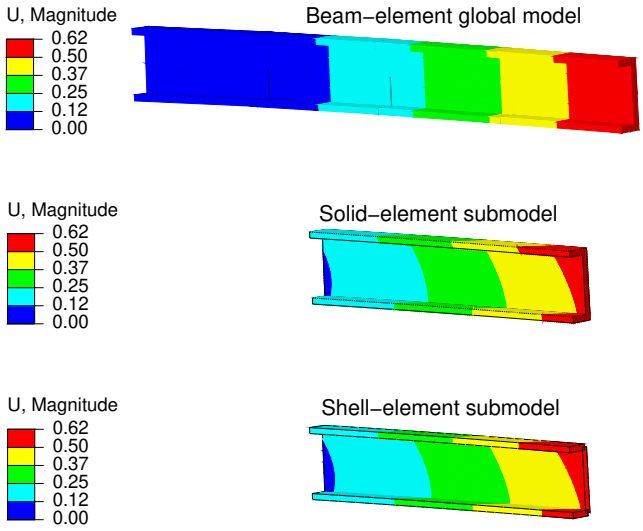


Representations of distributing couplings at submodel cuts are shown in both figures above. Reference nodes of these distributing couplings are used to impose conditions from the global model on the submodel.

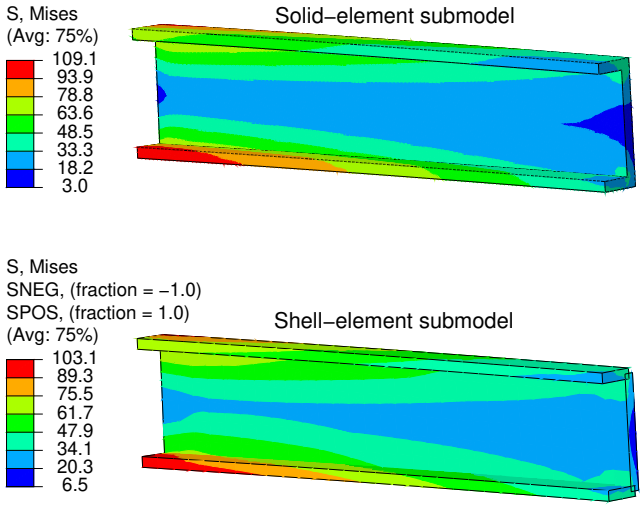
The figure below shows the global model with the beam cross-sections rendered.



The figure below provides a comparison of displacement contour plots for the beam-element global model and each of the submodels.



The figure below shows Mises stress contour plots for each of the submodels.



For more information, see *General Submodeling*.

2024 GA

Python 3

Abaqus Python is now based on Python 3.10 instead of Python 2.7. For more information about this major change and what it means for Abaqus users, visit *go.3ds.com/AbaqusPython*.

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

Formation of Tie Constraints in Import Analysis

In an Abaqus/Explicit import analysis, Abaqus/Explicit now forms tie constraints using the original instead of the updated configuration if the definitions of both the main and secondary surfaces to be tied are unchanged from the original analysis.

Benefits: The new protocol makes it easier to establish initial equilibrium due to tie constraints in the imported model.
The new protocol prevents the formation of new tie constraints for nodes on the surfaces that were previously untied. Such new tie constraints, if formed, can adversely affect the consistency of the imported model. For example, if the previously untied nodes are in a state of contact, tying these nodes results in deletion of the contact state associated with the nodes.

The new protocol also guarantees formation of the same tie constraints for the surfaces in the import analysis as in the original analysis. It is used regardless of whether you specify the original or the updated configuration as the reference configuration for import.

For more information, see *About Transferring Results between Abaqus Analyses*.

Adjoint Sensitivity Analysis for Sizing Optimization in Direct Steady-State Dynamic Analysis

You can now compute adjoint sensitivities for select node responses defined over node sets for a direct steady-state dynamic analysis to perform sizing optimization.

Benefits: You can now optimize for the steady-state response (direct-solution) of a structure subjected to harmonic excitations.

Abaqus/Standard now supports sizing optimization for select design variables (beam radii and shell thicknesses) in direct steady-state dynamic analyses for select design responses (nodal displacements/velocities/accelerations/rotations). You can define the range of frequencies over which you wish to compute the sensitivities for a particular design response.

You can use this new capability only for jobs run on a single domain and for node sets with nodes corresponding only to independent degrees of freedom.

For more information, see *Adjoint Design Sensitivity Analysis Overview*.

Benchmarks

This page describes recent changes in Abaqus Benchmarks.

2026 FD02

Constitutive Artificial Neural Networks (CANNs) Models

New tests verify the capability to define anisotropic hyperelastic models using constitutive artificial neural networks (CANNs) models.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Anisotropic hyperelastic modeling of arterial layers*.

2026 GA

Fatigue Life and Crack Path Predictions in Modified Compact Tension Specimens

New tests verify, validate, and illustrate the use of fatigue crack growth with the adaptive remeshing workflow.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Fatigue life and crack path predictions in modified compact tension specimens*.

2025 FD01

PDF Guides Available

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

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

For more information, see *Abaqus Benchmarks Guide*.

2024 FD03

Fatigue Crack Growth Analysis Using Field Import–Based Submodeling Approach

New tests verify fatigue crack growth analysis using both the traditional node-based submodeling approach and the field import–based submodeling approach.

Benefits: The tests verify the capability in Abaqus/Standard.
A single edge notch subjected to pure mode II cyclic fatigue loading is carried out using both the traditional node-based submodeling approach and the field import–based submodeling approach. Similar results are obtained when comparing the crack profiles obtained using both approaches.

For more information, see *Crack propagation of a single-edge notch simulated using XFEM*.

Using Low-Density Foam with Eulerian Elements and ALE Analysis

New tests verify the use of low-density foam with Eulerian elements and ALE adaptive meshing.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Indentation with different materials*, *Wave propagation with different materials*, and *Adaptivity patch test with different materials*.

Constraints

This page describes recent changes in Abaqus Constraints.

2025 FD01

Coupling of Reference Node Rotations for Distributing Coupling Constraints

For distributed couplings Abaqus/Explicit can now automatically determine whether to couple the reference node rotations to cloud node translations only or to both cloud node translations and rotations.

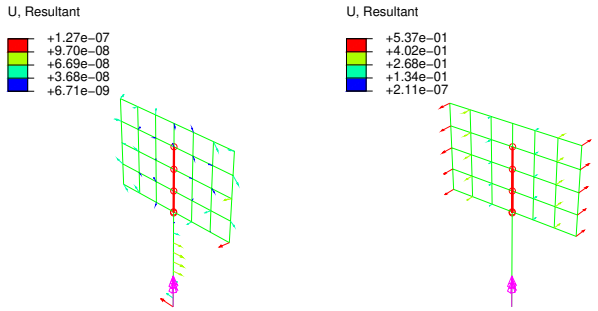
Benefits: This enhancement simplifies the use of distributed couplings and facilitates accurate, efficient solutions.

In earlier releases, Abaqus/Explicit always coupled reference node rotations to cloud node translations only, an approach that has undesired characteristics if all cloud nodes are positioned along a line. In this release, Abaqus can form valid coupling constraints even if cloud nodes are collinear. The new setting takes control of which terms are used in these couplings by default.

Now, Abaqus/Explicit can form valid coupling constraints if cloud nodes have rotational degrees of freedom regardless of the configuration of the cloud nodes. The new default setting takes control of which terms are used in these couplings by default.

The distributing coupling constraint in Abaqus/Explicit has been enhanced such that rotation of the reference node is coupled to cloud node rotations (if they exist) if the cloud nodes are nearly collinear. This capability already existed in Abaqus/Standard.

The left image below shows the results in earlier releases of Abaqus/Explicit. Abaqus/Explicit did not form a valid coupling between the reference node rotation and cloud node translations because of collinearity of the cloud nodes. Therefore, the four cloud nodes and the adjacent shell mesh do not rotate even though the reference node is rotating.



The right image illustrates the new behavior where Abaqus is able to form a valid coupling between the reference node rotation and the cloud system. Here the cloud node rotations are included in this coupling. The shell mesh spins around the cloud nodes consistent with the reference node rotation, which is the desired behavior.

By default, cloud node rotations in both Abaqus/Standard and Abaqus/Explicit now automatically influence reference node rotations only if the cloud nodes are deemed to be nearly collinear. You can dictate whether or not the reference node rotations are influenced by the cloud node rotations. You need to take care in choosing

the appropriate behavior. Including these extra terms when they are not required to form a valid constraint can increase computation time without having a significant effect on results. However, not including these extra terms when they are required to form a valid constraint can lead to poor results.

For more information, see *Controlling Whether Cloud Node Rotations Influence Reference Node Rotation* and **DISTRIBUTING*.

PDF Guides Available

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

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

For more information, see *Abaqus Constraints Guide*.

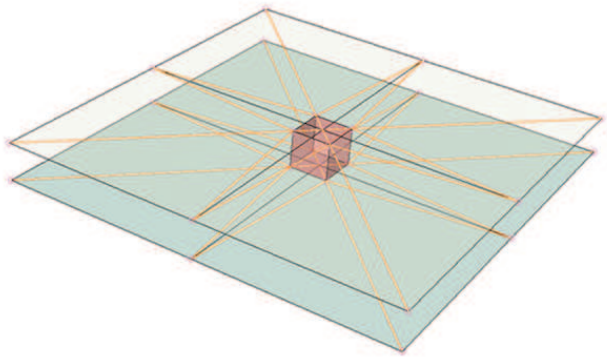
2024 FD01

Conversion of Select Distributed Couplings to Tie Constraints

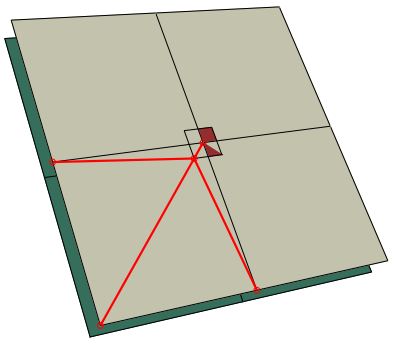
Abaqus/Explicit now converts distributing couplings to a tie constraint by default when the distributing couplings have cloud nodes associated with a single element facet and when the reference node of the coupling is not a node of a connector element.

Benefits: This enhancement avoids the creation of overlapping distributed couplings (which introduce a small amount of artificial mass for numerical purposes) when modeling fasteners using solid elements.

Some workflows involve fastening two surfaces together with the stiffness of the connection represented by a hexahedral element instead of a connector element, as shown in the image below.

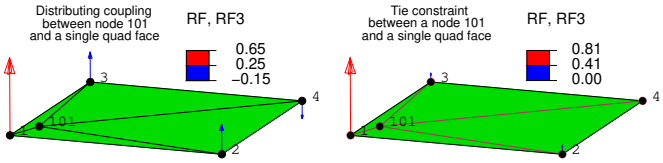


Each node of the hexahedral element acts as a reference node of a distributing coupling whose cloud nodes are associated with a single shell element, as shown in the image below.



This approach results in some shell nodes participating in two or four distributing couplings. Abaqus/Explicit generally introduces artificial mass to nodes acting as clouds nodes in multiple distributing couplings. By changing the default behavior for automatic conversion of distributing couplings to tie constraints, Abaqus/Explicit reduces the likelihood of adding artificial mass to a model.

Constraint coefficients are somewhat different for distributed couplings than they are for tie constraints in such cases. The image below provides perspective on these differences.



A downward unit force is applied at node 101. Node 101 is connected to a single square face with a distributed coupling for the case on the left versus a tie constraint for the case on the right. The four nodes of the square face are fixed with boundary conditions. Node 101 corresponds to the position one-tenth along the diagonal line from nodes 1 to 4. The vertical reaction forces (RF3) for nodes 1 through 4 are 0.65, 0.25, 0.25, and -0.15 for the distributing coupling case; and for the tie constraint case these reaction forces are 0.81, 0.09, 0.09, and 0.01, respectively. These sets of reaction forces are statically equivalent (both sets of reaction forces sum to 1.0 and do not generate a moment about the location of node 101). Because the constraint coefficients are different in the two approaches, some differences in the local behavior around the constraint are expected due to this conversion.

You can override this conversion from distributed couplings to tie constraints if desired.

The recommended technique for modeling compliant fasteners is to use connector elements instead of hexahedral “nuggets.” This enables the use of larger numbers of cloud nodes per coupling, which reduces mesh sensitivity without having nodes act as cloud nodes in multiple overlapping distributed couplings.

For more information, see *Converting a Distributing Coupling Attached to a Single Element Facet to a Tie Constraint*.

Coupled Simulation Interface

This page describes recent changes in Coupled Simulation Interface.

2026 FD04

PDF Guide Available

You can download a PDF version of this guide from the guide overview.

Benefits: You can easily access the PDF version of this guide.

For more information, see *Coupled Simulation Interface*.

2026 FD03

New Guide

This is the first release of the Coupled Simulation Interface guide.

Benefits: The Coupled Simulation Interface guide describes the SIMULIA Co-Simulation Services that enables you to perform coupled multiphysics and multiscale simulations between your application and the Dassault Systèmes physics solvers.

Elements

This page describes recent changes in Abaqus Elements.

2026 GA

Coupled Temperature Displacement Elements with Incompatible Modes

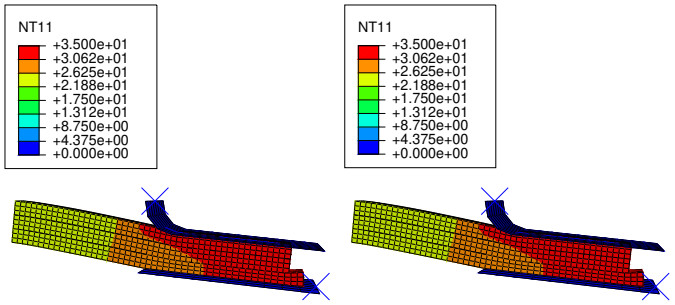
New incompatible elements with temperature degrees of freedom are now available in Abaqus/Standard (C3D8IT and C3D8IHT) and in Abaqus/Explicit (C3D8IT).

Benefits: The new elements provide improved bending behavior in coupled thermal-stress analysis of flexible structures.

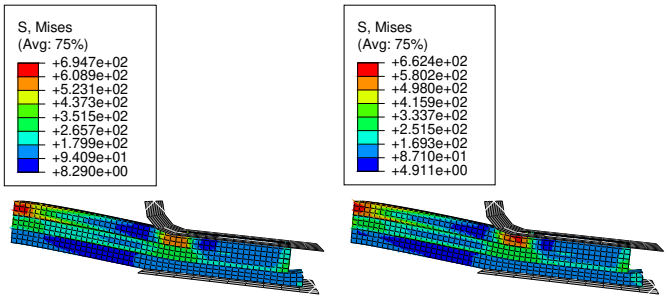
The incompatible mode element available in earlier releases, C3D8I, does not have temperature as a degree of freedom and cannot be used in heat transfer analyses. The C3D8T element has a temperature degree of freedom; however, this element is overly stiff when modeling bending deformation and is prone to shear locking. The new C3D8IT and C3D8IHT elements combine both temperature as a degree of freedom and incompatible modes, which makes them better options for modeling heat transfer in structures that undergo bending deformation.

The following example demonstrates the modeling of a beam-shaped structure in a coupled thermal-stress analysis. The beam structure contains three layers with different material properties, and contact gap conductance is defined between the adjacent layers. The bottom rigid surface holding the beam structure is fixed, and the top rigid surface is pushed down to introduce bending deformation of the beam structure during the analysis. A temperature boundary condition is also applied to the top surface of the beam structure once the top rigid surface is being pushed down.

For comparison purposes, both C3D8T and the new C3D8IT elements are used to model the beam structure. As shown below with C3D8T elements (left) and C3D8IT elements (right), the two thermal responses (node temperatures) are in close agreement.



However, the mechanical responses (stresses) are slightly different as shown below with C3D8T elements (left) and C3D8IT elements (right).



The C3D8T elements produce a stiffer response, which is expected since the beam structure undergoes bending deformation. The incompatible modes of the C3D8IT elements are activated during the analysis.

For more information, see *Three-Dimensional Solid Element Library*.

2025 FD01

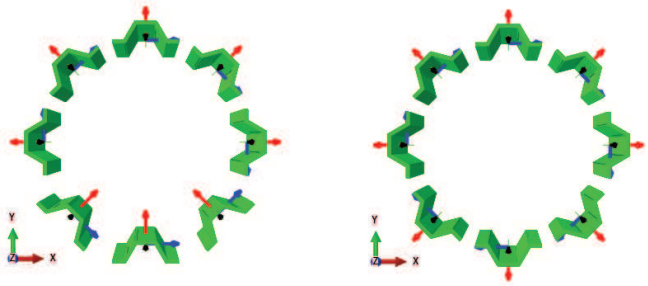
Fixed Nodal Normals for Beams in Space

You can now enforce the nodal normals for beams in space to be exactly as you specify them in the model definition.

Benefits: This new feature prevents nodal normals from getting reversed in certain special (rule-based) situations.

For beams in space you can define the nodal normal either by giving its direction cosines as part of each node definition or by giving them in a user-specified normal definition. By default, in certain situations, the nodal normal $\mathbf{n}_2$ that you specified might be automatically reversed by Abaqus (see *Defining Nodal Normals*). To prevent such default behavior, you can now enforce that the normal used by Abaqus is exactly as you defined it (and not its reverse).

In the example below, the user specified that all the beam normals point radially outward. However, as shown at left, by default three of the beam normals are automatically reversed by Abaqus during the preprocessing phase. These normals end up pointing radially inward, which leads to discontinuities in the beam normal in different regions of the model. You can use the new capability to prevent this from happening, as shown at right.



For more information, see **NORMAL* and *Beam Element Cross-Section Orientation*.

Surface Element Thickness

You can now define a constant section thickness as part of the surface section definition.

Benefits: Defining surface elements with a nonzero thickness enables you to better model contact in a similar way as shell and membrane elements.

User-defined surface elements have zero element thickness by default, but when used in contact, it is preferable to specify a thickness. In cases where surface elements are internally generated, those elements will have a nonzero thickness.

The specified thickness only affects contact calculations. Other element calculations related to rebar layers are unaffected.

For more information, see *Surface Elements* and **SURFACE SECTION*.

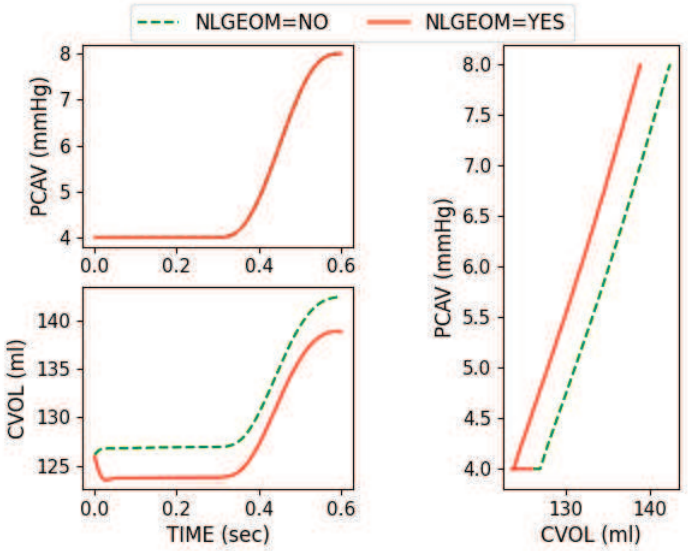
Geometrically Linear Analysis for C3D10 Elements

Geometrically linear analysis is now supported for C3D10 elements in Abaqus/Explicit.

Benefits: The ability to use C3D10 elements for the preload analysis of the Living Heart model provides more accurate initial stress states in the heart tissue.

The Living Heart model uses a preload analysis to compute stresses and corresponding stretches in the heart tissue corresponding to the initial loaded state. This preload analysis is performed using the geometrically linear option to calculate proper initial stress states in equilibrium with the applied loads in the reference configuration of the heart.

The following example demonstrates the effect of geometric linearity and geometric nonlinearity in the preload analysis on the pressure-volume response of a heart chamber. It consists of a preload analysis to compute the initial stress state of the heart tissue for the given applied cavity pressure, followed by a heartbeat analysis computing the response of the heart chamber as a function of the biological loads. Both geometric linearity and geometric nonlinearity are used to calculate the initial stress state in the preload analyses, which is then transferred to the subsequent heartbeat analyses. Pressure and volume versus time, as well as pressure versus volume results are compared between these two cases in the heartbeat analyses. As shown in the figure below, when geometric nonlinearity is used in the preload analysis, it shows a sharp drop in the heart chamber volume at the beginning of the heartbeat analysis; when geometric linearity is used in the preload analysis, the heart chamber volume results are more accurate and do not exhibit a sharp change.



For more information, see *Beginning a step in an Abaqus/Explicit analysis*.

PDF Guides Available

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

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

For more information, see *Abaqus Elements Guide*.

2024 FD03

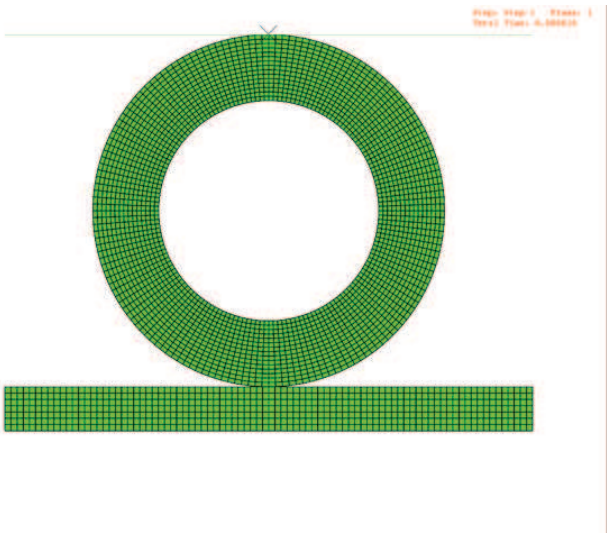
Bending Stabilization for Full-Integration Linear Elements

You can now control the activation and the level of bending stabilization for full-integration linear elements.

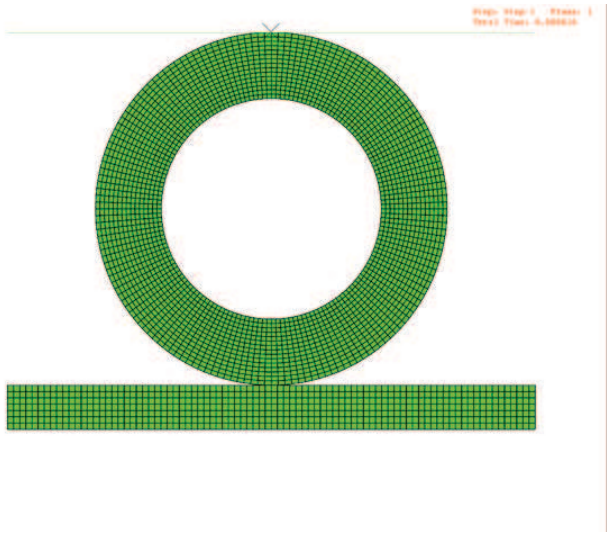
Benefits: Bending stabilization improves robustness for models using full-integration linear elements modeled with hyperelastic, hyperfoam, low-density foam, or user-defined materials and subjected to large bending deformations.

For full-integration linear elements modeled with hyperelastic, hyperfoam, low-density foam, or user-defined materials, spurious unstable patterns might develop in some situations due to excessively soft bending behavior. The formulation for this class of problems includes a bending stabilization term to improve the element performance. In earlier releases, the bending stabilization was active by default for built-in hyperelastic, hyperfoam, and low-density foam materials, but it was not active for user-defined materials. This could lead to different outcomes (convergence behavior and results) when you modeled the same material behavior using either a built-in material model or a user-defined material model. In addition, you had no control over the activation/deactivation of the stabilization algorithm or on the degree (magnitude) of stabilization.

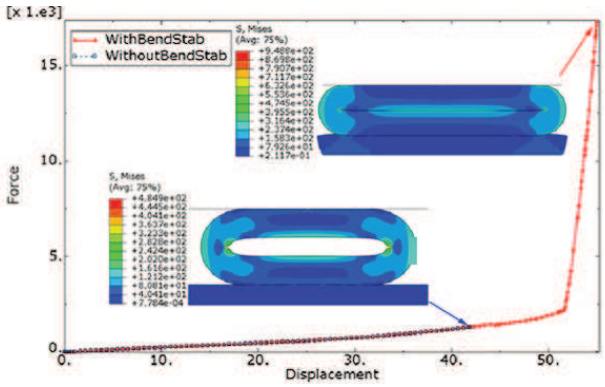
The example below shows a two-dimensional rubber ring compressed by a rigid plate against a base that is also made out of rubber. The image below shows the results from modeling a compressed O-ring that does not make use of the bending stabilization term. The analysis runs into convergence difficulties and fails to converge following the initiation of self-contact in the ring.



Conversely, the image below shows the results from a simulation that utilizes bending stabilization. In this situation, the analysis completes.



The image below shows the force-displacement behavior of the rigid plate's reference node for the two models (with and without bending stabilization).



The two responses are in close agreement until convergence is lost in the model without stabilization, demonstrating that apart from improving convergence, the bending stabilization's impact on the overall model response is small. The bending stabilization term is very important for problems involving a relatively soft material subjected to large bending deformations.

For more information, see *Using Section Controls for Bending Stabilization*.

2024 FD01

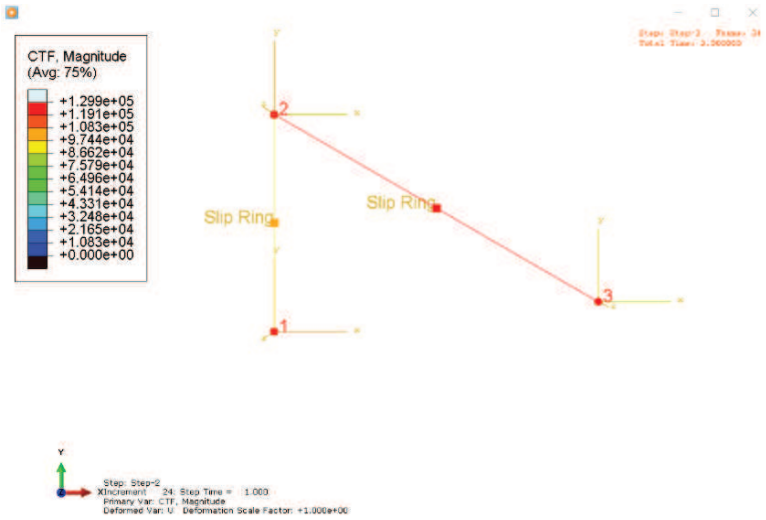
Varying Contact Angles for SLIPRING Connectors

You can now request that Abaqus/Standard compute the contact angle of a SLIPRING connector at the beginning of the analysis and update it through the analysis.

Benefits: If the slipring connector models a belt-pulley assembly, basing the calculations on the current value of the contact angle more accurately calculates the frictional force between the pulley and the belt.

In earlier releases, Abaqus/Standard assumed the SLIPRING contact angle to be either a constant (specified) value or zero (the default).

The animation below shows a belt-pulley system between nodes 1, 2, and 3.



The belt is modeled using two slipring connector elements and passes over a pulley at node 2. The frictional force between the belt and the pulley depends on the contact angle between them. In this example, the contact angle changes as nodes 1 and 3 are displaced from their original positions. Abaqus/Standard now accounts for the change in this contact angle for the friction calculations.

For more information, see *SLIPRING*.

Example Problems

This page describes recent changes in Abaqus Example Problems.

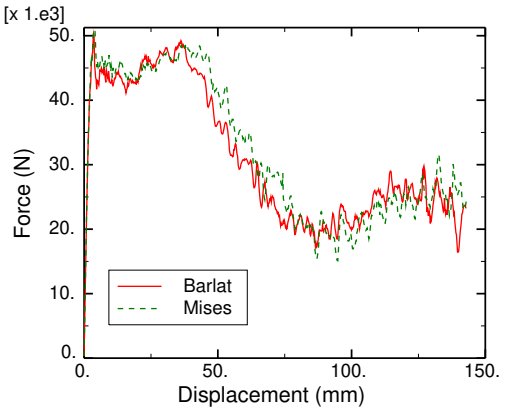
2026 FD04

Progressive Failure Analysis of Thin-Wall Aluminum Extrusion Under Quasi-Static and Dynamic Loads

The Barlat yield criterion is added to this example to model the plastic anisotropy.

Benefits: The new test demonstrates how to use the Barlat yield criterion to model the plastic anisotropy in an extruded aluminum alloy.

Extruded aluminum alloys display plastic orthotropy. In earlier releases, this example neglected this anisotropy. Now, the example includes the Barlat yield criterion with experimental data-fitted coefficients. The plot below shows close agreement with previous isotropic models, which validates the initial assumption of material isotropy for this specific example.



For more information, see *Progressive failure analysis of thin-wall aluminum extrusion under quasi-static and dynamic loads*.

Modeling Sodium Ion Batteries Using a Fully Coupled Thermal-Electrochemical-Structural Analysis

A new example demonstrates the capability of the fully coupled thermal-electrochemical-structural procedure for modeling coupled multiphysics-multiscale behaviors in sodium ion batteries under mechanical constraints.

Benefits: The test demonstrates the capability of a fully coupled thermal-electrochemical-structural analysis in Abaqus/Standard.

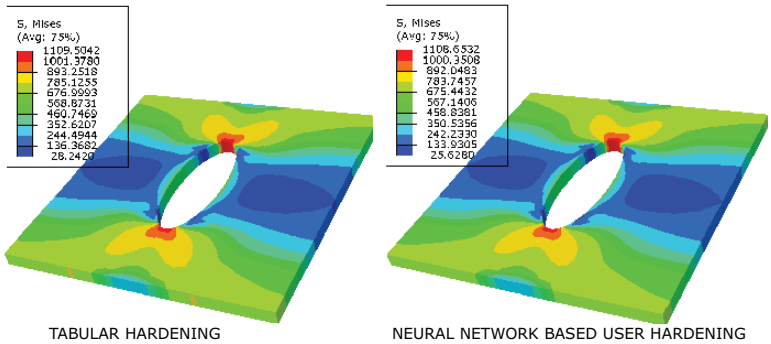
For more information, see *Fully coupled thermal-electrochemical-structural analysis in sodium ion batteries*.

2026 FD02

Plate with Elliptical Hole Under Thermal Loading

A new example illustrates an application of neural networks in the context of user plastic hardening.

Benefits: The test demonstrates the use of neural networks in Abaqus/Standard.



TABULAR HARDENING NEURAL NETWORK BASED USER HARDENING

A trained feed-forward neural network is employed in an Abaqus/Standard analysis to provide temperature-dependent plastic hardening. The defined neural network is evaluated in user subroutine `UHARD`. Results are compared to an analysis using an equivalent tabular hardening definition.

For more information, see *Plate with elliptical hole under thermal loading*.

2025 FD03

Fully Coupled Thermal-Electrochemical-Structural Analysis in Batteries

A new example demonstrates the capability of a fully coupled thermal-electrochemical-structural analysis for modeling coupled multiphysics-multiscale behaviors in lithium-ion batteries under mechanical constraints.

Benefits: The test demonstrates the capability of a fully coupled thermal-electrochemical-structural analysis in Abaqus/Standard.

For more information, see *Fully coupled thermal-electrochemical-structural analysis in lithium-ion batteries*.

Tread Wear Simulation in Abaqus/Standard

The example is updated to illustrate a new, simpler approach to modeling tread wear in a steady-state transport procedure.

Benefits: The new approach demonstrates the ease of setting up the model for wear calculations in a steady-state transport procedure and of visualizing wear distances.
In the new approach, you specify wear properties on contact surfaces as usual. However, wear calculations in the steady-state transport procedure are made internally; you do not need to code a user subroutine as you did in the approach available in earlier releases. In addition, the new approach enables you to visualize wear output.

For more information, see *Tread wear simulation in Abaqus/Standard*.

2025 FD02

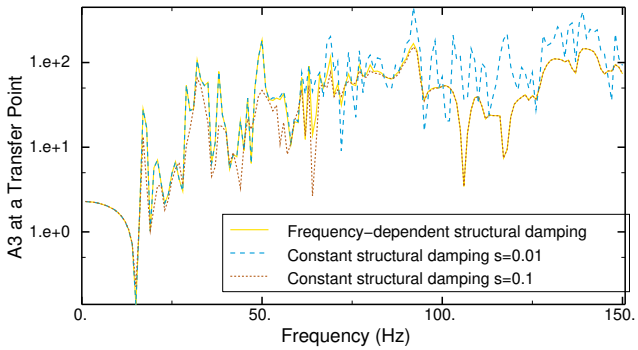
Steady-State Dynamic Analysis of a Vehicle Body-in-White Model

A new example demonstrates the use of frequency-dependent structural damping in the steady-state dynamic analysis of a vehicle body-in-white model.

Benefits: The test demonstrates the use of frequency-dependent structural damping in a steady-state dynamic analysis in Abaqus/Standard.

In the new example, the frequency-dependence of the structural damping is such that s = 0.01 for frequencies lower than 50 Hz and s = 0.1 for frequencies higher than 100 Hz, with a linear transition between the two regions.

The figure below shows the transfer inertance at the node located at the bottom of the driver's seat. As expected, the transfer inertance with frequency-dependent structural damping is identical to that for a constant s = 0.01 structural damping for frequencies lower than 50 Hz and that for a constant s = 0.1 structural damping for frequencies higher than 100 Hz.

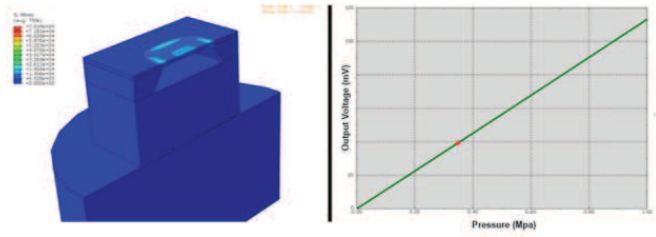


For more information, see *Steady-state dynamic analysis of a vehicle body-in-white model*.

Modeling of a Silicon Piezoresistive Pressure Sensor

A new example illustrates the use of a piezoresistive material model to simulate a silicon piezoresistive pressure sensor.

Benefits: The test demonstrates the use of a piezoresistive material model in Abaqus/Standard.



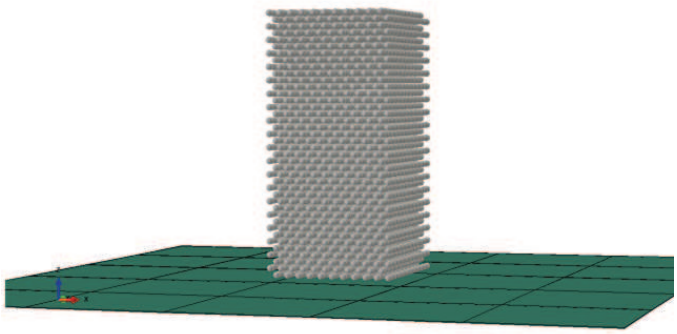
For more information, see *Modeling of a silicon piezoresistive pressure sensor*.

2025 FD01

Particle Cluster Drop

A new example demonstrates the use of particle clustering to describe the nonspherical shape of limestone aggregates that are dropped onto a rigid floor under gravity loading.

Benefits: The test demonstrates the use of clustering in the discrete element method in Abaqus/Standard.



For more information, see *Particle cluster drop*.

PDF Guides Available

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

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

For more information, see *Abaqus Example Problems Guide*.

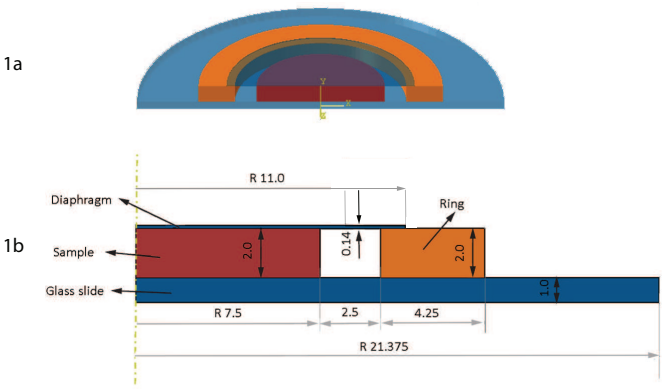
2024 FD04

Modeling the Watts Test of Adhesives

A new example demonstrates how to model the curing process of adhesives.

Benefits: The tests demonstrate the adhesive curing process modeling capability in Abaqus/Standard.

The Watts test measures the polymerization shrinkage of materials. In the Watts test, a disc-shaped specimen is sandwiched between two glass plates, as shown below.



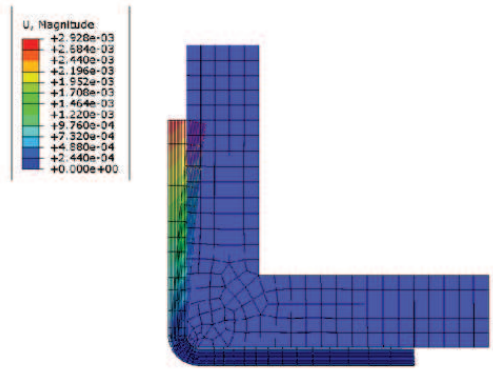
For more information, see *Modeling the Watts test of adhesives*.

2024 FD03

Cure Process Simulation: Prediction of Cure-Induced Spring-In of an Angle Bracket

A new example demonstrates the use of multiscale technology combined with cure modeling in Abaqus/Standard to model the cure-induced spring-in deformation of an angle bracket.

Benefits: The tests demonstrate the composite curing process modeling capability in Abaqus/Standard.



For more information, see *Prediction of cure-induced spring-in of an angle bracket*.

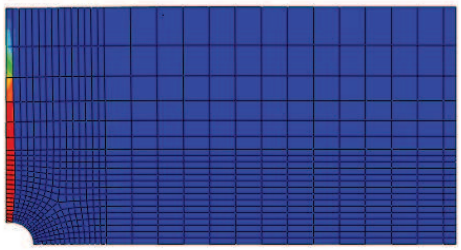
2024 FD01

Failure of Blunt Notched Fiber Metal Laminates

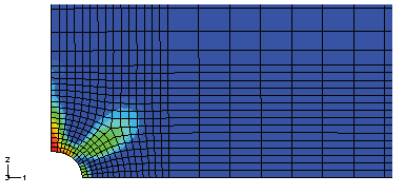
A new test simulates failure and damage in a fiber metal laminate containing a blunt notch subjected to quasi-static loading.

Benefits: The tests demonstrate the capability in Abaqus/Standard.
Multiscale modeling is used.

The image below shows the fiber damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using the Hashin damage model, DAMAGEFT contour plot).



The image below shows the matrix damage pattern in the 0° fiber-reinforced epoxy layer for the 0° loading direction (using the Hashin damage model, DAMAGEMT contour plot).



For more information, see *Failure of blunt notched fiber metal laminates*.

Execution

This page describes recent changes in Abaqus Execution.

2026 FD01

Enhanced Substructure Results Recovery

During substructure element recovery matrix generation, you can now recover substructure results at multiple section points and at all output positions supported in Abaqus/Standard. In addition, you can now generate the recovery matrices for exterior 3D elements.

Benefits: You can recover results at integration points or nodes, support multiple shell and beam section points, improve node-averaged results, output substructure modes, and accelerate writing recovered results to the output database.
You can use the enhanced element recovery matrices in conjunction with the **abaqus substructurerecover** utility to recover results within a substructure for static or dynamic time-domain or frequency-domain solutions obtained from Abaqus/Standard and from the flexible body dynamics workflows based on substructures generated by Abaqus. The **abaqus substructurerecover** utility has been enhanced substantially to take advantage of the enhanced recovery capabilities.

The following features were enhanced or enabled:

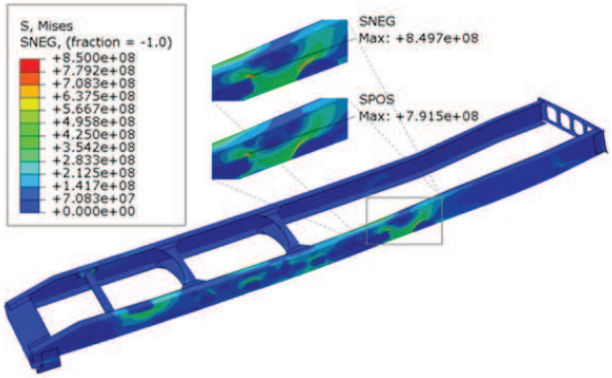
- Visualization of element results for multiple shell and beam element section points
- Element results output at integration points and at element nodes
- Improved calculation for results averaged at nodes
- Output results for the substructure modes
- Improved performance of writing the recovered results to the output database

For example, you can now use the **abaqus substructurerecover** utility to write an output database that contains stress results for the static and dynamic modes of a substructure, as shown below for the model of a cross hinge.



When combined with the temporal evolution of the substructure generalized displacements for representative load cycles, this output database can serve as input to a downstream analysis with fatigue software such as fe-safe.

You can also now use the utility to write recovered stress and strain for shell elements at all section points or at a selection of section points, the same way you would when you output it for the elements of a full finite element model, as shown in the example below of a truck frame model.



For more information, see *Recovering Results from a Substructure*.

2025 FD04

LS-DYNA to Abaqus Translator Enhancements

You can now translate nonstructural mass and additional set keywords using the LS-DYNA translator. The translation of piecewise linear plastic material now includes rate dependency. Imposition of initial velocity to a rigid body now includes its reference node.

Benefits: The enhancements improve the usability of the translator.

Translation of the following LS-Dyna keywords is now supported:

`*ELEMENT_MASS_PART`

`*ELEMENT_MASS_PART_SET`

`*SET_PART_ADD`

`*SET_NODE_ADD`

In the translation of `*MAT_PIECEWISE_LINEAR_PLASTICITY`, rate dependency data is translated based on the C, P and LCSS parameters associated with the keyword. The translation of `*PARAMETER` and `*PARAMETER_EXPRESSION` is revised to include use of the parameter names in the Abaqus data and is now supported in the translation of `*INITIAL_VELOCITY`. In addition, the imposition of initial velocity to a rigid body now includes the reference node generated in the translation.

For more information, see *Translating LS-DYNA Data Files to Abaqus Input Files*.

2025 FD02

Improved Substructure Results Recovery for Abaqus-Simpack Flexible Body Dynamics

You can now recover results obtained from Simpack dynamic time domain and frequency domain analyses.

Benefits: The enhancements improve the Abaqus-Simpack workflow for flexible body dynamic analysis. The **abaqus substructurerecover** utility now supports recovery of results obtained from Simpack dynamic time domain and frequency domain analyses.

For more information, see *Environment File Settings* and *Recovering Results from a Substructure*.

2025 FD01

Environment File Setting Customizes Warnings Level

You can now specify whether common warnings are reported by the Abaqus analysis input file processor.

Benefits: Reducing the number of warnings makes evaluation of the preprocessed model more convenient. The Abaqus analysis input file processor now suppresses common warnings for unsupported output requests by default. However, you can use the new environment variable `output_warnings_level` to specify if common warnings are reported.

Many warnings issued by the Abaqus analysis input file processor for invalid output requests, such as those issued for output requests that are not supported for specific element or procedure types, are informational only and warrant no action by the user. For large models the number of these warnings can be very high and can make it difficult to evaluate of the preprocessed model.

For more information, see *Environment File Settings*.

PDF Guides Available

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

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

For more information, see *Abaqus Execution Guide*.

2024 FD04

Optimization and Parallel Execution

For parallel execution using GPGPU acceleration, you now specify the number of GPUs as the total number of GPUs for the analysis rather than specifying the number of GPUs per MPI process.

Benefits: Specifying GPGPU acceleration using the total number of GPUs is more consistent with how you specify the number of CPUs allocated to an analysis and permits greater flexibility for job execution in heterogeneous compute environments.

For more information, see *Abaqus/Standard and Abaqus/Explicit Execution*, *Optimization Execution*, *Environment File Settings*, *Parallel Execution in Abaqus/Standard*, and *Parallel Execution in Abaqus/Explicit*.

2024 FD03

Direct Sparse Solver Support for Steps with Unsymmetric Matrix Storage

For shared memory parallel (SMP) execution of steady-state dynamic direct procedures with unsymmetric matrix storage, Abaqus/Standard now uses the pure thread-based complex solver by default.

Benefits: The pure thread-based complex solver provides a solution that is typically at least twice as fast as the previous default option, the hybrid direct sparse solver.

For more information, see *Parallel Execution in Abaqus/Standard*.

2024 FD02

Execution Procedures User Assistance Reorganized

Content in the Execution Procedures section of the Abaqus Execution Guide is now divided into several subsections.

Benefits: The new organization makes it easier to locate user assistance for the **abaqus** commands you want to run.
Topics are now divided into the following subsections:

- *Abaqus Execution*
- *Abaqus Co-Simulation*
- *Abaqus Utilities*
- *Translating Other Solver Input to Abaqus Input*
- *Translating Abaqus Output to Another Format*

For more information, see *Execution Procedures*.

2024 FD01

Enhanced Translation of LS-DYNA Keyword Files to Abaqus

You can now translate LS-DYNA keyword files that use the i10 format option in reading integer data.

Benefits: The ability to read these data improves the usability of the **abaqus fromdyna** translator.
You can specify the format for individual keywords by appending the percent sign at the end of the keyword, as shown in the following examples:

- `*ELEMENT_BEAM %`
- `*ELEMENT_SHELL %`
- `*ELEMENT_SOLID %`
- `*NODE %`

In addition, you can specify the format globally for all applicable keywords in the LS-DYNA keyword file, by adding an option to the execution command of the translator or by declaring the following keyword option:

```
*KEYWORD i10=y
```

For more information, see *Translating LS-DYNA Data Files to Abaqus Input Files*.

Getting Started with Abaqus/CAE

This page describes recent changes in Getting Started with Abaqus/CAE.

2025 FD01

PDF Guides Available

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

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

For more information, see *Getting Started with Abaqus/CAE*.

GUI Toolkit

This page describes recent changes in Abaqus GUI Toolkit.

2025 FD01

PDF Guides Available

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

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

For more information, see *Abaqus GUI Toolkit User's Guide*.

Interactions

This page describes recent changes in Abaqus Interactions.

2026 FD04

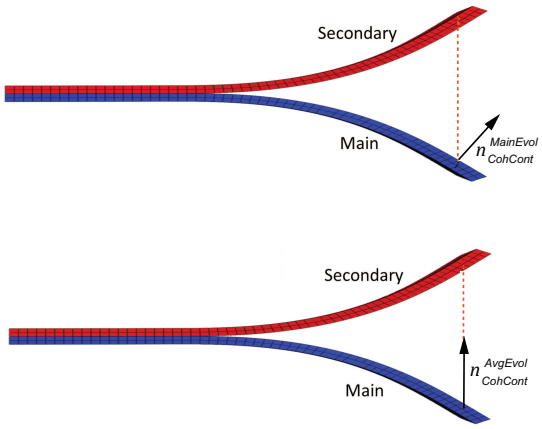
Cohesive Contact Calculations Involving Large Cohesive Strains

When modeling ductile cohesive contact bonds in Abaqus/Explicit, you can now specify that the main and secondary surfaces both influence the evolution of the cohesive contact normal direction equally.

Benefits: Modeling of ductile cohesive contact bonds in Abaqus/Explicit provides results more similar to cohesive element modeling.

In earlier releases, the contact normal direction for Abaqus/Explicit contact with cohesive behavior was always based solely on the main surface. This behavior remains the default. However, you can now specify that the cohesive contact normal direction be based on equal influence of the evolution normal directions of both the main and secondary surfaces.

The figure below shows deformed configurations for peeling test simulations with ductile cohesive contact bonds. The two parts are flat initially (not shown) with cohesive contact along the entire length of the interface. Symmetry exists in the mesh refinement and loading conditions. In the top image, the cohesive contact normal direction evolves with only the main surface such that the cohesive strain develops normal and shear components. In the bottom image, the cohesive strain is aligned with the cohesive contact normal direction, which evolves taking into consideration changes in both the main and secondary surface normal directions, resulting in cohesive strain with no shear.



For more information, see **CONTACT CONTROLS ASSIGNMENT*, *Contact Cohesive Behavior*, and *Cohesive surface interaction*.

Modeling Threaded Bolt Connections with Surface Element Representations of Thread Geometry

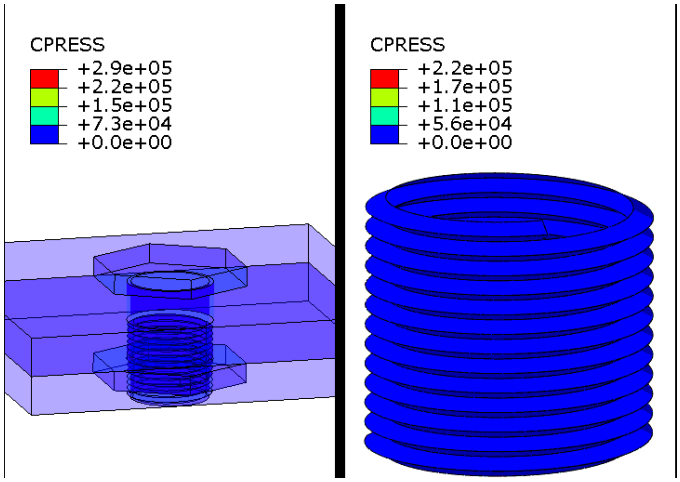
You can now specify an enhanced approach for modeling bolted connections with general contact in Abaqus/Standard that allows thread geometry to be created automatically using surface elements on top of nominal meshes.

Benefits: You can account for the true thread geometry for accurate and efficient modeling of large relative motions between the mating internal and external threads in bolt tightening and loosening simulations.

Refined solid meshes to capture thread geometry accurately on bolts and nuts in a bolted connection can make bolted connection simulations expensive. For efficiency, Abaqus allows you to specify nominal meshes with cylindrical surfaces to represent the internal and external mating surfaces for use with an approximate method to account for thread geometry. However, when large relative motion occurs between the threads during bolt tightening and loosening, the approximate method based on specifying bolt clearances with a small-sliding tracking approach might be inaccurate. The new approach generates thread meshes with surface elements that are constrained automatically to the user-defined nominal cylindrical meshes.

The generated thread meshes inherit the contact properties and initialization definitions from the nominal meshes. By default, finite-sliding contact is activated on the generated thread surfaces to model large relative motion accurately while maintaining efficiency.

The follow animation shows a bolt tightening simulation where the thread meshes are generated automatically for both the bolt and the nut based on the description of the thread geometry parameters:



For more information, see *About General Contact in Abaqus/Standard*.

2026 FD02

Mapping of Cohesive Contact State from Abaqus/Standard to Abaqus/Explicit Upon Import

You can now import the cohesive contact status and damage state from Abaqus/Standard to Abaqus/Explicit.

Benefits: You can more accurately model bonding behavior using workflows that involve import from Abaqus/Standard to Abaqus/Explicit.

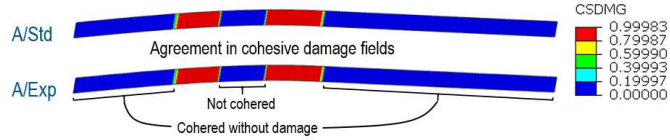
Cohesive contact status and damage are now mapped across import from Abaqus/Standard to Abaqus/Explicit, despite various differences in individual contact constraints between these two solvers. In earlier releases, the cohesive contact state (CSTATUS) was independently initialized in Abaqus/Explicit upon import from

Abaqus/Standard, such that CSTATUS would often switch to a "Not in contact" state where a tensile cohesive contact state existed prior to import.

The figure shows contour plots of CSTATUS of a partially bonded interface for an Abaqus/Standard simulation at the point of import (top) and the mapped state of the cohesive contact status upon import to Abaqus/Explicit (bottom).

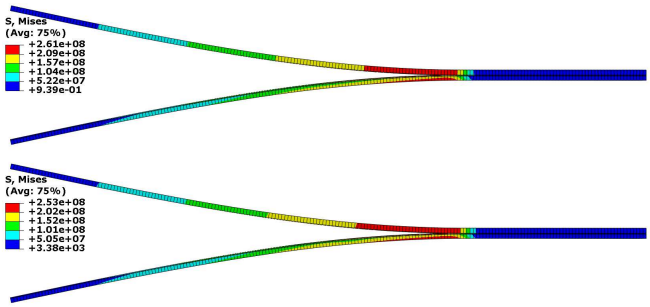


The figure below shows contour plots of cohesive contact damage (CSDMG) of a partially bonded interface for an Abaqus/Standard simulation at the point of import (top) and the mapped state of cohesive contact damage upon import to Abaqus/Explicit (bottom).



As expected, this enhancement results in consistent contour plots of contact status (CSTATUS) and cohesive damage (CSDMG) across Abaqus/Standard and Abaqus/Explicit import.

The figure below shows a comparison of Mises stress and overall delamination for the Abaqus/Standard simulation (top) and the imported Abaqus/Explicit simulation (bottom) after post-import loading.



For more information, see *Transferring Results between Abaqus/Explicit and Abaqus/Standard* and *Delamination analysis of laminated composites*.

Enhanced Contact Mass Scaling in Abaqus/Explicit

The nodal mass added due to contact mass scaling is now permanent.

Benefits: The new algorithm eliminates spurious oscillations and sudden instabilities in models with rapidly changing contact status.

Contact mass scaling is now applied permanently rather than incrementally during the analysis, which greatly improves stability and reduces artificial oscillations in models sensitive to contact chattering. The small permanent mass increase is generally negligible (and very localized) except for cases where the contact penalty stiffness is scaled up significantly.

For more information, see *Mass Scaling to Account for Contact Stiffness*.

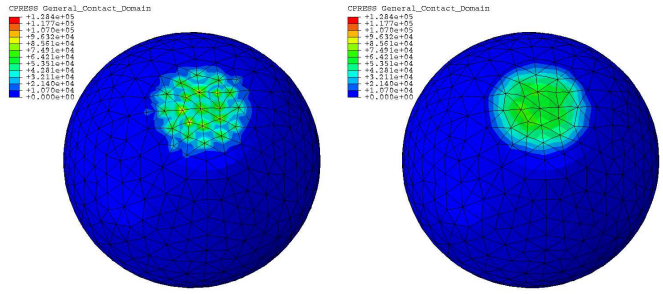
2026 FD01

Enhanced Contact Stress for C3D10 Elements in Abaqus/Explicit

Contact stresses are now smoothed during postprocessing operations for second-order tetrahedral elements (C3D10) in Abaqus/Explicit.

Benefits: The contact stress representation is more continuous and visually coherent, and it is consistent with the visualization in Abaqus/Standard.
Contact stress distribution is a key factor in many applications, including load transfer, wear, material failure, and interface behavior in mechanical analyses.

The example below illustrates a sphere compressed between two analytical rigid surfaces (not shown). The contact distributions clearly demonstrate the benefits of the new filtered output.



For more information, see *General Contact Output Variables*.

Separate Numerics and Physics in Contact Keyword Interface

Specific numerical controls of contact-related capabilities in the contact keyword interface have become separate from keywords that control the physics related to contact.

Benefits: The new contact keyword interface allows for numerical controls to be defined in more localized regions of the model.

The new contact keyword interface separates specific numerical contact controls from the physics controls. Physics related to contact continues being defined through interaction properties, while numerical controls are defined surface pairwise with contact controls assignment. Specifically, there are four numerical controls that have a new contact keyword interface: the contact enforcement method in Abaqus/Standard, the method of friction enforcement of the “stick” condition in Abaqus/Standard, the contact permeability constraint method in Abaqus/Standard, and the tracking thickness used to extend the range for which contact opening output is provided.

Th contact enforcement in Abaqus/Standard can be defined as direct, augmented Lagrange, linear penalty, or nonlinear penalty. The new surface pairwise method using contact controls assignment is mutually exclusive with the interaction property method of defining enforcement. For example, an error message occurs if user specified contact enforcement defined is using both the new interface and the previous interface for the same surface pairings in the contact domain.

The friction enforcement of the “stick” condition in Abaqus/Standard can be defined as penalty with a unitless slip tolerance, penalty with a slip tolerance distance, and Lagrange. The new surface pairwise method using contact controls assignment is mutually exclusive with the interaction property method of defining friction enforcement of the “stick” condition. For example, an error message occurs if user specified friction enforcement defined is using both the new interface and the previous interface for the same surface pairings in the contact domain.

The contact permeability constraint method in Abaqus/Standard is defined as direct or penalty. The new surface pairwise method using contact controls assignment is mutually exclusive with the interaction property method of defining the contact permeability constraint method. For example, an error message occurs if user specified contact permeability constraint method defined is using both the new interface and the previous interface for the same surface pairings in the contact domain.

The tracking thickness assigns a thickness that determines the contact tracking distance between surfaces. The new surface pairwise method using contact controls assignment is mutually exclusive with the interaction property method of defining the tracking thickness. For example, an error message occurs if the user specified tracking thickness defined is using both the new interface and the previous interface for the same surface pairings in the contact domain.

For more information, see **CONTACT CONTROLS ASSIGNMENT*, *Contact Constraint Enforcement Methods in Abaqus/Standard*, *Frictional Behavior*, *Pore Fluid Contact Properties*, and *About General Contact in Abaqus/Explicit*.

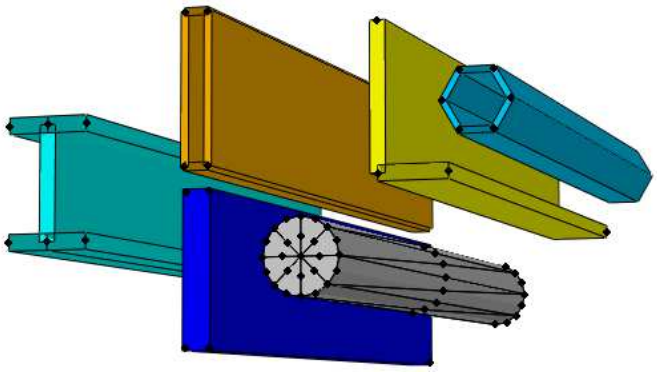
2026 GA

Contact Surface Representation for Beams with Circular Cross-Sections in Abaqus/Standard

You can configure Abaqus/Standard to internally generate surface elements that represent the circular cross-section of beams and pipes for use in general contact calculations.

Benefits: You can achieve more accurate modeling of beam ends and obtain detailed contact output on the contact surface.

In earlier releases, Abaqus/Standard automatically generated a mesh of contact faces, nodes, and edges near the exterior of noncircular beam cross-sections by default. Now, you can configure Abaqus/Standard to internally generate surface elements that represent the circular cross-sections of beams and pipes for general contact calculations. The image below shows contact surface representations for beams with various cross-sections based on surface element meshes that are internally generated by Abaqus/Standard for general contact.



For more information, see *About General Contact in Abaqus/Standard* and **CONTACT CONTROLS ASSIGNMENT*.

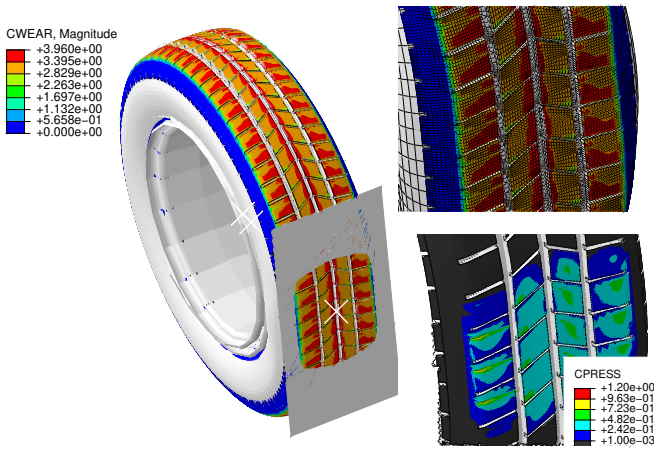
2025 FD03

Contact Wear with Steady-State Transport in Abaqus/Standard

You can now specify surface wear properties directly in steady-state transport analyses, and Abaqus/Standard calculates the wear rate. You can request output of the resulting total wear distances.

Benefits: It is much simpler to simulate wear accumulation in steady-state transport analyses with wear coefficients specified as surface properties and wear calculations done by Abaqus, compared to a preexisting approach involving user subroutine *UMESHMOTION*.

The animation below shows a passenger car tire rolling at 70 km/h and accumulating wear over a distance of 58,880 km. The 3D tire model is generated from periodic symmetric-model generation of a wedge with repeating tread configurations. Surface wear properties are specified on the tread surface. The wear distance contours show that a maximum wear distance of 3.96 mm is accumulated, which is approximately 63% of the tread depth. The animation also shows that the contact footprint expands as wear accumulates and the wear rate slows down as the contact foot print expands due to redistribution of contact stresses. In the wear calculations shown here, wear distances are treated as offsets for contact penetration calculations and the nodal locations of the underlying elements due to wear are not modified. With this built-in capability, Abaqus manages complex details involving streamline calculations that would be very challenging to manage within a user subroutine.



For more information, see *Contact Wear*, *Contact wear modeling*, and *Tread wear simulation in Abaqus/Standard*.

2025 FD01

Contact Involving Beams with Noncircular Cross-Sections in Abaqus/Standard

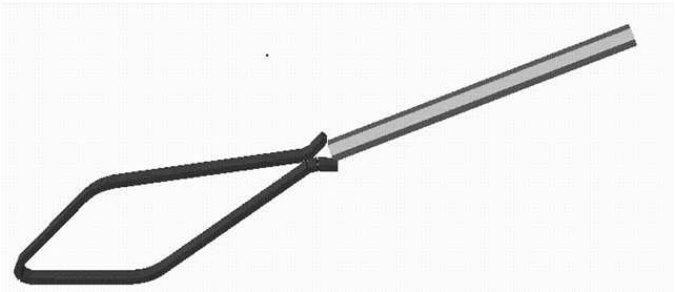
Contact calculations now consider realistic beam cross-section geometry in Abaqus/Standard by default. This change makes the behavior in Abaqus/Standard consistent with Abaqus/Explicit.

Benefits: The new default contact treatment of beams with noncircular cross-sections is more realistic.

In earlier releases, Abaqus/Standard approximately represented beams in contact calculations with nodes and edges of the beam mesh having a radius encompassing the actual beam cross-section.

The new default beam representation for contact uses an automatically generated mesh of contact faces, nodes, and edges close to the exterior of noncircular beam cross-sections. Some surface thickness is assigned automatically, which provides an accurate contact representation of the beam geometry with some rounding of edges.

The animation below shows a hexagonal cross-section beam inserted into the end of a part resembling tweezers and then rotated. Both parts are modeled with beam elements. In this example, the beam elements of the tweezers have a rectangular cross-section. The response of the tweezers to the insertion and rotation of the other part is intuitive. Realistic representation of the beam contact surfaces is crucial to the accuracy of this simulation.



You can revert to the previous default behavior, as discussed in **CONTACT CONTROLS ASSIGNMENT*.

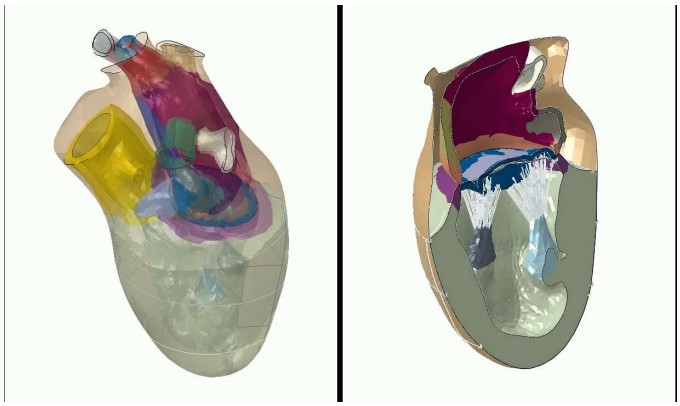
For more information, see *About General Contact in Abaqus/Standard* and **CONTACT CONTROLS ASSIGNMENT*.

No Separation Contact in Abaqus/Explicit

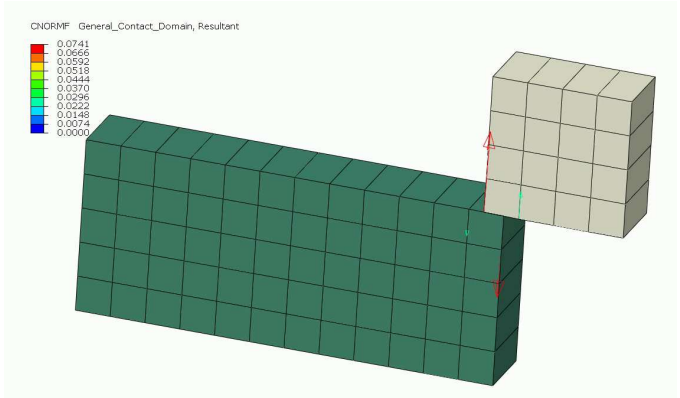
When using general contact in Abaqus/Explicit, you can now define a no separation contact relationship (a contact pressure-overclosure behavior that prevents surfaces from separating in the normal direction once they have come into contact) without the need to use a rough frictional model.

Benefits: You can solve new classes of problems in which surfaces in contact remain in contact but are able to slide tangentially relative to one another.

The animation below shows two views of an Abaqus/Explicit beating heart simulation. As the heart contracts in this simulation, it remains in contact with, and slides relative to, the pericardium (membrane sac). Despite tensile contact stress developing, it represents the effects of fluid within the pericardial cavity (although this fluid is not modeled directly in this simulation). The ability for contact to resist separation as sliding occurs is crucial to obtaining realistic behavior in this simulation. In earlier releases, you could only specify no-separation behavior in combination with “rough” friction for general contact in Abaqus/Explicit.



However, “no separation” contact behavior in the normal direction does not prevent loss of contact for general contact in Abaqus/Explicit when one surface slides off another surface, as shown in the animation below.



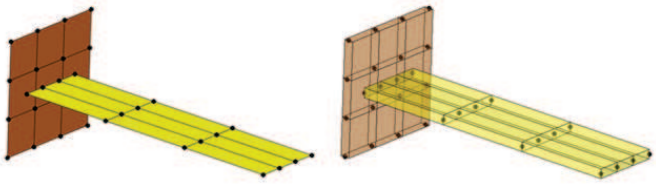
For more information, see *Using the No Separation Relationship* and **SURFACE BEHAVIOR*.

Contact Surface Representation for Shells in Abaqus/Standard

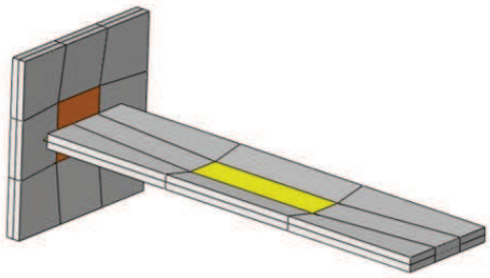
You can specify that Abaqus/Standard internally generate surface elements representing the shell perimeter for contact calculations in general contact.

Benefits: You can more accurately model the shell perimeter.

The default contact surface representation for a part modeled with shell elements is a double-sided shell reference surface that considers the shell thickness and shell offset. The example below shows two bodies meshed with shell elements in contact. The perimeter of one part is in contact with the shell element faces of the other part. The left image shows no thickness rendering, while the right image includes thickness rendering.



Now, you can specify that Abaqus/Standard internally generate surface elements near shell perimeters, as shown in the example below. Surface elements are created right at the shell perimeter, with an additional row of elements extending inward toward the shell interior, ensuring a continuous reference surface representation of the shell. These surface elements can be removed from display in the postprocessor to return to a view such as in the left image above.



For more information, see *About General Contact in Abaqus/Standard* and **CONTACT CONTROLS ASSIGNMENT*.

PDF Guides Available

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

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

For more information, see *Abaqus Interactions Guide*.

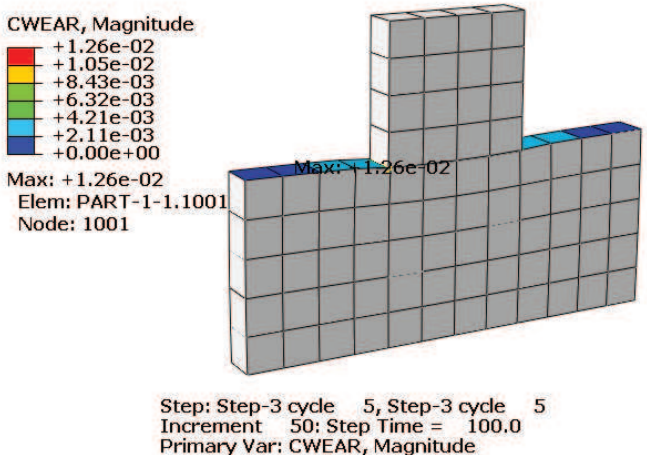
2025 GA

Step Cycling for Contact Wear Simulations

You can now use step cycling and step cycling controls to avoid explicitly specifying each instance of repeated steps associated with wear accumulation and to allow the number of physical wear cycles represented per simulated cycle to evolve based on current wear rates in Abaqus/Standard.

Benefits: Step cycling makes it easier to specify the model and enables more computationally efficient wear simulations.

In many contact wear workflows, surface wear accumulates over thousands of physical wear cycles. It is often prohibitively expensive to simulate each physical wear cycle, so each simulated cycle typically represents a "batch" of multiple similar physical cycles. The animation below shows a wear simulation over five simulated cycles that represent 110,000 physical wear cycles in total. Step cycling allows the "batch size" to evolve during the simulation based on the maximum incremental wear distance in the simulated cycle. The number of simulated cycles is often not known before running such a simulation. For each simulated cycle, the user-specified wear coefficients are scaled by the current batch size.



The animation shows the wear distance accumulation on the slider and the bottom block as the slider repeatedly moves back and forth. In this example, the simulated wear cycle is modeled in two steps:

- In “Step-2” the slider moves to the right and returns to the center.
- In “Step-3” the slider moves to the left and returns to the center.

These two steps are repeated using step cycling until a threshold wear distance exceeds 0.012 or reaches a cap of 10 repetitions of the simulated wear cycle. The batch size for scaling the wear coefficients has an initial value of 10,000 and a cap of 25,000. The target incremental wear distance is 0.006 accumulated per simulated cycle.

The batch size for the next simulated wear cycle is internally calculated so that the estimated maximum wear accumulation is less than the target value of 0.006 while respecting the batch size cap. The example simulation reaches the threshold wear distance in 5 simulated cycles. The batch size for the initial cycle is 10,000 based on user input. For the subsequent cycles, the batch sizes are calculated to be 25,000 (the cap value) while maintaining the maximum wear accumulated in each simulated cycle to be less than 0.006.

For more information, see *Contact Wear*, **STEP CYCLING*, and **STEP CYCLING CONTROL*.

Contact Involving Beams with Noncircular Cross-Sections in Abaqus/Explicit

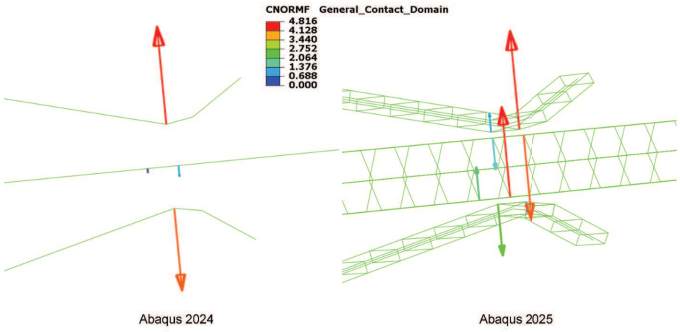
The contact surface representation of beams with noncircular cross-sections in Abaqus/Explicit now uses internally generated meshes of surface elements that are available for postprocessing.

Benefits: The new contact surface representation is consistent with Abaqus/Standard, and it more accurately and realistically represents contact output.

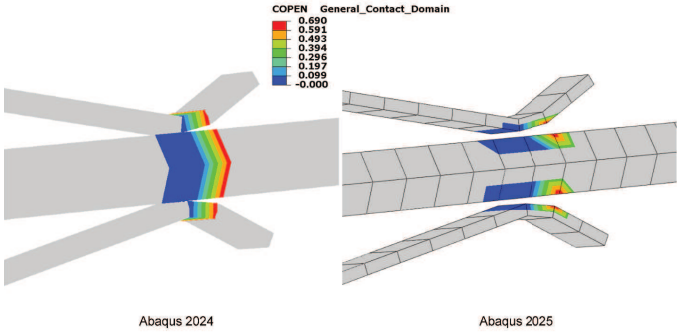
In earlier releases of Abaqus/Explicit, similar internally generated meshes could be used for simulations but they were not available for postprocessing. The images below compare plots of contact forces and contact opening distances on beam elements to plots on internally generated surface-element representations of the beams. Plots on surface elements show variations over the surfaces, similar to plots that would occur on the outer surface of a solid body, whereas plots based on beam elements are limited in that they represent a single value at each axial location.

Furthermore, some rounding of edges in the contact representation to improve numerics is achieved by offsetting nodes of internally generated surface elements inward and adding a small corresponding contact thickness. This contact thickness is not represented in plots, so the parts appear a bit smaller than the actual size.

The following image shows the tweezers beam model contact force vector plot across versions (beam rendering is not available for vector plots):



The following plot shows the tweezers beam model contact opening distance output in previous versions (beam rendering shown for Abaqus 2024):



For more information, see *About General Contact in Abaqus/Explicit*, *About General Contact in Abaqus/Standard*, and **CONTACT CONTROLS ASSIGNMENT*.

2024 FD03

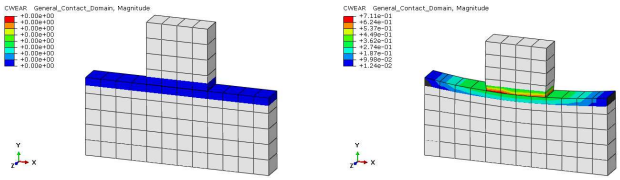
Arbitrary Lagrangian-Eulerian (ALE) Adaptive Meshing Supports Contact Wear in Abaqus/Explicit

You can now use arbitrary Lagrangian-Eulerian (ALE) adaptive meshing together with wear in Abaqus/Explicit, which accounts for local surface wear distances in underlying element calculations.

Benefits: Users can more easily and accurately account for wear distances that are large compared to element dimensions.

The capability to simulate wear in general contact in Abaqus/Explicit was introduced in 2024 FD02. In that release Abaqus/Explicit treated the evolution of wear exclusively by considering nodal wear distances in contact penetration calculations. Now you can use arbitrary Lagrangian-Eulerian (ALE) adaptive meshing together with wear in Abaqus/Explicit to account for surface wear (erosion) in part geometries while preserving mesh quality, even for wear distances on the order of element dimensions.

The simulation on the left below shows the preexisting behavior (wear distances accounted for with offsets in contact penetration calculations). The simulation on the right demonstrates the use of ALE in the context of wear modeling. Similar wear accumulation occurs for both models, but it is much easier to interpret the wear configurations with ALE. Most wear occurs on the smaller block in this model, with the maximum wear distance being approximately half of the original element depth. In this example, ALE mesh smoothing effectively provides a realistic representation of the evolving part geometry while maintaining good mesh quality.



For more information, see *Contact Wear* and **ADAPTIVE MESH*.

2024 FD02

Cohesive Contact in Abaqus/Explicit

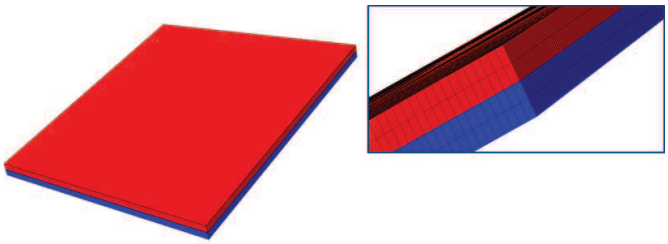
Performance for Abaqus/Explicit models with cohesive contact is improved compared to earlier releases. In addition, scaling the default contact penalty stiffness now also scales the default contact cohesive stiffness in Abaqus/Explicit; this behavior is consistent with that in Abaqus/Standard.

Benefits: The improvements facilitate more efficient user workflows and user adjustments to increase or decrease the contact cohesive stiffness from its default value.

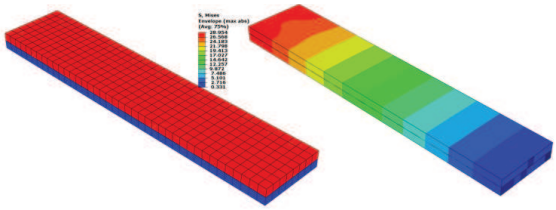
Enhancements to cohesive contact in Abaqus/Explicit make cohesive contact a preferred alternative to surface-based tie constraints for modeling permanent, stiff bonds in more cases (for example, models where ties result in large implicit constraint systems or when you require interfacial traction results across connected surfaces).

Performance improvements for cohesive contact are most noticeable for models with a large fraction of nodes participating in cohesive contact. For example, simulations for the models shown in the images below with permanent, stiff cohesive contact bonds have sped up by 13% and 12%, respectively, compared to simulations in 2024 FD01.

The following figure shows two plates modeled with solid elements bonded together with permanent, stiff cohesive contact:

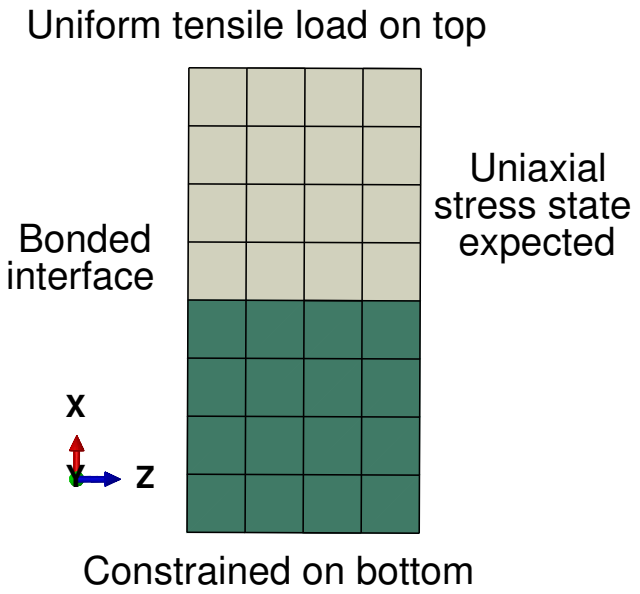


The following figure shows two plates modeled with shell elements (shown with shell thickness rendering) bonded together with permanent, stiff cohesive contact:

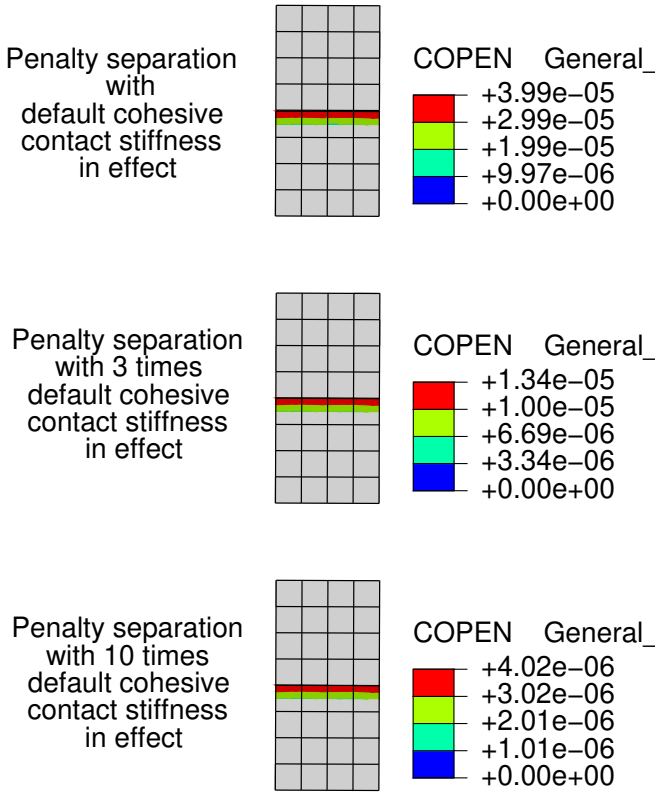


A second enhancement allows the default cohesive stiffness to be scaled by existing options to scale the contact penalty stiffness in Abaqus/Explicit. Cohesive contact, like other contact constraints, has a finite bond stiffness. You can assign the bond stiffness or allow it to take on the same contact penalty stiffness that would be in effect to resist penetrations for regular contact behavior, as discussed in *Contact Cohesive Behavior*. You can use a scale factor to increase the cohesive stiffness relative to the default value.

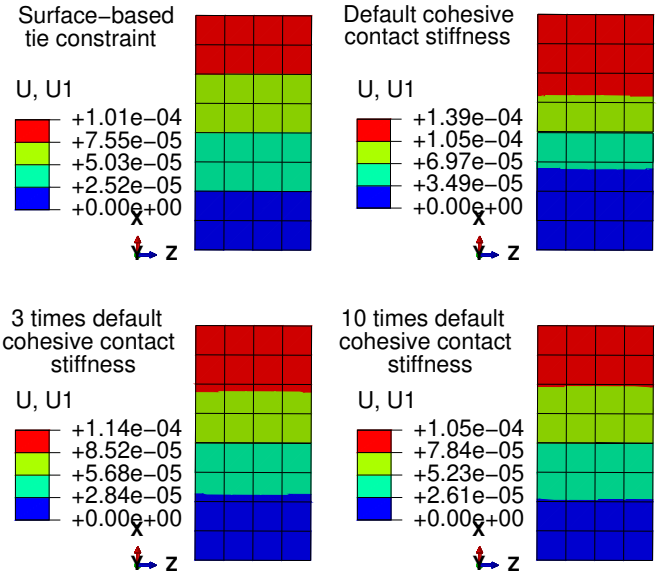
The image below shows two blocks bonded together with uniform tensile (negative pressure) loading of 1×10^6 N/m^2 applied to the top face of the top block while the bottom of the bottom block is constrained. Loading is smoothly ramped on and damping is included to obtain a quasi-static solution.



The image below shows the vertical stress (S11) for three model variants. The axial stress results correspond to the expected uniaxial stress solution regardless of details of how the bond is modeled.



The figure below shows the vertical displacement (U1) for four model variants. Some additional displacement occurs when a penalty stiffness is used to enforce the bond. Increasing the penalty stiffness via a scale factor is a convenient mechanism to control the resistance to relative motion across the bond.



The mesh is quite coarse in this example. The default penalty stiffness would be greater with a more refined mesh. Increasing the penalty stiffness tends to decrease the time increment size in Abaqus/Explicit unless contact mass scaling is used (see *Mass Scaling to Account for Contact Stiffness*).

Alternatively, you can directly specify cohesive stiffness components, as discussed in *Cohesive Stiffness*. When you specify the cohesive stiffness, the penalty stiffness scale factor has no effect.

For more information, see *Contact Cohesive Behavior*, **COHESIVE BEHAVIOR*, **CONTACT CONTROLS ASSIGNMENT*, **CONTACT MASS SCALING*, *Contact Controls for General Contact in Abaqus/Explicit*, and *Mass Scaling*.

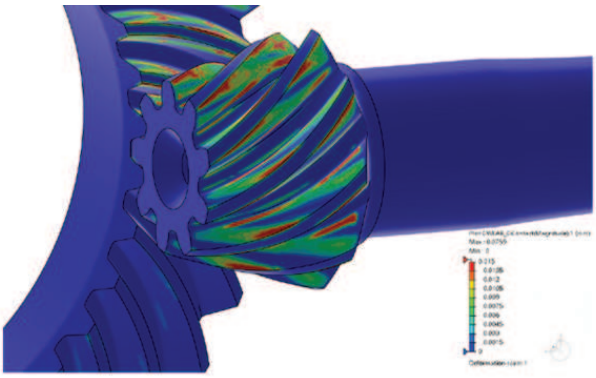
Contact Wear in Abaqus/Explicit

Abaqus/Explicit can now model the evolution of wear and visualize nodal wear distances on contact surfaces due to mechanical contact based on the Archard's wear equation.

Benefits: You can now model wear accumulated at the contact level in Abaqus/Explicit.

This new capability is available in Abaqus/Explicit for general contact analysis involving solid elements. Similar to the implementation introduced for Abaqus/Standard in 2024 FD01, the modeling approach accounts for local surface wear distances in contact penetration calculations but not in underlying element calculations.

The image below shows an example of accumulation of wear (shown as CWEAR contour output) in a simplified helical bevel gear analysis.



For more information, see *Contact Wear*, **SURFACE PROPERTY*, **SURFACE PROPERTY ASSIGNMENT*, and **WEAR SURFACE PROPERTIES*.

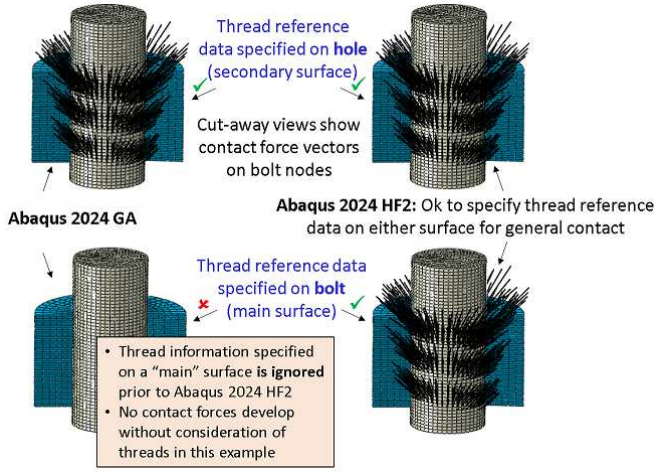
Specifying Reference Thread Geometry in Abaqus/Standard

Abaqus/Standard allows reference thread geometry to be specified on either surface (secondary or main) of a small-sliding interaction for general contact.

Benefits: Reference thread data for reference thread geometry can be specified for either surface of a small-sliding interaction of general contact.

Allowing reference thread geometry to be specified on either surface of a small-sliding interaction for general contact improves usability. Previously, this data was expected to be specified on whichever surface acting as the secondary surface of the contact formulation. However, main and secondary roles are typically left as default assignments for general contact in Abaqus/Standard, so a user typically would not know the secondary and main roles before running a simulation (and typically should not need to be aware of those roles).

The images below show cut-away views of the contact force vectors on the bolt nodes.The right side shows that results in Abaqus 2024 HF2 are insensitive to whether reference thread geometry is specified for the bolt or hole surface for small sliding interactions of general contact. In prior versions, the expected behavior was obtained only if reference thread geometry was specified for the secondary surface.



2024 FD01

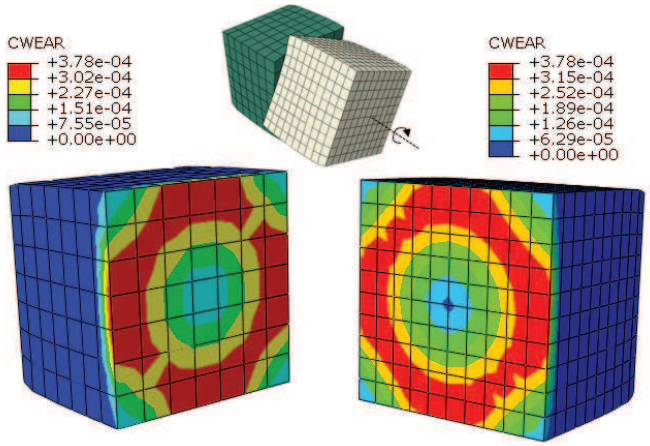
Contact Wear in Abaqus/Standard

You can now model the evolution of wear and visualize nodal wear distances on contact surfaces due to mechanical contact, based on the Archard's wear equation.

Benefits: Abaqus/Standard can now handle wear modeling workflows in automotive, aerospace, life sciences, and packaging applications.

In many workflows, mechanical contact can cause wear on contacting surfaces due to mechanisms like abrasion and fretting. The new wear capability in Abaqus/Standard models wear evolution based on the Archard's wear equation. Accordingly, contact nodal wear distances evolve based on wear properties of the surface material and on contact variables such as normal stress and slip distance. Abaqus/Standard treats wear distances as nodal offsets to the contact algorithm, resulting in a simplified approach to model wear fully within contact. It does not model the changes to geometry from wear explicitly. The approach considers the two-way physical influence of wear distance evolution on contact stresses (and underlying material stresses) and vice versa.

The figures below shows the wear accumulation on the contact surfaces for two blocks in contact.



Wear properties for the Archard's wear equation are specified on the contact surfaces. Nodal wear distances accumulate as the blocks slide relative to each other under compression. The nodal wear distances on the two surfaces are shown as CWEAR output contours at the end of the simulation.

For more information, see *Contact Wear*, *Contact wear modeling*, **SURFACE PROPERTY*, **SURFACE PROPERTY ASSIGNMENT*, and **WEAR SURFACE PROPERTIES*.

Keywords

This page describes recent changes in Abaqus Keywords.

2026 FD04

*CONTACT CONTROLS ASSIGNMENT

In Abaqus/Standard, you can use the new THREAD MESH GENERATION parameter to activate the surface element representations of the external and internal threads for general contact. In Abaqus/Explicit, you can now use the NORMAL EVOLUTION parameter to specify the method for evolving the contact normal direction.

Benefits: In Abaqus/Standard, you can more accurately simulate scenarios involving relatively large twisting motion between the external and the internal threads. In Abaqus/Explicit, modeling of ductile cohesive contact bonds is improved.
In Abaqus/Explicit, you can now specify that the cohesive contact normal direction should evolve based on considering the evolution of both the secondary and main surface normal directions when modeling ductile cohesive contact bonds. By default, the cohesive contact normal direction considers only the evolution of the main surface normal direction.

For more information, see **CONTACT CONTROLS ASSIGNMENT*.

*DSLOAD

You can now use an Eulerian material surface to define surface tension loads in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DSLOAD*.

*DYNAMIC

You can now use the RATE DEPENDENCE parameter to specify whether to consider or ignore material rate-dependence in an explicit dynamic step.

Benefits: This feature expands the capabilities of Abaqus/Explicit.
Ignoring material rate-dependence allows you to run explicit dynamic simulations with rate-independent (or long-term) material response without having to modify the material definition.

For more information, see **DYNAMIC*.

*DYNAMIC TEMPERATURE DISPLACEMENT

You can now use the RATE DEPENDENCE parameter to specify whether to consider or ignore material rate-dependence in a coupled dynamic thermal-stress step.

Benefits: This feature expands the capabilities of Abaqus/Explicit.
Ignoring material rate-dependence allows you to run explicit dynamic simulations with rate-independent (or long-term) material response without having to modify the material definition.

For more information, see **DYNAMIC TEMPERATURE-DISPLACEMENT*.

*POTENTIAL

You can now model nonquadratic yield in both Abaqus/Standard and Abaqus/Explicit. You can also define the Barlat Yld2000-2d yield surface by setting TYPE=BARLAT2000.

Benefits: The new Barlat Yld2000-2d yield surface expands the capabilities of Abaqus/Standard and Abaqus/Explicit and enables you to more accurately predict the forming behavior of sheet metals.

In earlier releases, nonquadratic yield was available only in Abaqus/Explicit.

Compared with Hill anisotropic plasticity, nonquadratic yield criteria such as Barlat anisotropic plasticity models provide flexibility for describing more complex anisotropic yield behavior.

For more information, see **POTENTIAL*.

*SUBMODEL LINEAR LOAD CASE REFERENCE (new)

Specify the name of the global model load case that refers to this submodel load case.

Benefits: Performance is improved for a single step that has multiple linear load cases in both the global model and submodel compared to multiple perturbations steps in both the global model and submodel.
Each perturbation step definition of a submodel can refer to the names of one or more linear load cases of a global model. Each named load case is used to process a separate perturbation instance.

For more information, see **SUBMODEL LINEAR LOAD CASE REFERENCE*.

*SUBMODEL TIME SYNCHRONIZATION (new)

Specify the step and increment number (if needed) in the global model to provide how the time in the submodel synchronizes with the time in the global model.

Benefits: This feature extends the capabilities of Abaqus/Standard. Separating the time synchronization from other aspects of submodel analysis setup allows you to more clearly understand the submodel setup.

The new keyword associates general steps and perturbation steps between a submodel and a global model in a well-defined manner.

By default, Abaqus/Standard now associates the *i*th general step of the submodel with the *i*th general step of the global model for general submodeling. If the submodel has more general steps than the global model, Abaqus/Standard issues an error message.

By default, Abaqus/Standard also now associates the *j*th perturbation step after the *i*th general step with the *j*th perturbation step after the *i*th general step of the glocal model. If the submodel has more perturbations steps than the global model after the corresponding general steps, Abaqus/Standard issues an error message.

For more information, see **SUBMODEL TIME SYNCHRONIZATION*.

*THREADED INTERFACE (new)

Specify the settings to be referenced in the surface element representation of the external and internal threads for general contact in Abaqus/Standard.

Benefits: This feature extends the capabilities of Abaqus/Standard by enabling you to accurately model scenarios involving relatively large twisting motion between the external and the internal threads.

For more information, see **THREADED INTERFACE*.

2026 FD03

*COUPLED TEMPERATURE-DISPLACEMENT

You can use the new THERMAL RUNAWAY parameter to model a thermal runaway event in a coupled temperature-displacement, electrochemical step.

Benefits: This feature expands the capabilities of Abaqus/Standard.

Thermal runaway of lithium-ion batteries impacts the usage and safety of rechargeable batteries. It is important to include this effect in battery simulations to engineer better and safer batteries.

This functionality can be used only with porous electrode batteries.

For more information, see **COUPLED TEMPERATURE-DISPLACEMENT*.

*COUPLED THERMAL-ELECTROCHEMICAL

You can use the new THERMAL RUNAWAY parameter to model a thermal runaway event in a coupled thermal-electrochemical step.

Benefits: This feature expands the capabilities of Abaqus/Standard.

Thermal runaway of lithium-ion batteries impacts the usage and safety of rechargeable batteries. It is important to include this effect in battery simulations to engineer better and safer batteries.

This functionality can be used only with porous electrode batteries.

For more information, see **COUPLED THERMAL-ELECTROCHEMICAL*.

*DISTRIBUTION

You can now define a varying distribution of particle radius and particle specific area per unit volume ($_s$) in a fully coupled electrochemical analysis to model lithium ion batteries.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DISTRIBUTION TABLE*.

*DISTRIBUTION TABLE

You can now use a distribution table to define a varying distribution of particle radius and particle specific area per unit volume ($_s$) in a fully coupled electrochemical analysis to model lithium ion batteries.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DISTRIBUTION TABLE*.

*ENERGY OUTPUT

The SURFACE parameter now supports contact surfaces.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ENERGY OUTPUT*.

*THERMAL RUNAWAY (new)

You can define the temperature at which thermal runaway can be activated in a material that uses porous electrode theory.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **THERMAL RUNAWAY*.

2026 FD02

*DISCRETE SECTION

You can use the new PARTICLE BEHAVIOR parameter to specify the name of the particle behavior associated with swelling of the discrete particles in a DEM analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DISCRETE SECTION*.

*NEURAL NETWORK (new)

Define the configuration of a feed-forward neural network in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.
You can evaluate a neural network from any user subroutine in Abaqus/Standard. You can use neural networks for various applications; for example, surrogate constitutive modeling, boundary conditions, etc.

This option must be used together with a single *NEURAL NETWORK DATA suboption.

For more information, see **NEURAL NETWORK*.

*NEURAL NETWORK DATA (new)

Specify the weights and biases (bulk data) for a neural network in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.
You can evaluate a neural network from any user subroutine in Abaqus/Standard. You can use neural networks for various applications.

This option must be used as a single suboption following a *NEURAL NETWORK option.

For more information, see **NEURAL NETWORK DATA*.

*PARTICLE BEHAVIOR (new)

Define particle behavior in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **PARTICLE BEHAVIOR*.

*PARTICLE SURFACE DEPOSITION (new)

Define particle surface deposition in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **PARTICLE SURFACE DEPOSITION*.

*PARTICLE SWELLING (new)

Define particle swelling in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **PARTICLE SWELLING*.

2026 FD01

*CONTACT CONTROLS ASSIGNMENT

The TYPE parameter can now take the values ENFORCEMENT METHOD, FRICTION ENFORCEMENT, PERMEABILITY CONSTRAINT METHOD, and TRACKING THICKNESS.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **CONTACT CONTROLS ASSIGNMENT*.

*CONTROLS

If you set the CONVERGE CHECK parameter equal to MODERATE, RELAXED, or CONCEPT DESIGN, Abaqus/Standard now uses a modified default value for the contact and slip compatibility tolerances.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTROLS*.

*DEBOND

You can use the new INTERACTION parameter to reference a surface interaction under which a fracture criterion is defined.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DEBOND*.

*ELASTIC

You can now use TYPE=BILAMINA in Abaqus/Standard. In earlier releases, this parameter value was available only in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ELASTIC*.

*ELEMENT RECOVERY MATRIX

You can now generate substructure stress and strain recovery data at all standard element output positions and at multiple section points, and you can restrict recovery matrices to exterior three-dimensional elements. Multiple *ELEMENT RECOVERY MATRIX requests can now be defined in the same step.

Benefits: You have improved control over substructure recovery workflows.
To generate substructure stress and strain recovery data at all standard element output positions, set the POSITION parameter equal to AVERAGED AT NODES, CENTROIDAL, INTEGRATION POINTS (the new default), or NODES.

To restrict recovery matrices to exterior three-dimensional elements, use the new EXTERIOR parameter.

For more information, see **ELEMENT RECOVERY MATRIX*.

*FOURIER TRANSFORM CONTROLS (new)

Specify the method to be used to calculate the Fourier transform as well as the scale factor that to be applied to the results of the calculations.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FOURIER TRANSFORM CONTROLS*.

*FRACTURE CRITERION

This keyword option was treated as history data for a debond crack propagation analysis in earlier releases. It is now treated as model data.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FRACTURE CRITERION*.

*FREQUENCY

The subspace iteration eigensolver now supports the optional parameters ACOUSTIC COUPLING, DAMPING PROJECTION, and RESIDUAL MODES. In addition, you can specify the acoustic range factor on the data line.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FREQUENCY*.

*INITIAL CONDITIONS

You can use the new LONG TERM parameter to specify initial stresses for a viscoelastic material in a fully relaxed state. You can now set the TYPE parameter to MULLINS to specify initial values of the maximum energy density in Mullins effect models.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **INITIAL CONDITIONS*.

*MODAL DYNAMIC

You can use the new SOLVER=FOURIER value to calculate the time domain response as the inverse Fourier transform of the frequency response obtained in the preceding mode- or subspace-based steady-state dynamic step. You can use the new SSD LOAD CASE parameter to obtain the transient response from a specific load case of a previous steady-state dynamics response.

Benefits: These features expand the capabilities of Abaqus/Standard.

For more information, see **MODAL DYNAMIC*.

*PLY FABRIC FAILURE

You can now use this option in Abaqus/Standard to define the conditions that trigger material failure. In earlier releases, this option was available only in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **PLY FABRIC FAILURE*.

*PLY FABRIC HARDENING

You can now use this option in Abaqus/Standard to specify the hardening behavior of the shear plasticity model for bidirectional fabric-reinforced composite materials with or without damage. In earlier releases, this option was available only in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **PLY FABRIC HARDENING*.

*SECTION CONTROLS

In Abaqus/Explicit, you can use the new SPH BULK VISCOSITY parameter to set a uniform bulk viscosity for all SPH particles or to set a vorticity weighted bulk viscosity for each individual SPH particle.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SECTION CONTROLS*.

*STEADY STATE DYNAMICS

You can use the new MODE BASED=FOURIER value to calculate the frequency response as the direct Fourier transform of the transient response obtained in the preceding modal dynamic analysis step.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **STEADY STATE DYNAMICS*.

*SUBMODEL CONDITIONS

You can use the new OP parameter to modify or remove existing submodel conditions.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **SUBMODEL CONDITIONS*.

*SUBMODEL CUT

This keyword option was treated as history data. It is now treated as model data.

Benefits: This feature expands the capabilities of Abaqus/Standard.
The MAIN IMPOSED CONDITIONS parameter and the OP parameter have been removed. In addition, this option no longer takes any data lines.

For more information, see **SUBMODEL CUT*.

*SURFACE PROPERTY

This option can now be used to define surface properties for surface tension loads.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SURFACE PROPERTY*.

*SURFACE TENSION

You can now use this option in Abaqus/Standard to define surface tension properties on an element-based surface. In earlier releases, this option was available only in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **SURFACE TENSION*.

*SURFACE TENSION CONTACT ANGLE

You can now use this option in Abaqus/Standard to specify the surface tension contact angle associated with a surface interaction definition where one of the surfaces has surface tension. In earlier releases, this option was available only in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Standard.
In Abaqus/Explicit, this keyword option has been enhanced and is now implemented as model or history data.

For more information, see **SURFACE TENSION CONTACT ANGLE*.

*VISCOUS SOFTENING (new)

Specify viscous softening of nonlinear viscous networks in materials defined with the parallel rheological framework.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **VISCOUS SOFTENING*.

2026 GA

*ADAPTIVE REMESH (new)

Activate the adaptive remeshing capability and specify an adaptive remesh domain in the model.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ADAPTIVE REMESH*.

*ADAPTIVE REMESH RETAIN (new)

Specify the mesh and/or mesh geometry entities in the adaptive domain that need to be reconstructed after each remeshing operation.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ADAPTIVE REMESH RETAIN*.

*CONTACT CONTROLS ASSIGNMENT

You can configure Abaqus/Standard to internally generate surface elements that represent the circular cross-section of beams and pipes for use in general contact calculations.

Benefits: You can achieve more accurate modeling of beam ends and obtain detailed contact output on the contact surface.

For more information, see **CONTACT CONTROLS ASSIGNMENT*.

*FATIGUE

For fatigue crack growth with adaptive remeshing, you can now specify the ratio of the crack extension length to a typical averaged element characteristic length on the data line.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FATIGUE*.

*FRACTURE CRITERION

In Abaqus/Standard, you can now use this option to specify a fatigue crack growth criterion with adaptive remeshing.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FRACTURE CRITERION*.

*PARTICLE GENERATOR MIXTURE

You can now use this option to specify particle species for elements sets consisting of PC3D elements that are associated with a solid section definition.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **PARTICLE GENERATOR MIXTURE*.

*RADIATION VIEW FACTOR

You can use the new AXISYMMETRIC ACCURACY parameter to control the accuracy of the view factor calculations in an axisymmetric analysis.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **RADIATION VIEW FACTOR*.

*VISCOELASTIC

You can now set the LAW parameter equal to ANAND to specify the Anand creep law for the nonlinear viscoelasticity definition.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **VISCOELASTIC*.

2025 FD04

*COUPLED CONSTITUTIVE RESPONSE

You can use the new PROPERTIES parameter to set the number of real (floating point) property values needed as data in user subroutine `UCOUPLEDCONSTRESP`.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **COUPLED CONSTITUTIVE RESPONSE*.

*INTERNAL FLUX GENERATION

You can use the new PROPERTIES parameter to set the number of real (floating point) property values needed as data in user subroutine `UINTFLUXGEN`.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **INTERNAL FLUX GENERATION*.

*MEAN FIELD HOMOGENIZATION

You can use the new ODF ADJUST parameter to specify if the orientation distribution function (ODF) reconstructed from the orientation tensor is adjusted to remove any negative values.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **MEAN FIELD HOMOGENIZATION*.

*SYMMETRIC RESULTS TRANSFER

You can use the new LIBRARY parameter to specify the name of the previous axisymmetric or partial three-dimensional analysis from which to transfer results to a three-dimensional model.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **SYMMETRIC RESULTS TRANSFER*.

2025 FD03

*COUPLED CONSTITUTIVE RESPONSE (new)

Define a user-defined constitutive behavior across multiple physics fields.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **COUPLED CONSTITUTIVE RESPONSE*.

*CURE KINETICS

You can use the new NUMBER TERMS parameter to define the number of terms in the Kamal model. In addition, you can now define the Kamal model with temperature and/or field variables dependent coefficients.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see *Curing Processes in Polymers* and **CURE KINETICS*.

*DECURRENT

You can use the new `DECURRENT` user subroutine together with table collections to specify electrical current density as a function of position, time, element number, field variables, etc. for coupled thermal-electrical, coupled temperature-displacement electrical, coupled thermal-electrochemical, and coupled temperature-displacement electrochemical analyses.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DECURRENT*.

*DSECURRENT

You can use the new `DECURRENT` user subroutine together with table collections to specify electrical current density as a function of position, time, element number, field variables, etc. for coupled thermal-electrical, coupled temperature-displacement electrical, coupled thermal-electrochemical, and coupled temperature-displacement electrochemical analyses.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DSECURRENT*.

*IMPORT

You can input a blank line as data to specify import of all element sets in the previous analysis with a section definition that is supported for import.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

This new format provides a simple way to import all elements in a model once per previous analysis without the need to list the element sets to be imported explicitly.

You can use the blank line input format with repositioning.

For more information, see **IMPORT*.

*SURFACE

You can now use the PROPERTY parameter to define named surface properties for surface tension loads.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SURFACE*.

*SURFACE PROPERTY

You can now use the PROPERTY parameter to define named surface properties for surface tension loads.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SURFACE PROPERTY*.

*SURFACE TENSION (new)

Specify surface tension properties on element-based surfaces.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SURFACE TENSION*.

*SURFACE TENSION CONTACT ANGLE (new)

Specify a surface tension contact angle as part of a surface interaction definition.

Benefits: This feature expands the capabilities of in Abaqus/Explicit.

For more information, see **SURFACE TENSION CONTACT ANGLE*.

2025 FD02

*DAMPING

You can use the new FREQUENCY DEPENDENT parameter to define frequency-dependent structural damping in Abaqus/Standard, and you can now use the DEPENDENCIES parameter to define variable alpha, beta, band-limited, and structural damping in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.
In addition, the ability to define tabular dependence of multiple damping coefficients in a single damping option was removed to simplify the keyword interface. Instead, you can use multiple **DAMPING* options to define the tabular dependence of individual material damping coefficients independently of each other.

For more information, see **DAMPING* and *Material Damping*.

*ELEMENT RESPONSE

You can use the new FATIGUE parameter to specify fatigue design responses for adjoint fatigue design sensitivity analysis using a combined Abaqus-fe-safe workflow.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ELEMENT RESPONSE* and *Adjoint Design Sensitivity Analysis Overview*.

*INITIAL CONDITIONS

For any of the coupled electrochemical analyses, you can now set the TYPE parameter to BINDERVF and SOLIDVF to give the initial volume fraction of the binder and solid phases, respectively and TYPE=PARTICLE SPECIES CONCENTRATION to give initial species concentrations in the particles present in the active electrodes.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **INITIAL CONDITIONS*.

*MATERIAL

You can use the new USER TYPE parameter to define a material using one of a set of special-purpose user material types.

Benefits: You can now use any name for these special-purpose user materials when the appropriate USER TYPE is specified.

The new USER TYPE parameter can be used to model metallurgical phase transformations and to model cardiac tissue in the Living Heart Human Model. In earlier releases, you had to use a material name starting with a specific “ABQ_” prefix.

For more information, see **MATERIAL* and *About Special-Purpose Material Modeling Techniques*.

*MULLINS EFFECT

You can now use the new DEFINITION parameter to specify the exponential and the slow decay forms to define the evolution of the damage variable for the Mullins effect model. In addition, you must now use the DEFINITION parameter to specify the Mullins effect models, which were available in the earlier releases.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **MULLINS EFFECT*.

*SRADIATE

You can use the new TRANSITION parameter to control the emissivity dependence on contact status.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SRADIATE*.

2025 FD01

*CLUSTER MASS INERTIA TABLE (new)

Specify mass and inertia for discrete particle clusters.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **CLUSTER MASS INERTIA TABLE*.

*CONTACT CONTROLS ASSIGNMENT

You can use the new SHELL REPRESENTATION parameter to specify how to represent shells for general contact in Abaqus/Standard. In addition, the default value of the BEAM CROSS SECTION parameter is now EXACT in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTACT CONTROLS ASSIGNMENT*.

*CONTACT FORMULATION

You now set the TYPE parameter equal to MAIN SECONDARY ROLES to control main-secondary roles for specific interactions in Abaqus/Explicit. In earlier releases, this value was supported only in Abaqus/Standard.

Benefits: This improvement unifies the user interface between Abaqus/Explicit and Abaqus/Standard.

For more information, see **CONTACT FORMULATION*.

*CONTACT INITIALIZATION DATA

For general contact in Abaqus/Standard, you can use ADJUST=NO for interactions that are permanently bonded with cohesive contact.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTACT INITIALIZATION DATA*.

*DIAGNOSTICS

For Abaqus/Explicit analyses, the default value of the CONTACT INITIAL OVERCLOSURE parameter is now SUMMARY. In earlier releases, the default value was DETAIL.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DIAGNOSTICS*.

*DISTRIBUTING

You can now use the ROTATIONAL COUPLING parameter in both Abaqus/Standard and Abaqus/Explicit to specify whether the rotational constraint of the reference node should include the rotations of the cloud nodes in addition to the displacements of the cloud nodes. The parameter can take the new default value of AUTOMATIC to allow Abaqus to couple the rotation of the reference node to the displacement and the rotation of the cloud nodes within the influence radius.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **DISTRIBUTING*.

*ENERGY OUTPUT

You can now use the NSET parameter with direct-solution steady-state dynamic analyses. Alternatively, you can use the new SURFACE parameter to specify the name of the element-based surface representing the boundary for energy flow, power flow, radiated acoustic energy, and radiated acoustic power computations.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ENERGY OUTPUT*.

*EQUIVALENT RADIATED SURFACE PROPERTIES (new)

Specify radiated surface properties.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **EQUIVALENT RADIATED SURFACE PROPERTIES*.

*FASTENER

You can now use the ROTATIONAL COUPLING parameter in both Abaqus/Standard and Abaqus/Explicit to specify whether the rotational constraint of the fastening point should include the rotations of the surface nodes in addition to the displacements of the surface nodes. The parameter can take the new default value of AUTOMATIC to allow Abaqus to couple the rotation of the fastening point to the displacement and the rotation of the surface nodes within the influence radius.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **FASTENER*.

*FRACTURE CRITERION

You can set the TYPE parameter equal to PRX-FATIGUE to use the PRX-Fatigue criterion as the fatigue crack propagation criterion.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **FRACTURE CRITERION*.

*HYPERELASTIC

You can now use the HENCKY, PROPERTIES, USER, and VALANIS-LANDEL parameters in both Abaqus/Standard and Abaqus/Explicit. In earlier releases, you could use these parameters only in Abaqus/Standard.

Benefits: This improvement unifies the user interface between Abaqus/Explicit and Abaqus/Standard.

For more information, see **HYPERELASTIC*.

*NORMAL

You can use the new BEAM NORMAL parameter to define nodal normals.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **NORMAL*.

*PAPERBOARD HARDENING (new)

Specify the hardening data of the in-plane paperboard plasticity model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **PAPERBOARD HARDENING*.

*PAPERBOARD PLASTICITY (new)

Specify the in-plane paperboard plasticity model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **PAPERBOARD PLASTICITY*.

*PAPERBOARD THICKNESS COMPRESSION ELASTIC (new)

Define the out-of-plane compression elasticity for the paperboard model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **PAPERBOARD THICKNESS COMPRESSION ELASTIC*.

*PAPERBOARD THICKNESS COMPRESSION PLASTICITY (new)

Define the out-of-plane compression yield surface for the paperboard model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **PAPERBOARD THICKNESS COMPRESSION PLASTICITY*.

*PAPERBOARD TRANSVERSE SHEAR PLASTICITY (new)

Define the transverse shear yield surface for the paperboard model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **PAPERBOARD TRANSVERSE SHEAR PLASTICITY*.

*SFILM

You can use the new TRANSITION parameter to control the film coefficient's dependence on the contact status.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SFILM*.

*STEP CONTROL

You can use the new TOLERANCE parameter to determine when the sensor has reached the threshold value.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **STEP CONTROL*.

*SURFACE BEHAVIOR

When used to modify the default surface behavior, you cannot use the PRESSURE-OVERCLOSURE parameter with the NO SEPARATION parameter in a general contact Abaqus/Explicit analysis, except for PRESSURE-OVERCLOSURE=LINEAR.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SURFACE BEHAVIOR*.

*SURFACE SECTION

You can now use this option to define a constant thickness surface element (independent of using the *REBAR LAYER option).

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SURFACE SECTION*.

PDF Guides Available

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

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

For more information, see *Abaqus Keywords Guide*.

2025 GA

*STEP CYCLING (new)

Specify a step or series of steps to be repeated over a set number of cycles.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **STEP CYCLING*.

*STEP CYCLING CONTROL (new)

Specify a sensor-based or a contact wear-based criterion to trigger early termination of step cycling before a set number of step cycles are complete.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **STEP CYCLING CONTROL*.

2024 FD04

*BEAM GENERAL SECTION

Use the new optional CONTROLS parameter to specify bulk viscosity scale factors or to ramp up the initial stress in an Abaqus/Explicit analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **BEAM GENERAL SECTION*.

*BEAM SECTION

Use the new optional CONTROLS parameter to specify bulk viscosity scale factors or to ramp up the initial stress in an Abaqus/Explicit analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **BEAM SECTION*.

2024 FD03

*CONTROLS

You can use the new CONVERGENCE CHECK parameter to specify which convergence conditions Abaqus/Standard must satisfy before it accepts the incremental solution as converged.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTROLS*.

*ELECTRICAL CONDUCTIVITY

You can now use this option to define electrical conductivity for elements that support the electric potential degree of freedom in a coupled thermal-electrochemical, coupled thermal-electrochemical-structural, or coupled thermal-electrochemical-structural-pore pressure analysis.

Benefits: This feature expands the capabilities of Abaqus/Standard.
In earlier releases, you could use **ELECTRICAL CONDUCTIVITY* only in coupled thermal-electrical and coupled thermal-electrical-structural analyses.

For more information, see **ELECTRICAL CONDUCTIVITY*.

*ELECTRICAL RESISTIVITY (new)

Define electrical resistivity for elements that support the electric potential degree of freedom in a coupled thermal-electrical, coupled thermal-electrical-structural, coupled thermal-electrochemical, coupled thermal-electrochemical-structural, or coupled thermal-electrochemical-structural-pore pressure analysis.

Benefits: This feature expands the capabilities of Abaqus/Standard.
You can use the new option to define electrical resistivity (as opposed to conductivity). This method of specifying electrical resistivity is required for piezoresistive materials.

For more information, see **ELECTRICAL RESISTIVITY* and *Piezoresistive Effect*.

*ELECTRIC MACHINE LOAD (new)

Define electromagnetic force loads for an electric machine.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ELECTRIC MACHINE LOAD*.

*ELECTRIC MACHINE PROPERTY (new)

Define electric machine properties (or parameters) that are needed to apply electric machine loads.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ELECTRIC MACHINE PROPERTY*.

*ELEMENT USER OUTPUT VARIABLES (new)

Specify the number of whole element and integration point user-defined variables defined for each element in user subroutine `UVAREL`.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ELEMENT USER OUTPUT VARIABLES*.

*EXPANSION

You can use the new DEFINITION parameter to specify secant or tangent thermal expansion coefficients.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **EXPANSION*.

*MODAL DYNAMIC

You can use the new SOLVER parameter to specify if you want to use the state space solver or the conventional solver.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **MODAL DYNAMIC*.

*PIEZORESISTIVITY (new)

Define the piezoresistivity coefficients for a piezoresistive material.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **PIEZORESISTIVITY*.

*SECTION CONTROLS

You can now use this option to control the activation of bending stabilization for full-integration linear elements modeled with hyperelastic materials and to specify the scale factor.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SECTION CONTROLS*.

*UNIT SYSTEM

You can use the new SYSTEM parameter to specify the unit system used in the model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **UNIT SYSTEM*.

*VISCOELASTIC

You now set the LAW parameter equal to MODIFIED DARVEAUX to choose a modified Darveaux law for a nonlinear viscoelastic material definition.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **VISCOELASTIC*.

2024 FD02

*CONTOUR INTEGRAL

You can use the LENGTH SCALING FACTOR parameter to indicate that Abaqus/Standard will not adjust the locations or the radii of all nodes along the crack front when evaluating the line integral.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTOUR INTEGRAL*.

*CO-SIMULATION

The CONTROLS parameter and the PROGRAM=ABAQUS value are deprecated. You can now set PROGRAM=MULTIPHYSICS for exchange of data between Abaqus and the SIMULIA Co-Simulation Services, which in turn can exchange data with third-party analysis programs.

Benefits: All co-simulation control parameters are defined in the Co-Simulation Engine Configuration File.

For more information, see **CO-SIMULATION*.

*CO-SIMULATION CONTROLS (removed)

This option is removed. You now define all co-simulation control parameters in the Co-Simulation Engine Configuration File.

Benefits: Defining a co-simulation process is easier.

For more information, see **CO-SIMULATION*.

*CURE GLASS TRANSITION TEMPERATURE

You can now use this option with all cure kinetics models. In earlier releases, you could use it only with the Grindling model.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CURE GLASS TRANSITION TEMPERATURE*.

*DENSITY

Abaqus/Explicit now supports the PORE FLUID parameter for undrained pore fluid flow and stress analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DENSITY*.

*DYNAMIC

You can now set the PORE PRESSURE parameter equal to UNDRAINED to specify that an undrained pore fluid flow and stress analysis is to be performed in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DYNAMIC*.

*EXPANSION

Abaqus/Explicit now supports the PORE FLUID parameter for undrained pore fluid flow and stress analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **EXPANSION*.

*HYPERELASTIC

You can use the new HENCKY parameter to use the Hencky model in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **HYPERELASTIC*.

*IMPORT

You can use the full-model input format to specify that all import model data, including elements and nodes, element sets and node sets, sections and materials, etc., are given in the input.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **IMPORT*.

*IMPORT CONTROLS

You can set the new MODEL parameter equal to YES to indicate that you want to use the full-model input format. In an Abaqus/Explicit analysis, you can use the new TIE CONFIG parameter to indicate whether to use the original or updated configuration in the tie constraint formation for all surfaces of the imported model.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **IMPORT CONTROLS*.

*INITIAL CONDITIONS

You can now set the TYPE parameter equal to PORE PRESSURE to define the initial pore pressure for undrained pore fluid flow and stress analysis in Abaqus/Explicit. Similarly, you can set the TYPE parameter equal to POROSITY to define the initial porosity values and to RATIO to define the initial void ratio values.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **INITIAL CONDITIONS*.

*POROUS BULK MODULI

You can now use this option in Abaqus/Explicit to define the porous bulk moduli for undrained pore fluid flow and stress analysis.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **POROUS BULK MODULI*.

*SURFACE PROPERTY

You can now use this option in Abaqus/Explicit to define named surface properties for contact-based wear modeling.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **SURFACE PROPERTY*.

*VISCOELASTIC

You now set the LAW parameter equal to DARVEAUX to choose a Darveaux law.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **VISCOELASTIC*.

*WEAR SURFACE PROPERTIES

You can use this option in Abaqus/Explicit to specify wear properties to activate wear distance accumulation at contact surface nodes.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **WEAR SURFACE PROPERTIES*.

2024 FD01

*ALLOWABLE STRESS (new)

Specify the allowable stress of a composite at the initiation of damage in a multiscale material.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **ALLOWABLE STRESS*.

*CFILM

You can use the new SINK AMPLITUDE parameter to define the variation of the sink temperature with time. The SINK AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier versions.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **CFILM* and *Thermal Loads*.

*CONNECTOR SECTION

For SLIPRING connections, you can now request that Abaqus/Standard computes the contact angle automatically during the analysis.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONNECTOR SECTION*.

*CONTOUR INTEGRAL

You can now use the SURFACE NORMAL parameter when tetrahedral elements are used for fracture mechanics studies based on the conventional finite element method.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **CONTOUR INTEGRAL*.

*CRADIATE

You can use the new AMBIENT AMPLITUDE parameter to define the variation of the ambient temperature with time. The AMBIENT AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier versions. You can also use the new RADIATION AMPLITUDE parameter to define a variation of the radiation flux with time.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **CRADIATE* and *Thermal Loads*.

*DAMAGE EVOLUTION

You can now set the TYPE parameter equal to DISCONTINUOUS to define the residual stiffness scaling factor after the initiation of damage.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **DAMAGE EVOLUTION*.

*DAMAGE INITIATION

You can now define general stress/strain-based damage initiation as a material property for continuum damage modeling in anisotropic materials. You can also now define damage initiation as a material property for continuum damage modeling in fiber-reinforced composites using the multiscale material model in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **DAMAGE INITIATION*.

*DAMPING

Abaqus/Standard now supports the BAND LIMITED, HIGH FREQUENCY CUTOFF, and LOW FREQUENCY CUTOFF parameters.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **DAMPING*.

*DISTRIBUTING

You can use the new SINGLE FACET parameter to specify whether or not to automatically convert distributing couplings where all cloud nodes are on a single shell facet to a constraint in Abaqus/Explicit.

Benefits: This feature expands the capabilities of Abaqus/Explicit.

For more information, see **DISTRIBUTING*.

*EXPANSION

In an Abaqus/Standard analysis, you can now use the PORE FLUID parameter to define field expansion of the pore fluid in a porous medium. Similarly, the USER and ZERO parameters now take field expansion into account.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **EXPANSION*.

*FILM

You can use the new SINK AMPLITUDE parameter to define the variation of the sink temperature with time. The SINK AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier versions.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **FILM* and *Thermal Loads*.

*NSET

You can now use the new SURFACE parameter to generate a node set from a previously defined surface.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.
This enhancement makes it easier to create a node set from a surface definition. All the nodes included in the surface are assigned to the specified node set.

For more information, see **NSET*.

*RADIATE

You can use the new AMBIENT AMPLITUDE parameter to define the variation of the ambient temperature with time. The AMBIENT AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier

versions. You can also use the new RADIATION AMPLITUDE parameter to define a variation of the radiation flux with time.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **RADIATE* and *Thermal Loads*.

*REDUCED BASIS GENERATE (new)

Indicate that the step should be analyzed as a reduced basis generation step.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **REDUCED BASIS GENERATE*.

*SFILM

You can use the new SINK AMPLITUDE parameter to define the variation of the sink temperature with time. The SINK AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier versions.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SFILM* and *Thermal Loads*.

*SRADIATE

You can use the new AMBIENT AMPLITUDE parameter to define the variation of the ambient temperature with time. The AMBIENT AMPLITUDE parameter supersedes the AMPLITUDE parameter used in earlier versions. You can also use the new RADIATION AMPLITUDE parameter to define a variation of the radiation flux with time.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see **SRADIATE* and *Thermal Loads*.

*STEADY STATE DYNAMICS

You can use the new MODE BASED parameter to define the conventional modal solver or the spectral modal solver that can perform better than the conventional solver in analyses with many modes, frequency points, and load cases for finite element models with special, but commonly used, damping properties.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **STEADY STATE DYNAMICS* and *Mode-Based Steady-State Dynamic Analysis*.

*SUBMODEL CUT (new)

Identify a submodel cut on which to impose the submodel conditions and specify the submodel control settings specific to a submodel cut.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **SUBMODEL CUT*.

*SURFACE PROPERTY

You can now use this option to define named surface properties for contact-based wear modeling.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **SURFACE PROPERTY*.

*WEAR SURFACE PROPERTIES (new)

Specify wear properties to activate wear distance accumulation at contact surface nodes.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **WEAR SURFACE PROPERTIES*.

2024 GA

*NODE OUTPUT

You can use the new SURFACE parameter to specify the name of the surface for which to request nodal output.

Benefits: This feature makes it easier to request nodal output on a surface in Abaqus/Standard and Abaqus/Explicit. The SURFACE parameter is available when the *NODE OUTPUT option is used in conjunction with the *OUTPUT, HISTORY option for Abaqus/Standard and Abaqus/Explicit or when the *NODE OUTPUT option is used in conjunction with the *OUTPUT, FIELD option for Abaqus/Explicit.

For more information, see **NODE OUTPUT* and **OUTPUT*.

*NODE RESPONSE

You can use the new RANGE OPERATOR parameter to define the operator used for aggregating the sensitivities and design response values over the specified frequency range for the node responses in direct steady-state dynamic adjoint sensitivity analysis.

Benefits: This parameter allows you to control the calculation of the design response over a frequency range for the newly supported steady-state dynamic adjoint sensitivity analysis technique.
By default, Abaqus/Standard outputs the maximum of the design response values as the design response over the frequency range specified for the node response in direct steady-state dynamic analysis when you request adjoint sensitivity computation. Alternatively, you can request output of the sum or average of the node response values over the specified frequency range for the response.

For more information, see **NODE RESPONSE* and *Adjoint Design Sensitivity Analysis Overview*.

*OPERATOR OUTPUT

You can now use the *OPERATOR OUTPUT option in an uncoupled heat transfer analysis. In addition, the LOAD parameter can now take the value NET to output the net flux vector in an uncoupled heat transfer analysis.

Benefits: This feature expands the capabilities of Abaqus/Standard.

For more information, see **OPERATOR OUTPUT*.

Materials

This page describes recent changes in Abaqus Materials.

2026 FD04

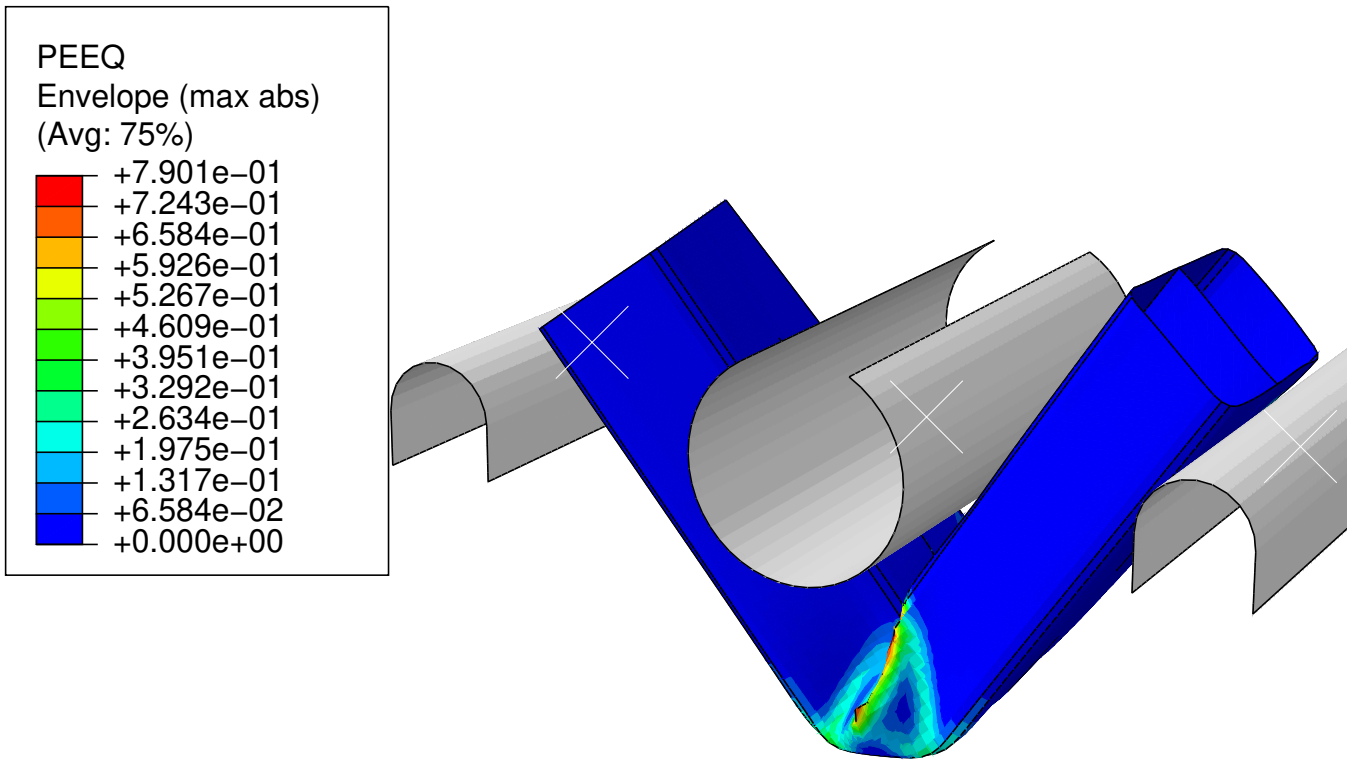
Modeling Nonquadratic Yield Using Plane Stress, Shell, and Membrane Elements

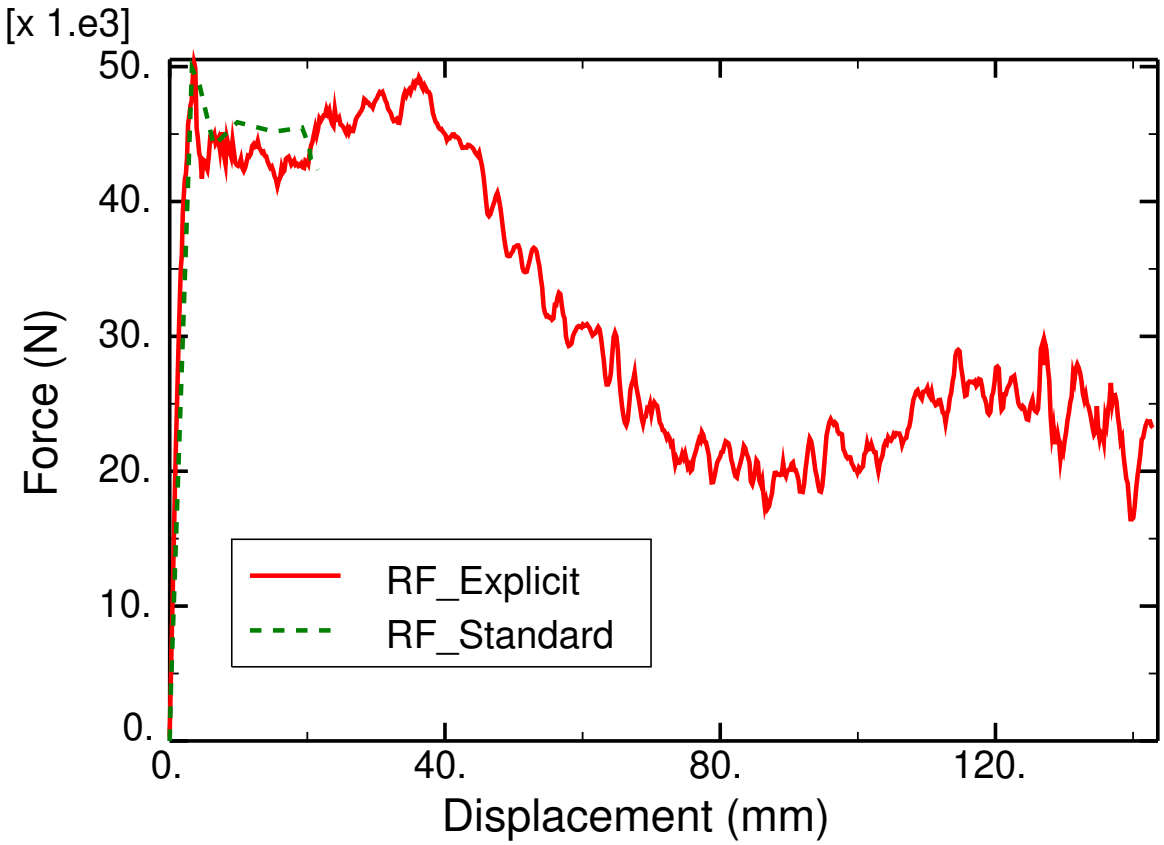
You can now model nonquadratic yield using plane stress, shell, and membrane elements in Abaqus/Standard and Abaqus/Explicit. You can also use a new yield criterion (Barlat Yld2000-2d) to model sheet metal forming.

Benefits: Compared with Hill anisotropic plasticity, nonquadratic yield criteria such as Barlat anisotropic plasticity models provide flexibility for describing more complex anisotropic yield behavior. The new Barlat Yld2000-2d yield surface provides a more accurate prediction of the forming behavior of sheet metals.

In earlier releases, nonquadratic yield was available only for three-dimensional continuum elements and plane strain elements in Abaqus/Explicit. You can now use this functionality for plane stress, shell, and membrane elements as well, and it is available in both Abaqus/Standard and Abaqus/Explicit.

The example below shows the progressive failure analysis of a thin-wall aluminum extrusion under quasi-static and dynamic loads. The yield behavior is modeled with the Barlat Yld2000-2d model. The analysis is performed with both Abaqus/Standard and Abaqus/Explicit.


PEEQ
Envelope (max abs)
(Avg: 75%)
+7.901e−01
+7.243e−01
+6.584e−01
+5.926e−01
+5.267e−01
+4.609e−01
+3.951e−01
+3.292e−01
+2.634e−01
+1.975e−01
+1.317e−01
+6.584e−02
+0.000e+00



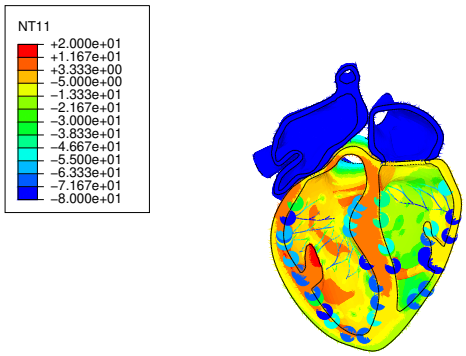
For more information, see *Nonquadratic Yield*.

Modeling the Electrical and Mechanical Behaviors of Cardiac Tissue

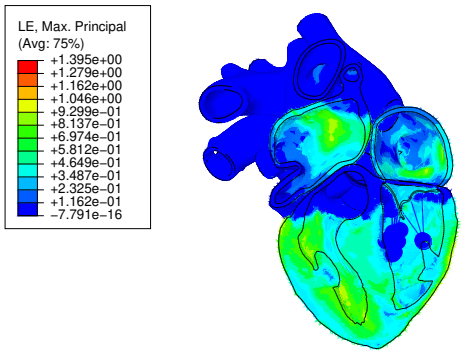
You can now model the electrical behavior of cardiac tissue in Abaqus/Standard and its mechanical behavior in Abaqus/Explicit.

Benefits: You can use these material models to simulate the electrical and mechanical responses of cardiac tissue.

Abaqus/Standard now provides a special-purpose material model for the electrical behavior of cardiac tissue. This model relies on a built-in implementation of user subroutine `HETVAL` and can be used to simulate the electrical response of cardiac tissue.



Abaqus/Explicit now provides a special-purpose material model for the mechanical behavior of cardiac tissue. This model relies on a built-in implementation of user subroutine `VUANISOHYPER_INV` and can be used to simulate the mechanical response of cardiac tissue.



For more information, see *Modeling the Electrical Response of Cardiac Tissue* and *Modeling the Mechanical Response of Cardiac Tissue*.

2026 FD03

Modeling Multiscale Materials with Deep Material Networks (DMNs)

You can now model composite materials using deep material networks (DMNs) in Abaqus/Standard.

Benefits: DMNs can be used for multiscale material modeling of heterogeneous materials such as composites.

Deep material network (DMN) has recently emerged as a data-driven surrogate model for heterogeneous materials. DMN captures the microstructural morphology by analyzing the relationship between the constituent properties and the homogenized effective stiffness tensor. The coefficients of DMN are computed by minimizing the difference between the DMN and high-fidelity FE-RVE predictions, as shown in the figure below.



DMN is a physics-informed model built on building blocks with analytical homogenization solutions; therefore, it inherently satisfies conservation laws and thermodynamic consistency. DMN acts as a high-efficiency reduced-order model, providing a faster alternative to computationally expensive FE-RVE simulations.

For more information, see *Modeling Multiscale Materials with Deep Material Networks (DMNs)*.

2026 FD02

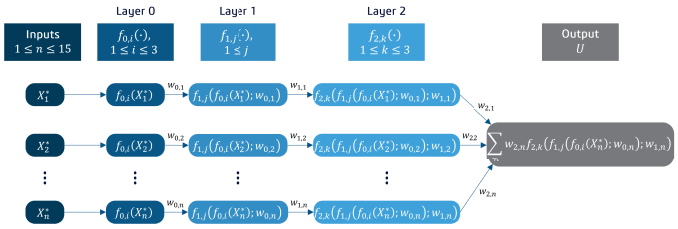
Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs) Models

You can now model soft matter using constitutive artificial neural networks (CANNs) models both in Abaqus/Standard and Abaqus/Explicit.

Benefits: CANNs models enable you to model many biological tissue and soft material systems.

Constitutive artificial neural networks (CANNs) models are a general class of anisotropic hyperelastic constitutive models that are described by a neural network architecture. They can be used to model the response of biological tissue and other soft material system.

Abaqus now provides a general framework to integrate trained CANNs models into an Abaqus simulation with minimal user effort. This framework is available as a special-purpose material that relies on a built-in implementation of user subroutines `UANISOHYPER_INV` and `VUANISOHYPER_INV`.



For more information, see *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*.

2026 FD01

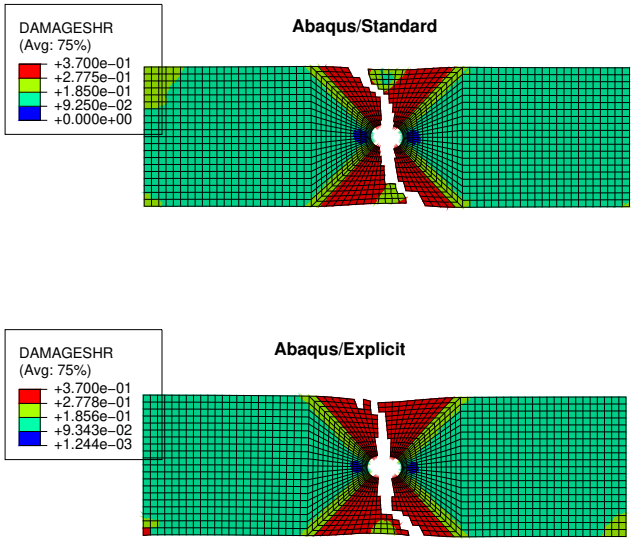
Ply Fabric Material Model in Abaqus/Standard

You can now use the ply fabric material to model bidirectional fabric-reinforced composite materials in Abaqus/Standard, and you can import results of a bidirectional fabric-reinforced composite material from Abaqus/Standard to Abaqus/Explicit, and vice versa.

Benefits: This feature expands the capabilities of Abaqus/Standard.

In earlier releases, the capabilities to model bidirectional fabric-reinforced composite materials were available only in Abaqus/Explicit. You can now define the bilamina elasticity, shear plasticity, ply fabric damage initiation and evolution, and ply fabric failure for a bidirectional fabric-reinforced composite material in a static analysis.

The example shown below is the tension test of a plate with an open hole.



Bidirectional fabric-reinforced composite material is used in the plane. The off-axis angle of the material is 45 degrees. The mesh uses membrane elements, and Abaqus deletes elements when the principal stresses in the elements exceeded the maximum value of 5%. The figure shows the comparison of shear damage in the composite plate. The results given by Abaqus/Standard and Abaqus/Explicit show good agreement.

For more information, see *Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials*.

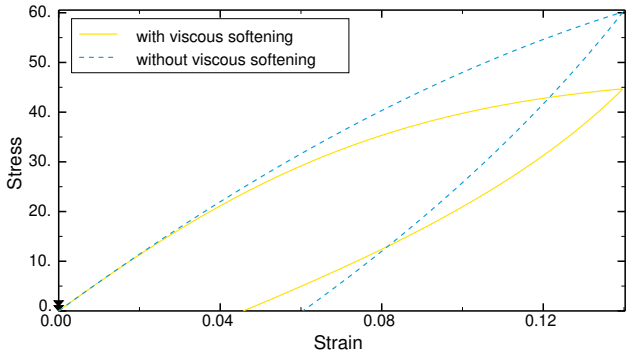
Viscous Softening

You can now model the viscous softening of the nonlinear viscoelastic networks of models defined using the parallel rheological framework in both Abaqus/Standard and Abaqus/Explicit.

Benefits: You can now model many thermoplastic materials and other materials that exhibit this behavior.

Some nonlinear viscoelastic materials undergo viscous softening; that is, their deviatoric stiffness softens with the evolution of the equivalent creep strain. Each nonlinear viscoelastic network within the parallel rheological framework can undergo viscous softening according to the equivalent creep strain it experiences.

To illustrate the effect of viscous softening, the following example considers the uniaxial loading and unloading of a nonlinear viscoelastic material, with and without viscous softening.



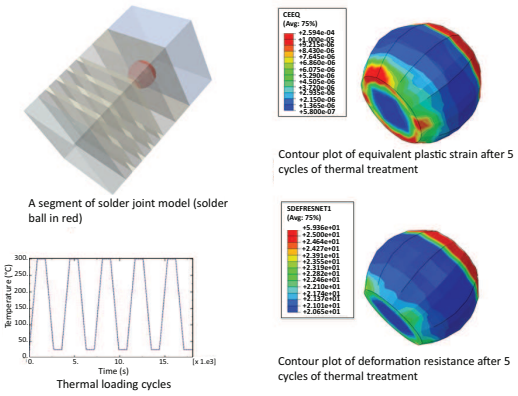
For more information, see *Parallel Rheological Framework*.

2026 GA

Anand Creep Model in Parallel Rheological Framework

You can now use the Anand model to define creep response in the parallel rheological framework (PRF).

Benefits: You can better simulate the creep behavior of solder using the Anand creep model in PRF materials. The Anand creep law is particularly well suited for modeling solder materials used in electronic packages by capturing both the primary and secondary (steady-state) creep along with the temperature and rate dependence. Anand creep is available in both Abaqus/Standard and Abaqus/Explicit.
The figure below is from an example of a solder ball under cyclic thermal loading, a test to evaluate the reliability of solder joints. The creep strain in the solder ball is predicted using the Anand model.



For more information, see *Parallel Rheological Framework*.

2025 FD03

Enhancement to Ductile Damage Initiation

You can now compute ductile damage initiation criteria for hyperelastic materials with isotropic plasticity.

Benefits: This feature expands the damage modeling capabilities of Abaqus/Standard.

In earlier releases, the ductile damage initiation criteria were available in Abaqus/Standard only for material models with elastic behavior defined using a linear elastic material. Now Abaqus/Standard can compute the ductile damage initiation criteria for hyperelastic materials used with isotropic plasticity (such as the model to predict permanent set and models defined using the parallel rheological framework that includes plasticity).

The new feature can be particularly useful for preloading models in Abaqus/Standard with loads that do not initiate damage evolution and then importing them into Abaqus/Explicit where the analysis continues and material degradation takes place. In many cases preloading can be performed more efficiently in Abaqus/Standard than in Abaqus/Explicit; therefore, this approach can substantially reduce the overall computation time.

For more information, see *Damage Initiation for Ductile Metals*.

Defining Temperature- and Field Variable–Dependent Coefficients for the Kamal Cure Model

You can now define coefficients that are functions of temperature and/or field variables when using the Kamal model.

Benefits: This feature expands the cure modeling capabilities of Abaqus/Standard.
The Kamal model is a widely used model for accurately describing accurately the evolution of the degree of cure. In earlier releases, Abaqus/Standard required the coefficients of the Kamal model to be constant. However, there are situations when other quantities (for example, ultraviolet radiation) can influence the curing process. You can now take such factors into account by making the Kamal model coefficients functions of field variables (for example, representing the ultraviolet radiation).

For more information, see *Curing Processes in Polymers*.

Plane Stress Orthotropic Failure Measures

You can use a new stress-based failure measure based on the YAMADA-SUN theory.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

For more information, see *Yamada-Sun Theory*.

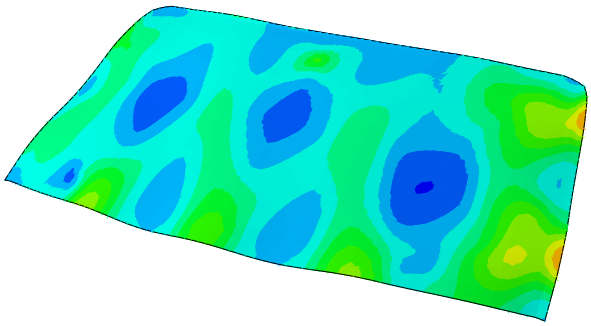
2025 FD02

Frequency-Dependent Structural Damping

You can now model frequency-dependent structural damping in Abaqus/Standard.

Benefits: You can model structural damping more accurately. In earlier releases, structural damping definitions could not depend on frequency.
Structural damping is used primarily to model material damping forces associated with frictional effects in frequency-domain dynamic procedures. You can use this new functionality to describe the frequency-dependent frictional response of materials in structural components (such as brakes and door seals) and when modeling adhesives (such as in windshields).

To illustrate this new functionality, the animation below shows a steady-state dynamic analysis of a windshield. The material definition includes frequency-dependent structural damping to model the damping forces resulting from frictional effects in the windshield.



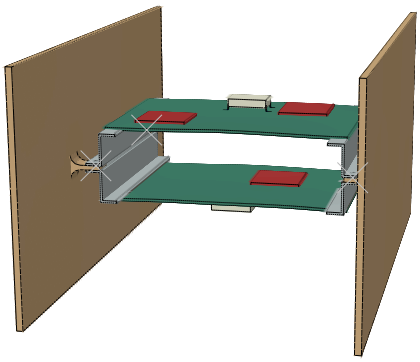
For more information, see *Defining Variable Frequency-Dependent Structural Damping in Abaqus/Standard* and *Structural damping*.

Field Variable Dependence of Alpha, Beta, Band-Limited, and Structural Damping

Alpha damping, Beta damping, band-limited damping, and structural damping that depend on field variables are now supported in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.
Possible applications of the new functionality include:

- Dependence on a real field generated by a previous simulation
- Dependence on an artificial field that you define to modify certain material properties during the course of an analysis
- Spatial dependence by assigning different field variable values to different nodes in the model



For more information, see *Material Damping* and *Predefined Field Variables*.

Defining Material Softening Using the Mullins Effect Model

You can now define material softening by specifying the exponential and the slow decay forms for evolution of the damage variable in the Mullins effect model.

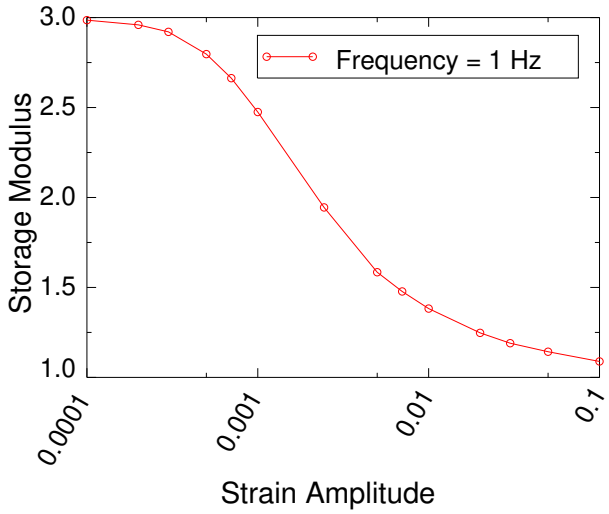
Benefits: You can now model amplitude-dependent material softening in rubber under dynamic loading accurately.

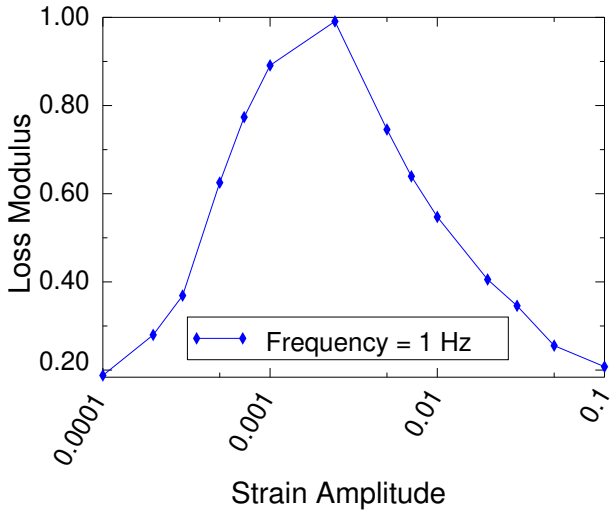
In earlier releases, Abaqus offered the Ogden-Roxburgh model to predict the softening in rubber materials subjected to quasi-static cyclic loading. This phenomenon is known as the Mullins effect and is characterized by irreversible softening of the stress-strain response that occurs as the material is stretched beyond its previous maximum stretch.

Two new models for modeling softening in rubber, which can be more suitable for analyzing material damage as well as amplitude-dependent response, are now available. The new models use an approach based on continuum damage mechanics, with the damaged stress-strain curve deviating from the (undamaged) hyperelastic response specified for the material even under monotonic loading conditions. This is different from the approach used by the Ogden-Roxburgh model, where the material always follows the specified hyperelastic response under monotonic loading.

Abaqus supports two forms of the evolution of the damage variables: the exponential form and the slow decay form. An example use case for these new models is modeling amplitude-dependent material softening, which is commonly observed in filled rubber subjected to cyclic loading. Modeling the amplitude-dependent softening captures the decrease of the storage modulus with increased dynamic strain, a phenomenon that is known as the Payne or Fletcher-Gent effect.

To illustrate the ability to predict the strain dependence of the storage modulus, we consider a specimen subjected to cyclic loading at different values of strain amplitudes. The material behavior is specified using the parallel rheological framework with neo-Hookean hyperelastic material, Bergstrom-Boyce nonlinear creep law, and Mullins effect with the slow decay form of evolution of the damage variable. An amplitude sweep is performed to verify the behavior of the model, and the results are presented below. The storage modulus decreases with increasing amplitude, whereas the loss modulus increases with increasing amplitude initially until it reaches a maximum value beyond which it decreases with increasing amplitude. These results agree qualitatively with the results observed experimentally.





For more information, see *Stress Softening in Rubberlike Materials*.

Using Special-Purpose Material Modeling Techniques

You can use material user types to invoke some special-purpose material modeling techniques.

Benefits: You can now use any name for these special-purpose user materials when the appropriate material user type is specified.
In earlier releases, you had to use a material name starting with a specific “ABQ_” prefix to identify these special-purpose material modeling techniques.

For more information, see **MATERIAL* and *About Special-Purpose Material Modeling Techniques*.

2025 FD01

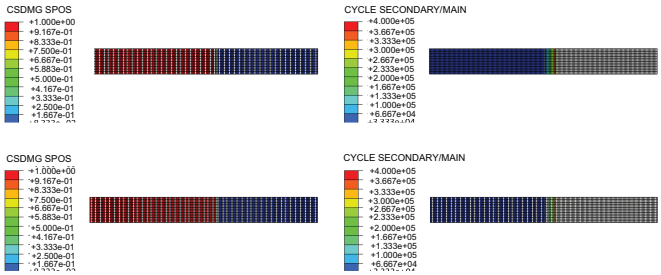
PRX-Fatigue Criterion

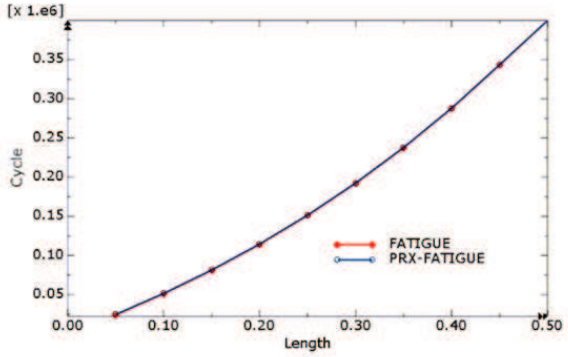
You can use the PRX-Fatigue criterion to model delamination propagation under cyclic loading in Abaqus/Standard.

Benefits: The PRX-Fatigue criterion offers smoother delamination fronts and faster solution times due to its better convergence characteristics.

The Progressive Release eXplicit Virtual Crack Closure Technique (PRX-VCCT) criterion has been extended to support delamination propagation under cyclic fatigue loading. PRX-Fatigue is available only in Abaqus/Standard. Similar to the PRX-VCCT criterion, it incorporates a progressive nodal release strategy within a conventional VCCT framework as well as an "explicit" computation of the delamination increment based on energy release rates of the previous increment.

The following figures show results from a linear elastic fatigue crack growth simulation in a single leg bend specimen, where the fracture energy release rates are computed using either the traditional-VCCT technique (top) or the PRX-VCCT technique (bottom). The results obtained by using the two different approaches agree very well with one another.





For more information, see *Crack Propagation Analysis*, *Fatigue Crack Growth Laws for Linear Elastic Materials*, **FRACTURE CRITERION*, and *Debonding behavior of a single leg bending specimen*.

Hencky Hyperelastic Model in Abaqus/Explicit

You can now use the Hencky hyperelastic model to define hyperelastic material response in Abaqus/Explicit.

Benefits: This enhancement extends the functionality of Abaqus/Explicit.
The Hencky hyperelasticity model was first introduced in Abaqus/Standard in 2024 FD02 (see *Hencky Hyperelastic Model in Abaqus/Standard*). The model is now supported in Abaqus/Explicit, including results transfer between Abaqus/Standard and Abaqus/Explicit.

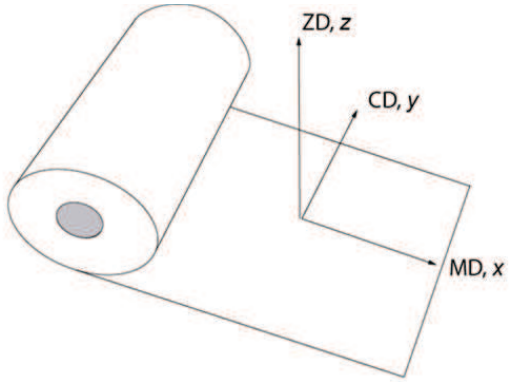
For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

Paperboard Material Model

You can now model paperboard material in Abaqus/Explicit and Abaqus/Standard.

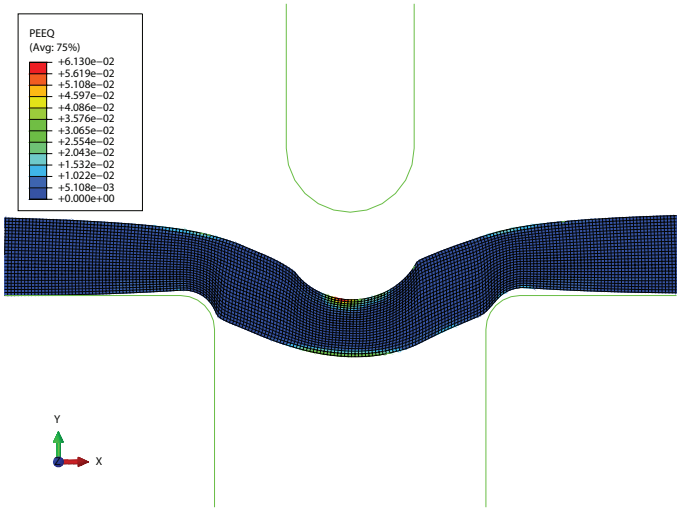
Benefits: You can now simulate workflows in industries that use paperboard material widely, such as in the packaging industry.

A paperboard material is highly orthotropic with local directions as shown below.



The in-plane behavior and the out-of-plane behavior can be considered to be decoupled. The out-of-plane behavior includes both the compression in the thickness direction and the out-of-plane shear. Plastic behavior is modeled both in the in-plane model and the out-of-plane model.

The image below shows the plastic strain results of the analysis modeling the creasing of paperboard.



For more information, see *Paperboard Material Model*.

PDF Guides Available

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

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

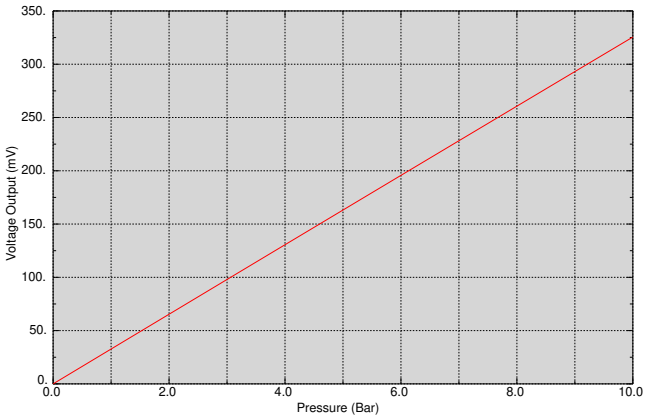
For more information, see *Abaqus Materials Guide*.

2024 FD03

Piezoresistive Effect

You can now model materials exhibiting piezoresistive behavior in Abaqus/Standard.

Benefits: You can now analyze and design piezoresistive sensors such as silicon piezoresistive pressure sensors. A piezoresistive material changes its electrical resistivity when stressed. This coupling between stress and electrical resistivity is the material's piezoresistivity property. Silicon piezoresistive sensors have been widely used in the field of microelectromechanical systems. The figure below shows the simulated results between pressure and voltage output of a silicon piezoresistive pressure sensor.



For more information, see *Piezoresistive Effect*.

Electrical Resistivity

You can now define electrical resistivity directly in Abaqus/Standard.

Benefits: This provides an easier and more robust way to define electrical resistivity.
In earlier releases, you could define only electrical conductivity in coupled thermal-electrical elements and coupled thermal-electrical-structural elements. Calculating electrical resistivity from electrical conductivity requires an inversion and might introduce complications for temperature/field variable-dependent properties. Now you can define electrical resistivity directly. Electrical properties must be defined in terms of resistivity for materials with piezoresistive behavior.

For more information, see *Electrical Resistivity*.

Modified Darveaux Viscous Behavior for Parallel Rheological Framework

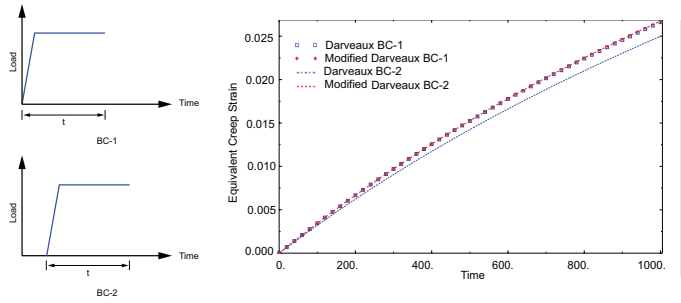
You can now use the modified Darveaux model to define creep response in the parallel rheological framework (PRF).

Benefits: The modified Darveaux model provides more accurate results when the step contains idle time or periods of zero loading.

The Darveaux creep law previously available with the PRF material model introduces an explicit dependency of creep strain rate on total time, which can lead to unexpected results when the load is not applied at time zero.

In the modified Darveaux model, the creep evolution law is revised by eliminating total time and using the accumulated total steady-state creep strain instead. This approach provides a solution that does not depend on the loading sequence when the step time has an idle or no-load period. Both models provide the same response under constant load and temperature applied at time zero.

To illustrate the difference between the original and modified Darveaux creep law, the following example considers uniaxial loading of a specimen fixed at one end and with a load applied at the other end. The load is ramped up rapidly and then kept constant. Two different loading cases are considered. For BC-1, the load is applied instantaneously at the first increment; for BC-2, the load is applied after some time with no loading. In both cases, the load is kept constant for the same amount of time. The results are compared below for both the original and modified Darveaux laws. For easier comparison, the results obtained from BC-2 are superimposed on BC-1 after shifting the time axis by the initial period with no load.



The figure illustrates that when the load is applied at the very first increment (BC-1), both the original and modified Darveaux models produce the same time history of equivalent creep strain. However, the original Darveaux model displays a different evolution of equivalent creep strain when the load is applied after a certain amount of time (BC-2). By contrast, the modified Darveaux model produces the same equivalent creep strain for both BC-1 and BC-2. This behavior is the expected result because both BC-1 and BC-2 have the same physical loading time.

For more information, see *Parallel Rheological Framework*.

Unsymmetric Solution Scheme for the Superelastic Model

Abaqus/Standard now automatically activates the unsymmetric solution scheme for the superelastic material model if you use the nonassociated flow rule.

Benefits: This capability improves convergence of the superelastic model with the nonassociated flow rule.

For more information, see *Superelasticity*.

Tangent Thermal Expansion

You can now define thermal expansion effects by specifying tangent thermal expansion coefficients.

Benefits: This capability provides flexibility in defining thermal expansion effects and eliminates the need to convert tangent thermal expansion coefficients when tangent coefficients are available.

In earlier releases, Abaqus/Standard required that the thermal expansion coefficients be defined in secant (or total) form. The total form is generally recommended because it leads to thermal strain results that are not dependent on the temperature and field variable history or sensitive to the time increment size. However, when a material undergoes an irreversible transformation (such as during the curing process of polymers), the evolution of thermal strains is better characterized in terms of tangent (or differential) expansion coefficients that are a

function of the degree of transformation. In these cases, it is generally not possible to convert the tangent form to an equivalent secant form of thermal expansion, and the only solution was to use user subroutine UEXPAN. Now you can define thermal expansion behavior directly in tangent form in Abaqus/Standard. As in the case of secant thermal expansion, the tangent thermal expansion can be isotropic, transversely isotropic, orthotropic, and fully anisotropic.

Both the secant and the tangent methods produce identical results if the coefficients are constant. If the expansion coefficients depend only on temperature, the conversion from tangent to secant form is always possible; therefore, we recommend that you use the secant form to eliminate the sensitivity of thermal strain results on time increment size. However, in situations when the coefficients depend on field variables, the two methods are not equivalent in general, and it is not possible to perform the conversion. For these situations you can now use the tangent definition.

To illustrate the benefits of the new functionality, consider an example of resin curing. During the curing process, the resin material undergoes two transformations. The first one is a transformation from a liquid to a solid phase, known as gelation. At this point, the material is in a rubbery state. As the glass transition temperature, T_g, increases and eventually exceeds the temperature of the curing process, the material transforms from the rubbery state to the glassy state. Assume that the tangent thermal expansion coefficients for the liquid, rubbery, and glass phases are given by ^{l}, ^{r}, and ^{g}, respectively. Using a value of degree of cure (DOC) of 0.5 as a measure of the gelation transformation from liquid to rubbery state, we can express the tangent thermal expansion coefficient as:

- = ^{l} if DOC 0.5,
- = ^{r} if DOC > 0.5 and (T_g-T) 0,
- = ^{g} if DOC > 0.5 and (T_g-T) > 0.

You can now use the tangent thermal expansion definition to define these thermal expansion coefficients in tabular form, as functions of two field variables associated with the degree of cure (DOC) and the difference between the glass transition temperature and temperature TGTDIFF=(T_g-T) as shown below (see also *Prediction of cure-induced spring-in of an angle bracket*).

```
*EXPANSION, DEFINITION=TANGENT, DEPENDENCIES=2
  l, , 0.50, 0.0
  r, , 0.51, 0.0
  l, , 0.50, 0.1
  g, , 0.51, 0.1
*USER DEFINED FIELD, TYPE=SPECIFIED
 1, DOC
 2, TGTDIFF
```

For more information, see *Thermal Expansion*.

Using Low-Density Foam with Eulerian Elements and ALE Analysis

You can now perform a coupled Eulerian-Lagrangian analysis or an analysis involving ALE adaptive meshing for a model with a low-density foam material.

Benefits: You can complete more effective simulations for analyses that involve extreme deformation or distortion of the mesh.
For analyses that involve extreme mesh deformation, coupled Eulerian-Lagrangian analyses are more effective than traditional Lagrangian analyses because the elements do not deform and materials are allowed to flow through the element faces. For an analysis involving ALE adaptive meshing, allowing the mesh to move

independently of the material enables you to maintain a high-quality mesh throughout an analysis, even when large deformation or loss of material occurs.

For more information, see *Low-Density Foams*.

2024 FD02

Nonlinear Viscoelastic Shear Behavior for EOS Materials

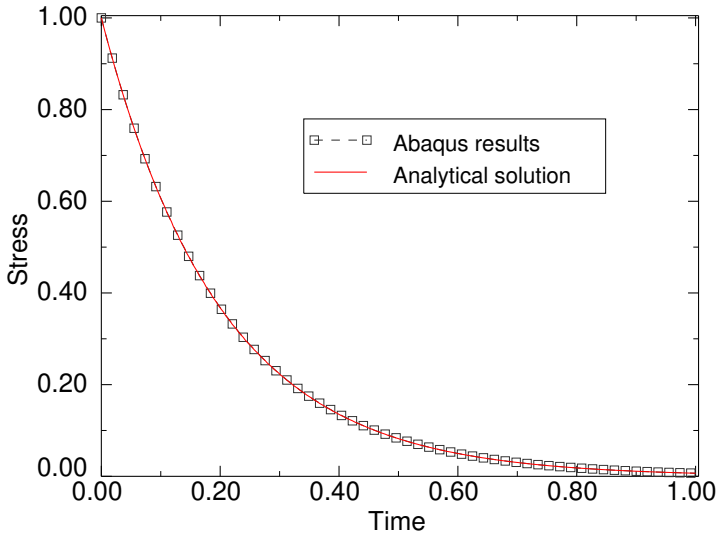
You can now combine nonlinear viscoelastic shear behavior with equation of state (EOS) materials in Abaqus/Explicit to model the response of viscoelastic fluids.

Benefits: The new material capability lets you model viscoelastic fluids and improves the stable time increment when you model fluids with large viscosities.

The family of EOS materials in Abaqus/Explicit now allows you to define nonlinear viscoelastic shear behavior based on a Maxwell-type model (that is, elastic and viscous effects acting in series). In earlier releases, you could define the deviatoric response of EOS materials using only elastic or viscous shear behavior.

The new functionality also provides an efficient method to model viscous fluids with very high viscosity. As the viscosity of a fluid increases, it leads to a significant reduction in the stable time increment required for an explicit dynamic simulation. You can use the new nonlinear viscoelastic behavior to approximate a purely viscous shear response by treating the shear modulus, μ, as a penalty stiffness that you can adjust to achieve a reasonable stable time increment without significant influence on the overall shear viscous response.

The figure below shows the history output of shear stress in a simple shear stress relaxation test with the new material model (dashed line with symbols). The plot compares well with the analytical solution (solid line).



For more information, see *Deviatoric Behavior*.

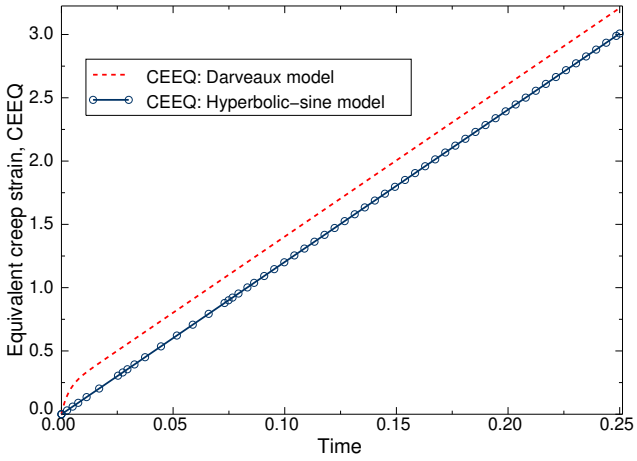
Darveaux Viscous Behavior for Parallel Rheological Framework

You can now use the Darveaux model to define creep response in the parallel rheological framework.

Benefits: The Darveaux model allows you to consider both primary and secondary creep, which enables more accurate results.

In earlier releases, the Darveaux creep model was available in Abaqus/Standard to define creep behavior with linear elasticity as well as the viscous behavior for a two-layer viscoplastic material. This creep model can be useful for capturing rate-dependent behavior in certain classes of polymer such as Nylon 12. You can now use this creep model to define the viscous behavior in models defined using the parallel rheological framework in both Abaqus/Standard and Abaqus/Explicit. The model takes into account both primary (transient) and secondary (steady-state) creep. The secondary creep in the Darveaux model is described by the hyperbolic-sine law, which is modified to include the primary creep contribution that occurs immediately after the stress is applied and before a steady-state creep stage is reached.

To illustrate the difference between the hyperbolic-sine and the Darveaux models, the following example considers a specimen that is fixed at one end and to which a load is applied at the other end. The load is applied rapidly and then kept constant. The specimen's elastic response is modeled using a Hencky hyperelastic material, and the creep behavior is described either by the Darveaux or by the hyperbolic-sine creep model. The results are compared below.



The hyperbolic-sine law model predicts a response that shows the constant equivalent creep strain rate, which often accurately captures the steady-state response at a constant load accurately. The response of the Darveaux model includes the transient creep effects, where the equivalent creep strain rate increases very quickly, stabilizes when the steady-state creep stage is reached, and then remains constant (as in the case of the hyperbolic-sine law).

For more information, see *Parallel Rheological Framework*.

Hencky Hyperelastic Model in Abaqus/Standard

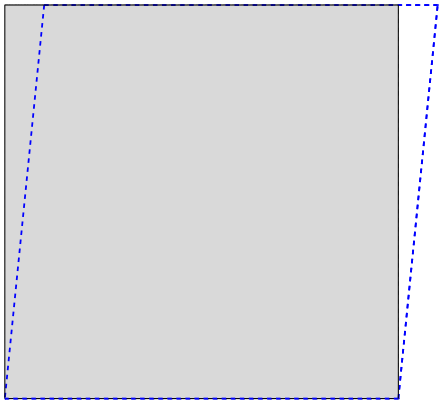
You can now use the Hencky model to define hyperelastic material response accurately.

Benefits: You can now accurately predict elastic material response for moderately large deformations.

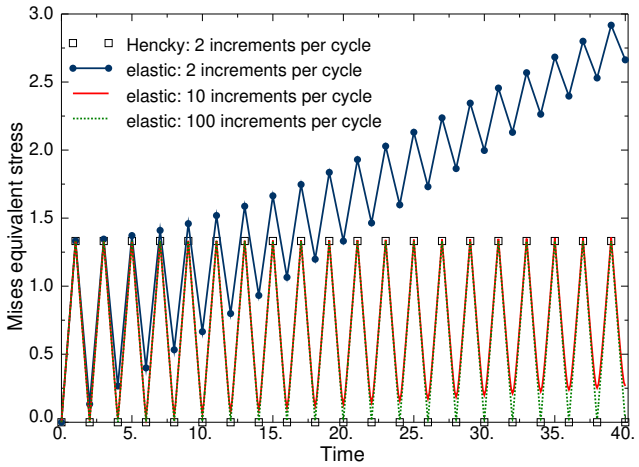
The Hencky model is an isotropic hyperelastic model that assumes a linear relation between the Kirchhoff stress tensor and the Hencky or logarithmic strain tensor. It is a generalization of the classical Hooke's law for isotropic infinitesimal elasticity. You define the model by providing two material parameters that can be measured at infinitesimal strains. The Hencky model can produce results that show good agreement with experimental data for a wide class of materials for moderately large deformations. You can also use the model as a finite strain alternative to the isotropic linear elastic model in situations where the isotropic linear elastic model is known to

perform poorly (for example, for materials subjected to cyclic loads). In this case, the isotropic linear elastic material, which uses a rate formulation, might produce residual stresses when it returns to the initial (stress-free) configuration. The Hencky model does not show this undesirable behavior because it uses the total deformation to compute stresses.

The figure below shows a specimen subjected to simple shear cycling deformations with the mechanical response modeled using the Hencky and isotropic linear elastic models. Twenty cycles are performed with a different number of increments per cycle. The reference (shaded area) and final (dashed line) configurations are shown.



The plots below show the Mises equivalent stresses as a function of time.



The results for the isotropic linear elastic material accumulate significant errors for a small number of increments per cycle, resulting in large residual stresses after a few cycles. They improve for larger numbers of increments per cycle and stabilize when the number increases to about 100. In contrast, the Hencky model shows no residual stresses even in the case where only two increments are used per cycle.

For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

2024 FD01

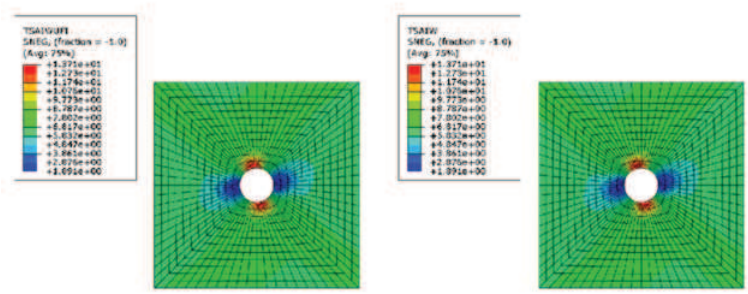
Generalized Damage and Failure for Anisotropic Materials

You can now specify new failure criteria for fully anisotropic materials to evaluate the strength of the material based on stress or strain results. You can also use the same failure criteria at the pseudo-grain level for multiscale materials.

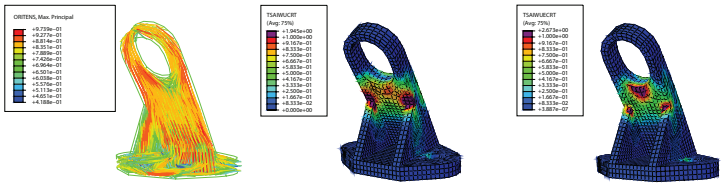
Benefits: The new stress/strain-based failure criteria are available with any material type and all stress/displacement elements, which gives you more options to evaluate the strength of materials.

Abaqus now provides a collection of stress/strain-based generalized damage initiation criteria as indicators of material failure. These general stress/strain-based damage initiation criteria are based on Tsai-Wu theory or maximum stress/strain theory. Abaqus does not support damage evolution laws for these damage criteria.

The figure below shows the failure index evaluated using the Tsai-Wu failure theory in an open hole tension test. The coupon plate is modeled with composite shell elements with $[45/0/45/90]_s$ layups.



The following figure shows the failure criteria evaluated using the Tsai-Wu stress and strain theory at the pseudo-grain level in an injection-molded plastic part. The maximum principal direction of the fiber orientation tensor is also shown below.



For more information, see *Generalized Damage and Failure for Anisotropic Materials*.

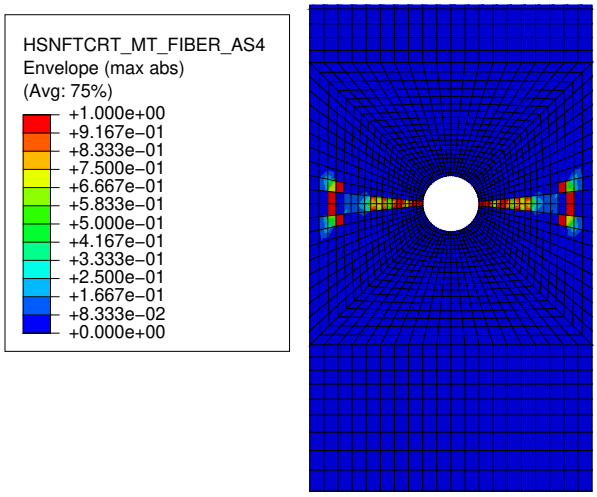
Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling

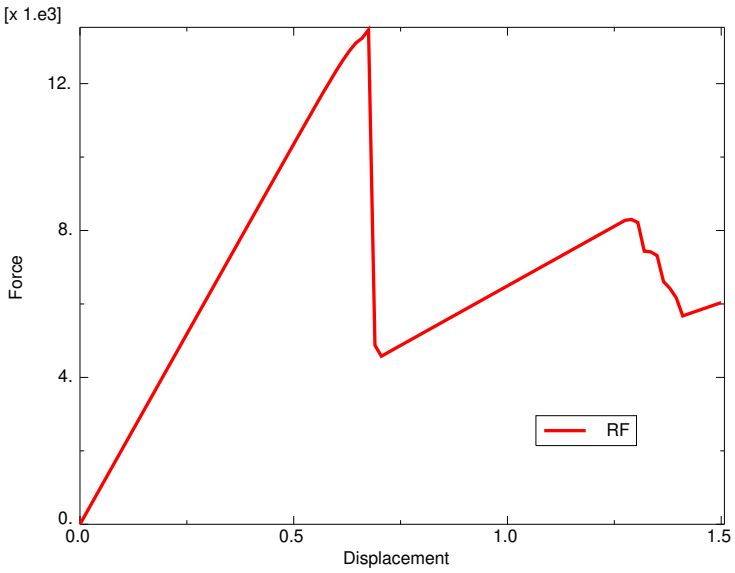
You can now specify new failure criteria for fiber-reinforced composite materials to evaluate the strength of the material based on stress invariants at the micro-level. You can also specify damage evolution laws for these damage criteria.

Benefits: The new damage initiation criteria allow you to evaluate damage at the composite constituent level using multiscale modeling, which simplifies damage modeling.

Abaqus/Standard offers capabilities for predicting the onset of damage and for modeling progressive damage and failure at the constituent level in fiber-reinforced composites, which include unidirectional and woven fiber-reinforced composite materials. You can use the criteria with both 2D and 3D solid and shell stress/displacement elements. You can specify only linear responses in the constituents. You can also use the new damage initiation criteria in 3D elements; in earlier releases, these criteria were available for plane stress elements only.

The figures below show the results of an open hole tension test modeled with composite solid elements with $[45/0/45/90]_s$ layups. The top figure shows the fiber damage initiation criteria, and the bottom figure shows the reaction force result versus the displacement.





For more information, see *Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling*.

Field Expansion in Pore Fluid in a Porous Medium

You can now use field expansion with the pore fluid phase in a porous medium.

Benefits: The ability to define field expansion in the solid and the fluid phases in a porous medium independently provides a convenient way to introduce expansion of the material due to physical sources other than temperature, without having to model such sources explicitly.

In earlier releases, you could only define field expansion for the solid grains that make up the skeleton in a porous medium. Now, you can define field expansion in the different phases in a porous medium; that is, the solid grains and the pore fluid, independently. You can define only isotropic field expansion behavior for the pore fluid.

In addition, you can also output the strains due to both field expansion and thermal expansion in the pore fluid.

For more information, see *Thermal Expansion* and *Field Expansion*.

Band-Limited Damping in Abaqus/Standard

You can now specify band-limited damping in Abaqus/Standard.

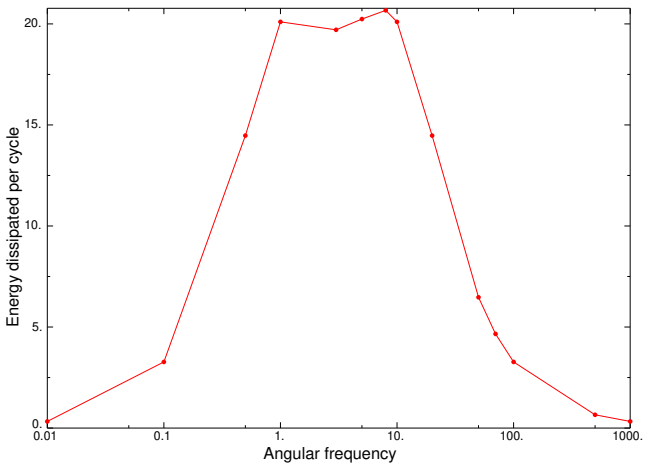
Benefits: The band-limited damping allows you to specify a damping ratio that remains approximately constant within a frequency range specified.

In many solid materials subjected to oscillatory loads, the energy dissipated in a cycle is controlled by internal friction and is insensitive to the frequency of the oscillation. This behavior cannot be captured by the stiffness proportional (beta) or mass proportional (alpha) damping options previously available in Abaqus/Standard, but can be modeled using the band-limited damping option. For this model, you specify the damping ratio, which can be defined as a function of temperature variables, field variables, or both. In addition, you provide the low and high frequency cutoff values to specify the frequency range, and Abaqus maintains approximately constant

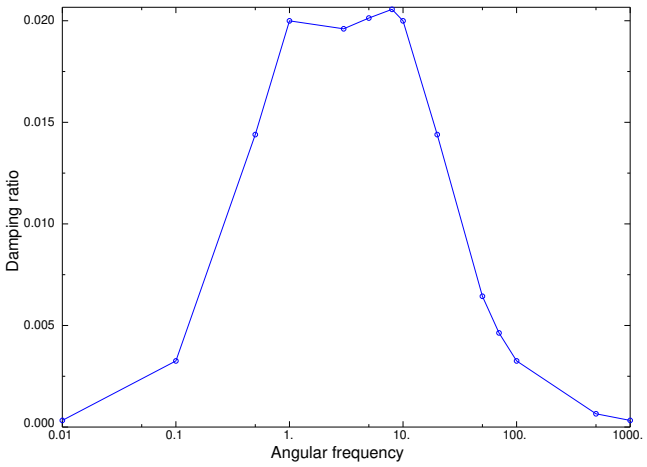
value of the damping ratio (and energy dissipated per cycle) within this range. In Abaqus/Standard, the effect of band-limited damping is considered only with the implicit dynamic procedure.

The image below shows a bar subjected to harmonic excitations. The bar is fixed at the left end, and a displacement boundary condition is applied at the other end with different forced frequencies. The bar is modeled using a linearly elastic material in combination with band-limited damping. A constant damping ratio of 0.02 is specified, the low angular frequency is 1, and the high angular frequency is 10.

The plot below shows the energy dissipated as a function of angular frequency, which remains approximately constant within the frequency range specified and decreases rapidly outside of the range.



The plot below shows the damping ratio, obtained by scaling the energy dissipated in a cycle by the maximum value of the strain energy, which follows a similar pattern. As expected, Abaqus achieves a damping ratio approximately equal to 0.02 within the specified range of frequencies.



For more information, see *Material Damping*.

Output

This page describes recent changes in Abaqus Output.

2026 FD04

Damage Dissipation Output for Individual Contact Surfaces

You can now use output variable ALLDMD to request the damage dissipation associated with individual contact surfaces.

Benefits: You can separate the contact contributions on individual surfaces to damage dissipation from contributions from other sources (such as damage in underlying elements).
You can specify the SURFACE parameter on the **ENERGY OUTPUT* option to request accumulated damage dissipation on individual surfaces due to their participation in damage evolution from contact. Therefore, you can identify the contact regions that contribute most to damage dissipation.

In Abaqus/Explicit you can further refine the request by including the SECOND SURFACE parameter to limit the output to interactions between a specific pair of surfaces.

For more information, see *Whole and Partial Model Variables* and **ENERGY OUTPUT*.

Cap Position Output for Cap Plasticity Materials

You can use the new output variable PBCAP to output the cap position in materials with cap plasticity.

Benefits: This feature expands the functionality of Abaqus/Standard and Abaqus/Explicit.
In earlier releases, the cap position was available only via output variable PEEQ. Now, when you request PEEQ, PBCAP is used instead. The numerical value of the output is the same.

For more information, see *Element Integration Point Variables* and *Element Integration Point Variables*.

Output of Nonquadratic Yield in Abaqus/Standard

You can now use output variable YIELDPOT to output the yield potential in Abaqus/Standard.

Benefits: This feature expands the functionality of Abaqus/Standard.
In earlier releases, output variable YIELDPOT was available only in Abaqus/Explicit.

For more information, see *Element Integration Point Variables* and *Element Integration Point Variables*.

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.
Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
EREQ	yes	yes	no	n/a	Abaqus/Explicit

Output Variable	Field	History	.fil	.dat	Product
Strain rate used to evaluate strain-rate dependent damage initiation criteria.					
PBCAP Cap position for the cap plasticity model.	yes	yes	yes	yes	Abaqus/Standard
	yes	yes	yes	n/a	Abaqus/Explicit
PBCAPMAX Maximum cap position, PBCAP, among all the section points. For a shell element, PBCAPMAX represents the maximum PBCAP value among all the section points in the layer. For a beam element, PBCAPMAX is the maximum PBCAP among all the section points in the cross-section. For a solid element, PBCAPMAX represents the PBCAP at the integration points.	yes	no	no	no	Abaqus/Standard
	yes	no	no	n/a	Abaqus/Explicit
PBCAPVAVG Cap position for the cap plasticity model, computed as a volume fraction weighted average of all the materials in the Eulerian element.	yes	no	no	n/a	Abaqus/Explicit

Surface Output

Output Variable	Field	History	.fil	.dat	Product
CICPS	no	yes	no	n/a	Abaqus/Explicit

Output Variable	Field	History	.fil	.dat	Product
Scalar integration of the contact pressure over the surface.					

Total Energy Output

Output Variable	Field	History	.fil	.dat	Product
ALLDMD Contact damage dissipation energy.	no	yes	no	n/a	Abaqus/Explicit

For more information, see *Element Integration Point Variables*, *Element Integration Point Variables*, and *Total Energy Output*.

2026 FD03

Filtering Output

Abaqus/Explicit now supports output filtering for total model energy history and incrementation output.

Benefits: The new supported output requests extend the output filtering functionality of Abaqus/Explicit.

For more information, see *Filtering Output and Operating on Output in Abaqus/Explicit*.

Sensor Functionality

Abaqus/Standard and Abaqus/Explicit now support the total energies as sensors.

Benefits: The new sensors expand the functionality of Abaqus/Standard and Abaqus/Explicit.
The new sensors can be used now with user subroutines *UAMP* and *VUAMP* and with step control capability in Abaqus/Standard.

For more information, see *Defining Sensors*.

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.
Whole and Partial Model Variables

Output Variable	Field	History	.fil	.dat	Product
VOLUMEFRAC Total fluid volume contained within the fracture.	no	yes	no	no	Abaqus/Standard

Output Variable	Field	History	.fil	.dat	Product
VOLUMELEAKBOT Total leak-off volume from the bottom fracture surface.	no	yes	no	no	Abaqus/Standard
VOLUMELEAKTOP Total leak-off volume from the top fracture surface.	no	yes	no	no	Abaqus/Standard
VOLUMESLURRY Total slurry volume within the fracture (slurry workflows only).	no	yes	no	no	Abaqus/Standard

Contact and Element Output Quantities

Output Variable	Field	History	.fil	.dat	Product
CDISPR Contact opening and slip rates.	yes	no	no		Abaqus/Explicit
ALLCDCOH Viscous regularization energy in cohesive elements and cohesive contact.	no	yes	no	no	Abaqus/Standard

Element Section Variables

Output Variable	Field	History	.fil	.dat	Product
STHIN Section thinning or thickening.	yes	yes	no	no	Abaqus/Standard

For more information, see *Element Section Variables*, *Surface Variables*, and *Whole and Partial Model Variables*.

2026 FD01

Filtering Output

Abaqus/Explicit now supports output filtering for all general contact field output variables and for all element set energy history output.

Benefits: The new supported output requests expand the output filtering capabilities of Abaqus/Explicit.

For more information, see *Filtering Output and Operating on Output in Abaqus/Explicit*.

Support of FILM and RADIATION Specific Output Variables with Evolving Faces of an Element

Output variables FILM and RADIATION are now supported with the usage of specific output variables with evolving faces of an element.

Benefits: The support expands the capabilities of Abaqus/Standard.

Abaqus/Standard now supports the following:

- Output variables FILMCOEF and SINKTEMP with the usage of element-based film conditions on evolving faces of an element.
- Output variable AMBIENTTEMP with the usage of element-based radiation conditions on evolving faces of an element.

These output variables are available only on the existing element facets and not on internal cut facets. In the presence of **TIE*, the evolving film and radiation conditions are automatically turned off when the elements involved in the TIE come active.

For more information, see *Element Face Variables* and **TIE*.

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.
Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
MSDMG	yes	yes	no	no	Abaqus/Standard
The Mullins effect damage variable. Available only when the Mullins effect is modeled.	yes	yes	no		Abaqus/Explicit
PEEQTSHR	yes	yes	no	no	Abaqus/Standard
Equivalent plastic strain in out-of-plane shear for paper board transverse shear plasticity.	yes	yes	no		Abaqus/Explicit
UMULMAX	yes	yes	no	no	Abaqus/Standard
The maximum deviatoric strain energy density for	yes	yes	no		Abaqus/Explicit

Output Variable	Field	History	.fil	.dat	Product
rubberlike materials. The maximum strain energy density for elastomeric foams. Available only when the Mullins effect is modeled.					

Nodal Variables

Output Variable	Field	History	.fil	.dat	Product
PAC Acoustic pressure. Alias for POR output variable.	yes	yes	yes	yes	Abaqus/Standard

For more information, see *Mullins Effect Quantities*, *Mullins Effect Quantities*, *Tensors and Invariants*, *Tensors and Associated Principal Values and Invariants*, and *Acoustic Quantities*.

2026 GA

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.

Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
CEEQNET	yes	yes	no	no	Abaqus/Standard
Equivalent creep strain in all the networks. Available only for the parallel rheological framework.	yes	yes	no		Abaqus/Explicit
CEEQNET*k*	yes	yes	no	no	Abaqus/Standard
Equivalent creep strain in the *k*th network (1 *k*). Available only for the parallel rheological framework.	yes	yes	no		Abaqus/Explicit

Output Variable	Field	History	.fil	.dat	Product
CEEQSSNET Steady-state equivalent creep strain in all the networks that use the modified Darveaux creep law. Available only for the parallel rheological framework.	yes	yes	no	no	Abaqus/Standard
	yes	yes	no		Abaqus/Explicit
CEEQSSNET*k* Steady-state equivalent creep strain in the *k*th network (1 *k*) if it uses the modified Darveaux creep law. Available only for the parallel rheological framework.	yes	yes	no	no	Abaqus/Standard
	yes	yes	no		Abaqus/Explicit
SDEFRESNET Deformation resistance in all the networks that use the Anand creep law. Available only for the parallel rheological framework.	yes	yes	no	no	Abaqus/Standard
	yes	yes	no		Abaqus/Explicit
SDEFRESNET*k* Deformation resistance in the *k*th network (1 *k*) if it uses the Anand creep law. Available only for the parallel rheological framework.	yes	yes	no	no	Abaqus/Standard
	yes	yes	no		Abaqus/Explicit

For more information, see *Parallel Rheological Framework Quantities* and *Parallel Rheological Framework Quantities*.

2025 FD01

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.

Total Energy Output Quantities

Output Variable	Field	History	.fil	.dat	Product
ALLERPWR Equivalent radiated power emitted by a panel.	no	yes	no	no	Abaqus/Standard

Nodal Variables

Output Variable	Field	History	.fil	.dat	Product
ERPAC Complex-valued acoustic pressure on the radiating element-based structural surface.	yes	yes	no	no	Abaqus/Standard
ERPWR Equivalent radiated power on an element-based structural surface.	yes	yes	no	no	Abaqus/Standard
ERPWRDEN Equivalent radiated power density on an element-based structural surface.	yes	yes	no	no	Abaqus/Standard

For more information, see *Nodal Variables* and *Whole and Partial Model Variables*.

PDF Guides Available

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

Benefits: You can easily access the PDF versions of the guides.

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

For more information, see *Abaqus Output Guide*.

2024 FD03

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.

Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
SOC_*i* State of charge in particle *i* (1 *i* 3).	yes	yes	no	no	Abaqus/Standard

Element Section Variables

Output Variable	Field	History	.fil	.dat	Product
UVARPT Element user-defined output variables.	yes	yes	no	no	Abaqus/Standard
UVARPT*n* Element user-defined output variable *n*.	yes	yes	no	no	Abaqus/Standard

Whole Element Variables

Output Variable	Field	History	.fil	.dat	Product
UVARE Element user-defined output variables.	yes	yes	no	no	Abaqus/Standard
UVARE*n* Element user-defined output variable *n*.	yes	yes	no	no	Abaqus/Standard

Whole and Partial Model Variables

Output Variable	Field	History	.fil	.dat	Product
SOC Current state of charge in the specified region of the model.	no	yes	no	no	Abaqus/Standard

For more information, see *Element Integration Point Variables*, *Whole Element Variables*, and *Whole and Partial Model Variables*.

2024 FD02

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.

Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
PORVAVG Equivalent pore pressure, computed as a volume fraction weighted average of all materials in the element.	yes	no	no		Abaqus/Explicit
SDEFRES Deformation resistance in Anand's creep model.	yes	yes	no	no	Abaqus/Standard
TG Glass transition temperature.	yes	yes	no	no	Abaqus/Standard
TGTDIFF The difference between the glass transition temperature and the temperature.	yes	yes	no	no	Abaqus/Standard

Element Section Variables

Output Variable	Field	History	.fil	.dat	Product
SK Section curvature change and twist components.	yes	yes	yes	yes	Abaqus/Standard

For more information, see *Element Integration Point Variables*, *Element Section Variables*, and *Element Integration Point Variables*.

2024 FD01

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.
Element Integration Point Variables

Output Variable	Field	History	.fil	.dat	Product
MSTRESSCRT Maximum stress-based damage initiation criterion.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
MSTRAINCRT Maximum strain-based damage initiation criterion.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
TSAIWUCRT Tsai-wu stress-based damage initiation criterion.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
TSAIWUECRT Tsai-wu strain-based damage initiation criterion.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
DMIFI All active components of the damage initiation failure indices.	yes	yes	no	no	Abaqus/Standard
	yes	yes	no		Abaqus/Explicit
MSTRESSFI Maximum stress-based damage initiation failure index.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
MSTRAINFI Maximum strain-based damage initiation failure index.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
TSAIWUFI Tsai-Wu stress-based damage initiation failure index.	yes	yes	no	no	Abaqus/Standard
	no	yes	no		Abaqus/Explicit
TSAIWUEFI	yes	yes	no	no	Abaqus/Standard

Output Variable	Field	History	.fil	.dat	Product
Tsai-Wu strain-based damage initiation failure index.	no	yes	no		Abaqus/Explicit
TSINVMTCRT Transversely isotropic stress invariant–based matrix tensile damage initiation criterion.	yes	yes	no	yes	Abaqus/Standard
TSINVMCCRT Transversely isotropic stress invariant–based matrix compression damage initiation criterion.	yes	yes	no	yes	Abaqus/Standard
THEFL Thermal strain in the pore fluid in a porous medium.	yes	yes	no	no	Abaqus/Standard
FVE All tensor components of all field expansion strain tensors.	yes	yes	no	no	Abaqus/Standard
FVEFL All field expansion strains in the pore fluid in a porous medium.	yes	yes	no	no	Abaqus/Standard
FVE*n* All tensor components for the field expansion strain tensor due to field variable number *n*.	yes	yes	no	no	Abaqus/Standard
FVEFL*n* Field expansion strain in the pore fluid in a porous medium due	yes	yes	no	no	Abaqus/Standard

Output Variable	Field	History	.fil	.dat	Product
to field variable number *n*.					

Element Face Variables

Output Variable	Field	History	.fil	.dat	Product
CWEAR Accumulated scalar nodal contact wear distance.	yes	yes	no	no	Abaqus/Standard

Surface Variables

Output Variable	Field	History	.fil	.dat	Product
AMBIENTTEMP Reference ambient temperature on element faces from prescribed boundary radiation.	yes	no	no	no	Abaqus/Standard

For more information, see *Element Integration Point Variables* and *Element Integration Point Variables*.

2024 GA

New Output Variables

New variables are available.

Benefits: The new output variables expand the capabilities of Abaqus/Standard and Abaqus/Explicit.

Nodal Variables

Output Variable	Field	History	.fil	.dat	Product
STIFN Local normal stiffness.	yes	no	no	no	Abaqus/Standard
VN Complex-valued surface normal velocity.	yes	yes	no	no	Abaqus/Standard—available only for steady-state (frequency domain) dynamic analyses (modal and direct)
VNSQ	yes	yes	no	no	Abaqus/Standard—available only for steady-state

Output Variable	Field	History	.fil	.dat	Product
Real-valued surface normal velocity squared.					(frequency domain) dynamic analyses (modal and direct)
AVNSQ Area-weighted surface normal velocity squared or acoustic power normalized by the acoustic impedance of the surrounding fluid.	yes	yes	no	no	Abaqus/Standard—available only for steady-state (frequency domain) dynamic analyses (modal and direct)
FEXT All components of external point loads from a co-simulation or external field definition.	yes	yes	no		Abaqus/Explicit
MEXT All components of external point moments from a co-simulation or external field definition.	yes	yes	no		Abaqus/Explicit

Structural Optimization Variables

Output Variable	Field	History	.fil	.dat	Product
DISP_NORMAL_VAL The value of the bead optimization displacement along the node normal vector.		no	no	no	Abaqus/Standard—output automatically during shape optimization

For more information, see *Abaqus/Standard Nodal Variables*, *Structural Optimization Variables*, and *Abaqus/Explicit Nodal Variables*.

Prescribed Conditions

This page describes recent changes in Abaqus Prescribed Conditions.

2026 FD04

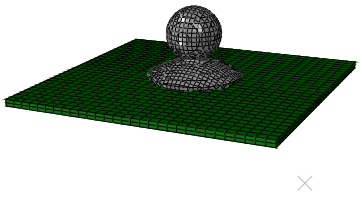
Modeling Surface Tension Loads with Eulerian Elements in Abaqus/Explicit

You can now specify surface tension loads on Eulerian material surfaces in Abaqus/Explicit.

Benefits: You can model large deformation under surface tension loads, including droplet coalescence and droplet break up.

In earlier releases, surface tension loads on element-based surfaces were available only for Lagrangian solid, membrane, and surface elements. Lagrangian elements can get distorted under very large deformations, limiting their ability to model droplet coalescence or break up. Modeling surface tension loads with Eulerian elements is more suitable for such applications.

The animation below shows two fluid droplets with an initial cubic shape, transforming into spherical shapes and ultimately coalescing into a single body under the action of surface tension loads in Abaqus/Explicit. The two fluid droplets have a fixed contact angle and are modeled as a linear equation of state (EOS) material of type USUP using EC3D8R elements. A surface tension load is applied on the Eulerian material surface, which is created automatically on the exterior of a particular Eulerian material instance.



For more information, see **DSLOAD* and *Surface Tension Loads*.

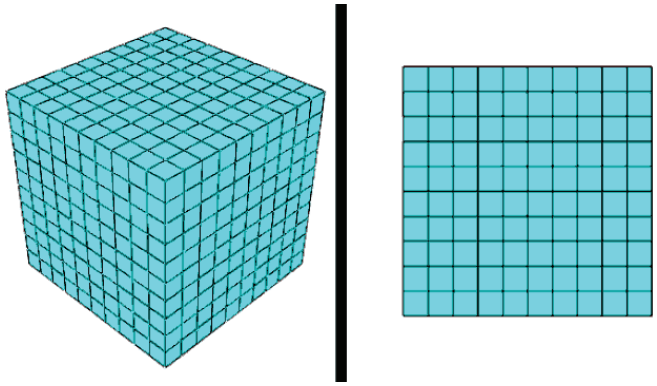
2026 FD01

Surface Tension Loads in Abaqus/Standard

You can now specify surface tension loads on element-based surfaces in Abaqus/Standard.

Benefits: This feature expands the capabilities of Abaqus/Standard.
In earlier releases, surface tension loads on element-based surfaces were available only in Abaqus/Explicit.

The animation below shows a cube being transformed into a sphere under surface tension loads in Abaqus/Standard. The cube is modeled as an elastic plastic material with a very small yield stress using C3D8H elements in a static analysis. A surface tension load is applied on all facets of the elements forming the exterior face.



For more information, see **DSLOAD* and *Surface Tension Loads*.

Long-Term Stress Initial Conditions

You can now specify initial stresses for viscoelastic materials in a fully relaxed state.

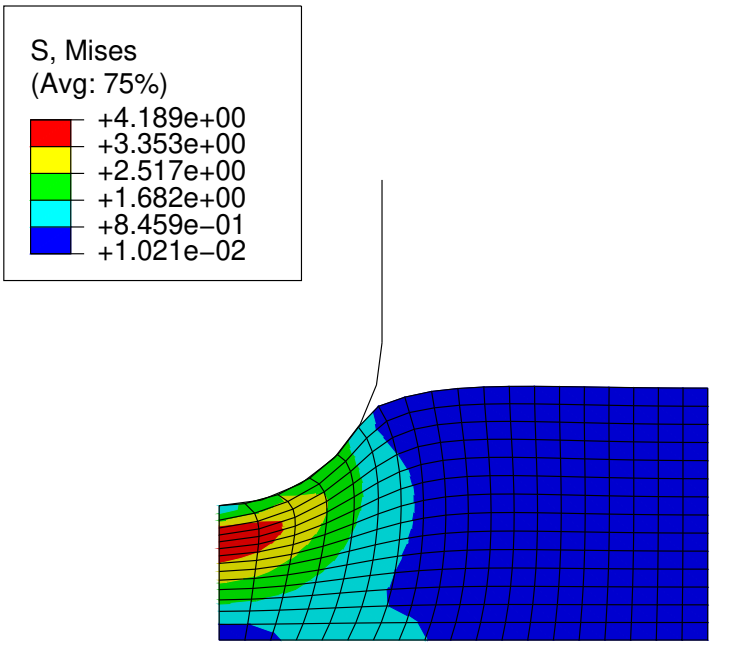
Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

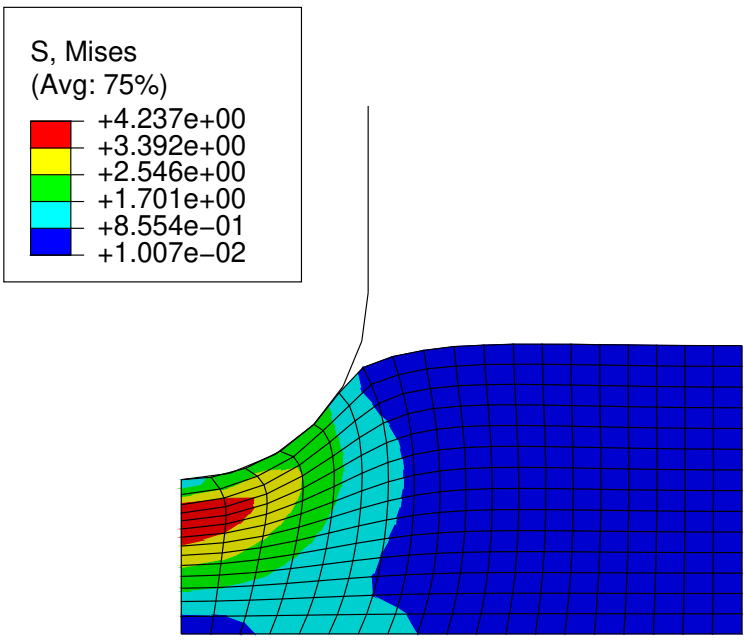
By default, when you specify initial stresses for a viscoelastic material, Abaqus assumes that they are instantaneous (that is, applied very fast). Now, you can specify initial stresses in the case where load is applied for a long enough time for the material to reach a fully relaxed state. An example of such a situation is a part that is preloaded during installation and then left in this state for a long time. In this case, if the stresses at the end of this period are known and are used to specify initial stresses, you should use the new long-term initial stress capability.

To illustrate this capability, we consider a specimen that is indented by a hemispherical punch. The specimen is modeled using a hyperelastic material with finite-strain linear viscoelasticity. Two Abaqus/Standard analyses are performed as follows:

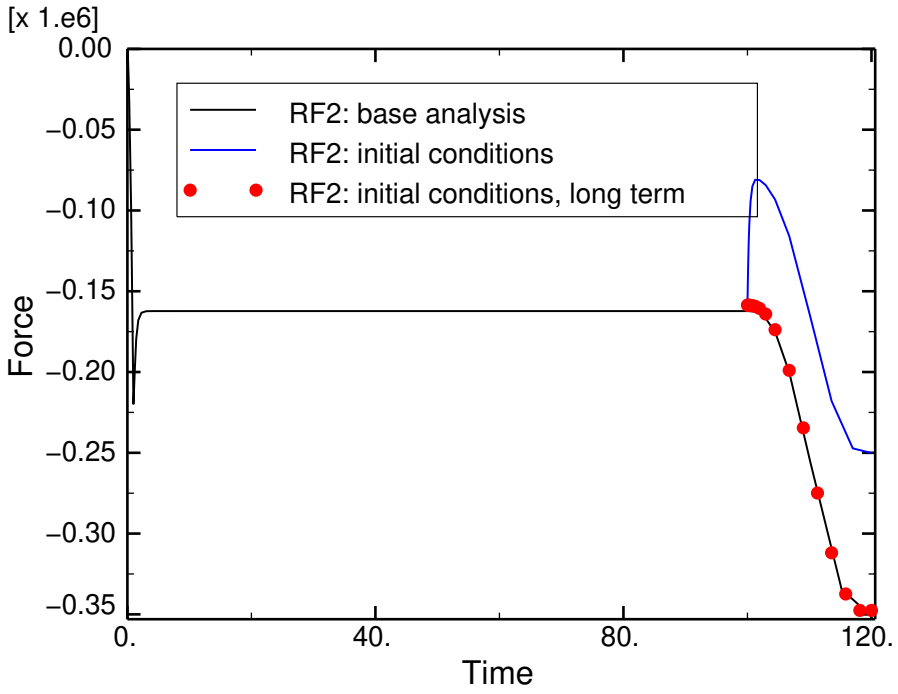
- The first analysis consists of two steps. In the first step, the punch moves down and then stays in this position long enough to reach the fully relaxed state. In the second step, the punch is moved down again.
- The initial configuration of the second analysis is created using the deformed configuration at the end of the first step of the previous analysis. The long-term initial stresses are specified, which correspond to the end of the first step. Finally, the punch is moved down the same way as in the second step of the first analysis.

The contour plots below show the stresses at the end of the second step of the first analysis (top) and at the end of the second analysis (bottom). It is expected that these results will match, and they show a good agreement.


S, Mises
(Avg: 75%)
+4.189e+00
+3.353e+00
+2.517e+00
+1.682e+00
+8.459e−01
+1.021e−02


S, Mises
(Avg: 75%)
+4.237e+00
+3.392e+00
+2.546e+00
+1.701e+00
+8.554e−01
+1.007e−02

Similarly, the reaction forces at the punch reference node correlate well, as shown below. However, the reaction force for the case when instantaneous initial stresses are prescribed instead of long-term stresses shows substantial discrepancies.



For more information, see *Defining Initial Stresses*.

Initial Stress for Materials with Mullins Effect

You can now specify initial stresses for materials with any of the Mullins effect models available in Abaqus.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

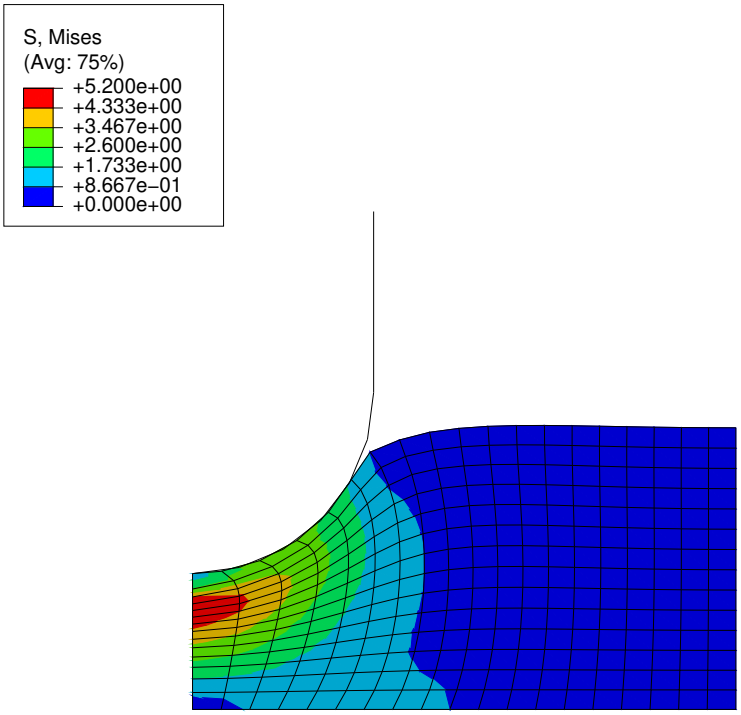
In earlier releases, initial stresses were supported only for materials using the Ogden-Roxburgh or user-defined Mullins effect models. Now, you can specify initial stresses for materials using any of the Mullins effect models available in Abaqus. In addition, you can now use initial stresses in conjunction with the initial maximum energy density in the Mullins effect model to determine if the stress lies on the loading or the unloading path. If the initial maximum energy density is larger than that computed using the initial stresses, the stress lies on the unloading path; otherwise, the stress lies on the loading path.

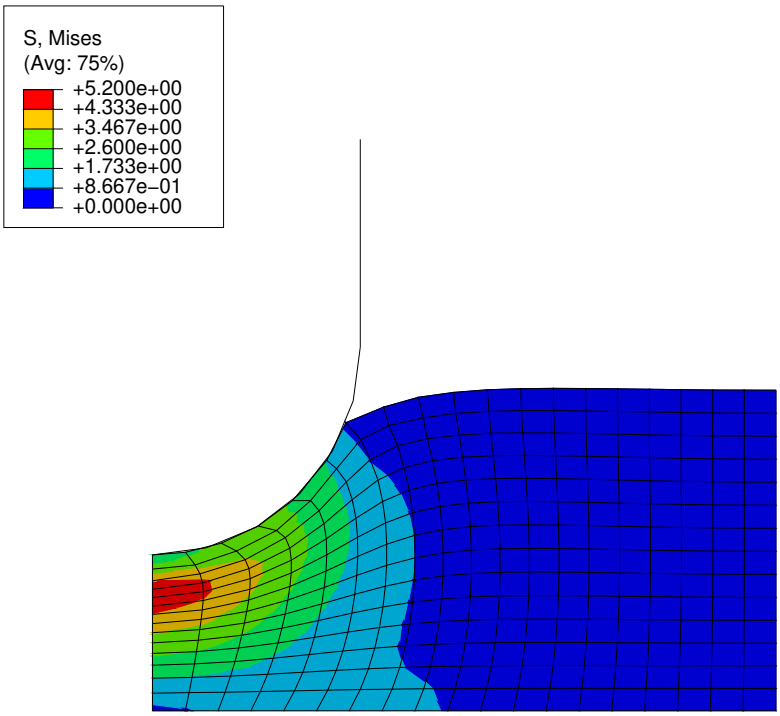
In the example below, a specimen modeled using a hyperelastic material with the Ogden-Roxburgh Mullins effect model is subjected to a nonmonotonic load by a hemispherical punch. Two analyses are performed as follows:

- First, a two-step Abaqus/Standard analysis is performed. In the first step, the punch is moved down by 120 mm and then moved up by 30 mm. In the second step, the punch is moved down again by 60 mm.
- The initial configuration of the second, Abaqus/Explicit, analysis corresponds to the deformed configuration of the Abaqus/Standard analysis at the end of the first step. The initial stresses and initial maximum energy

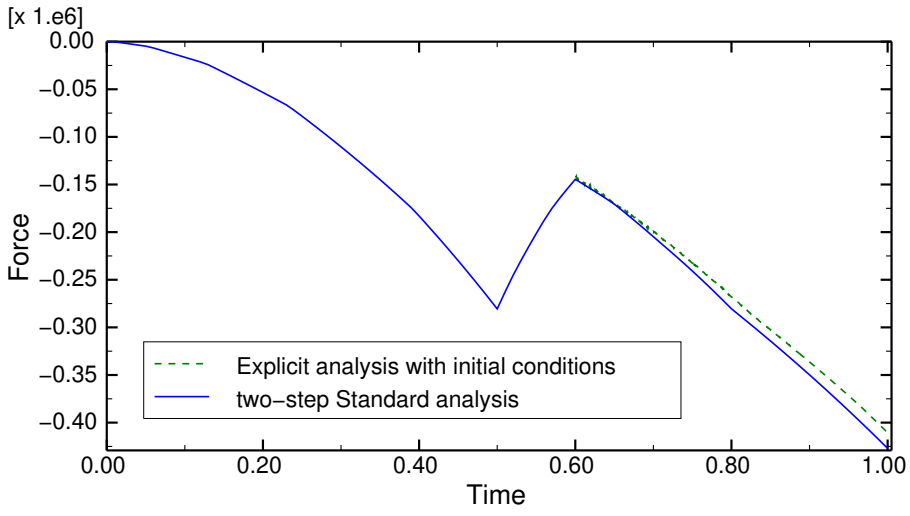
density of the Mullins model are prescribed from the end of the first step of the Abaqus/Standard analysis. The second analysis consists of one step in which the punch is moved down by 60 mm.

The contour plots below show the stress results at the end of the first analysis (top) and at the end of the second analysis (bottom).





The reaction forces in the punch reference node are plotted below.



Both the stress results and the reaction forces show good agreement as expected.

For more information, see *Defining Initial Stresses*.

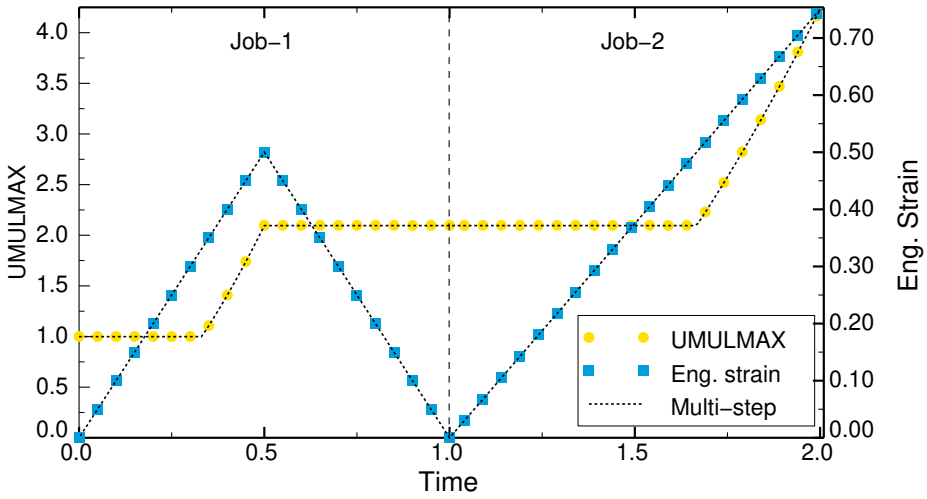
Initial Maximum Energy Density in Mullins Effect Models

You can now define the initial value of the maximum energy density used in Mullins effect models.

Benefits: This feature expands the capabilities of Abaqus/Standard and Abaqus/Explicit.

To specify the initial value of the maximum energy density used in Mullins effect models, use the new **INITIAL CONDITIONS*, TYPE=MULLINS option. If required, you can use the **EXTERNAL FIELD* option in conjunction with this option.

The plot below shows the evolution of UMULMAX across two separate simulations, Job-1 and Job-2, of a unit cube subjected to uniaxial tension.



The value of UMULMAX is set to 1.0 at the start of Job-1 without the use of **EXTERNAL FIELD* and reaches a value of 2.09 upon deformation. This new value is set as the initial condition for Job-2 using **EXTERNAL FIELD*. These results match the corresponding results for the equivalent single multistep simulation that are included for reference.

For more information, see *Defining Initial Maximum Energy Density in Mullins Effect Models*.

2026 GA

Contact-Dependent Thermal Radiation Surface Loads in Abaqus/Explicit

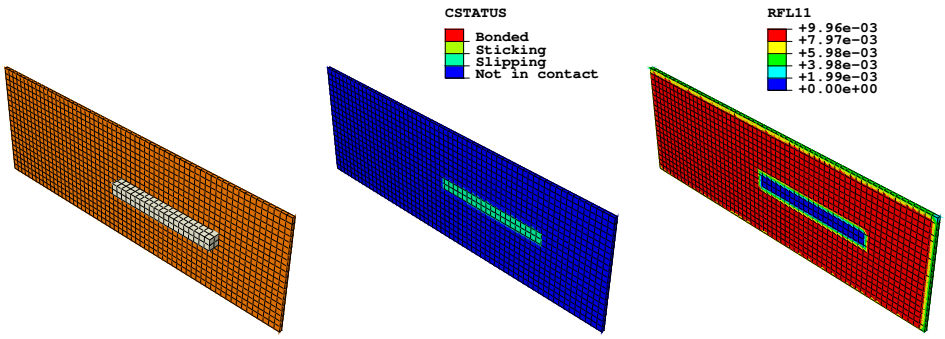
You can now have surface-based thermal radiation loads automatically suspended where contact is active in Abaqus/Explicit. In earlier releases, this functionality was available only in Abaqus/Standard.

Benefits: You can model surface-based thermal loads more accurately by accounting for shielding effects of contact on these loads.

In earlier releases, thermal radiation loads on individual surfaces associated with surface-based radiation to ambient temperature were active regardless of the contact status in Abaqus/Explicit. Now, you can model shielding of surface-based radiation to ambient temperature due to active contact.

To use the active contact status information to turn off heat exchange from surface radiation loads, you can explicitly define surface-based radiation to depend on the contact status. Alternatively, if thermal shielding due to contact is not desired, you can turn off contact status dependence for surface-based radiation.

The animation below shows two blocks in contact and maintained at a constant temperature with specified temperature boundary conditions. The blocks exchange heat with their surroundings with prescribed surface-based radiation on the exterior surfaces in a coupled temperature-displacement analysis. Active contact regions shield the thermal radiation by default and evolve as the top block moves. The shielded regions with zero reactive heat flux RFL11 also evolve accordingly.



For more information, see **SRADIATE* and *Thermal Loads*.

2025 FD03

User-Defined Electrical Current Density Values

In Abaqus/Standard, you can now define nonuniform electrical current density values using the new user subroutine `DECURRENT`.

Benefits: You can account for electrical current and its effects in a variety of scientific, industrial, and other applications.
You can define the electrical current density value as a function of position, time, element number, field variables, property and parameter tables in table collections, etc. for coupled thermal-electrical, coupled temperature-displacement electrical, coupled thermal-electrochemical, and coupled temperature-displacement electrochemical analyses.

For more information, see **DECURRENT*, **DSECURRENT*, and *Electromagnetic Loads*.

Surface Tension Loads in Abaqus/Explicit

You can now specify surface tension loads on element-based surfaces in Abaqus/Explicit.

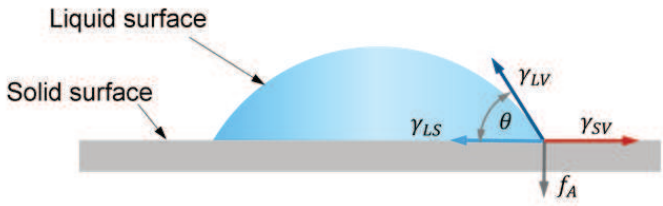
Benefits: You can account for surface tension and its effects in a variety of scientific, industrial, and medical applications.

Surface tension is a physical property of liquids that arises from cohesive forces between molecules at an interface. It behaves like an invisible, elastic membrane on the liquid surface and plays a critical role in the analysis of liquid interfaces and deformable structures in various scientific, industrial, and medical applications. In computational modeling, accurately representing surface tension is essential for capturing phenomena such as

droplet formation, capillary effects, and fluid-structure interaction for processes such as solder reflow or glass manufacturing.

Surface tension loads in Abaqus/Explicit enable you to simulate these effects in soft or fluid-like materials. You can specify the contact angle, which is defined as the equilibrium angle between the liquid-solid and liquid-vapor interfaces, to represent surface wettability characteristics that directly influence interface shape and movement.

The image below shows the equilibrium configuration of a fluid in contact with a solid. The contact angle, θ, is the equilibrium angle formed between the liquid-solid interface and the liquid-vapor interface due to the combined effect of the liquid-vapor surface tension, γ_{LV}, the solid-vapor surface tension, γ_{SV}, and the liquid-solid interfacial tension, γ_{LS}. The contact angle is a measure of surface's wettability, describing the tendency of the liquid to spread or bead up.



The animation below shows the contact of a cubic solder droplet with a pad under surface tension loads. A surface tension load is applied on each exterior face. The melted state of the solder material is modeled using a linear equation of state (EOS) material of type USUP. A no-separation, free-slip condition is enforced for the contact between the solder and pad. The contact angle θ is defined as 5 degrees.

For more information, see **DSLOAD* and *Surface Tension Loads*.

2025 FD02

Contact-Dependent Thermal Radiation Surface Loads

You can now deactivate surface-based thermal radiation loads on portions of surfaces in active contact.

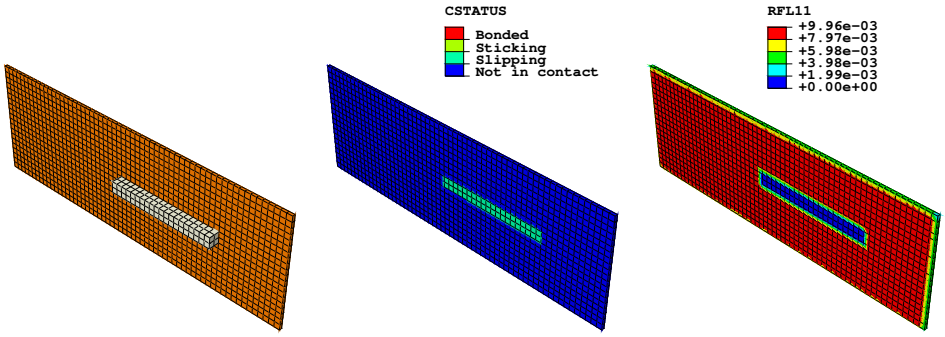
Benefits: You can model surface-based thermal loads more accurately by accounting for the effect of contact on these loads.

In earlier releases, thermal radiation loads on individual surfaces assumed that the surrounding environment was nonreflective. In this release, you can model thermal shielding on a surface due to active contact with a fully reflective surface. By default, active contact regions turn off surface thermal radiation loads altogether.

To use the active contact status information to turn off heat exchange from surface radiation loads, set the new TRANSITION parameter on **SRADIATE* equal to TOUCHING. Alternatively, if thermal shielding is not

effective, you can retain radiation heat exchange with the surroundings by setting TRANSITION=NONE to ignore active contact.

The animation below shows two blocks in contact and maintained at a constant temperature with specified temperature boundary conditions. The blocks exchange heat with surroundings with prescribed surface-based radiation on the exterior surfaces in a coupled temperature-displacement analysis. Active contact regions shield thermal radiation by default and evolve as the top block moves. The shielded regions with zero reactive heat flux RFL11 also evolve accordingly.



For more information, see **SRADIATE* and *Thermal Loads*.

Contact-Dependent Surface-Based Thermal Convection Loads in Abaqus/Explicit

You can now disable surface-based thermal convection loads automatically on portions of surfaces in active contact in Abaqus/Explicit. This functionality is previously available in Abaqus/Standard.

Benefits: You can more accurately model shielding of portions of a surface in active contact from specified thermal convection loads. In addition, it is easier to select surfaces for surface-based thermal convection loads.

Abaqus internally determines active contact regions and automatically turns off thermal convection loads on such regions. Surfaces with prescribed thermal film conditions model convective heat exchange with the surrounding gas or fluid to which they are directly exposed. When portions of these surfaces are in active contact with other surfaces, convective heat exchange with the surrounding gas or fluid is negligible due to shielding from active contact.

For more information, see **SFILM* and *Thermal Loads*.

2025 FD01

Contact-Dependent Surface-Based Thermal Convection Loads

You can now disable surface-based thermal convection loads automatically on portions of surfaces in active contact.

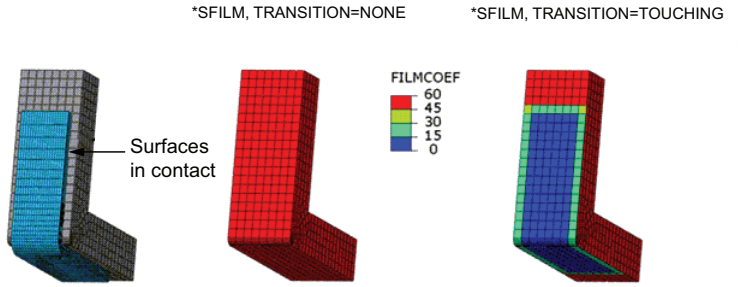
Benefits: You can more accurately model shielding of portions of a surface in active contact from specified thermal convection loads. In addition, it is easier to select surfaces for surface-based thermal convection loads.

Abaqus internally determines active contact regions and automatically turns off thermal convection loads on such regions. Surfaces with prescribed thermal film conditions model convective heat exchange with the

surrounding gas or fluid to which they are directly exposed. When portions of these surfaces are in active contact with other surfaces, convective heat exchange with the surrounding gas or fluid is negligible due to shielding from active contact.

You can set the new TRANSITION parameter equal to TOUCHING to use the active contact status information to turn off convective heat exchange. This value is the default. Alternatively, you can set TRANSITION=NO to ignore the dependence on the current contact status.

The image below shows two angle brackets in contact. Surface-based convective thermal loads are prescribed on the exterior surfaces of the brackets by specifying a uniform film coefficient value of 60.



The contour plot on the right shows a film coefficient of zero on the larger bracket in regions with active contact in order to shield convective heat exchange with the surroundings. In earlier releases, you had to select the surfaces carefully to exclude active contact regions, which is cumbersome and impractical especially when active contact regions evolve.

For more information, see **SFILM* and *Thermal Loads*.

PDF Guides Available

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

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

For more information, see *Abaqus Prescribed Conditions Guide*.

2024 FD01

Modified Interface to Define Time Variation of Ambient Temperature for Thermal Radiation

For prescribed thermal radiation boundary conditions, the new AMBIENT AMPLITUDE replaces the AMPLITUDE parameter available in earlier versions.

Benefits: The new parameter name better describes the capability, which enhances the clarity of the user interface.

The new AMBIENT AMPLITUDE parameter behaves identically to the AMPLITUDE parameter available in earlier releases. Abaqus will continue to support the AMPLITUDE parameter for some time; however, you are encouraged to replace it with the new parameter in all your simulations.

For more information, see **RADIATE*, **CRADIATE*, **SRADIATE*, and *Thermal Loads*.

Time Variation of Radiation Flux

For prescribed thermal radiation boundary conditions, you can use the new RADIATION AMPLITUDE parameter to define the time variation of the prescribed radiation flux directly through an amplitude definition.

Benefits: This capability provides additional modeling flexibility in some situations.

For more information, see **RADIATE*, **CRADIATE*, **SRADIATE*, and *Thermal Loads*.

Modified Interface to Define Time Variation of Sink Temperature for Convection Boundary Conditions

For prescribed convective (film) boundary conditions in a thermal analysis, the new SINK AMPLITUDE parameter replaces the AMPLITUDE parameter available in earlier versions.

Benefits: The new parameter name better describes the capability, which enhances the clarity of the user interface. The new SINK AMPLITUDE parameter behaves identically to the AMPLITUDE parameter available in earlier releases. Abaqus will continue to support the AMPLITUDE parameter for some time; however, you are encouraged to replace it with the new parameter in all your simulations.

For more information, see **FILM*, **CFILM*, **SFILM*, and *Thermal Loads*.

Algorithmic Improvements to Convection and Radiation Boundary Flux Calculations

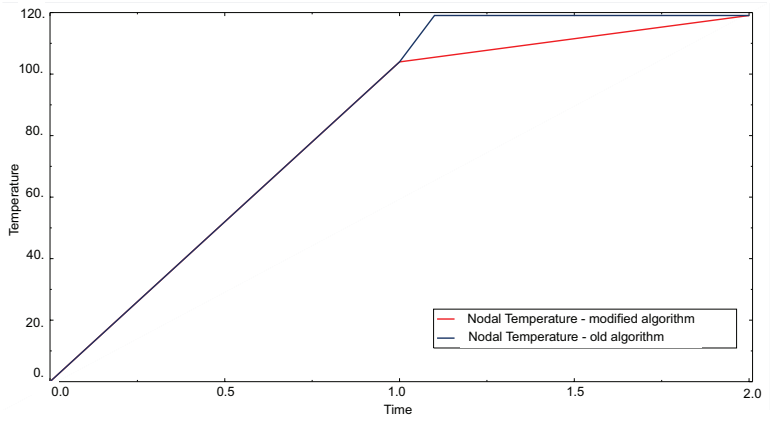
The flux calculations associated with prescribed convection and/or radiation boundary conditions in Abaqus/Standard now use an improved algorithm.

Benefits: The improved algorithms ensure continuity of convective/radiative fluxes across step boundaries, resulting in smooth temperature variations, and lead to better overall convergence characteristics for certain types of multi-step steady-state thermal analyses.

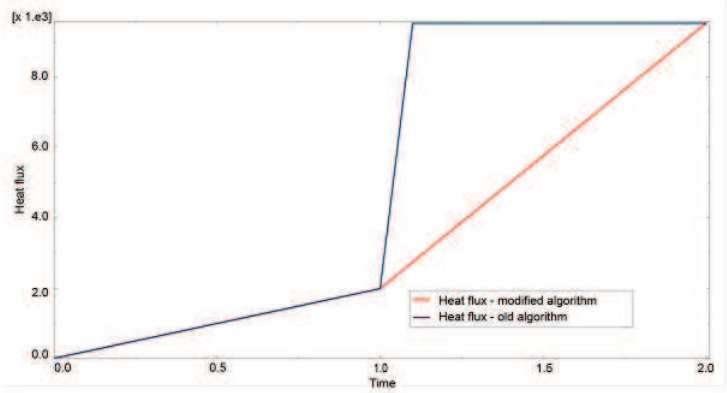
By default, in the first step of a steady-state thermal analysis earlier versions of Abaqus/Standard assumed that the sink temperature was ramped up from zero to its specified value over the step time period. In the presence of prescribed nonzero initial temperatures, this assumption led to discontinuities in the convective/radiative fluxes, as well as the temperature fields. In addition, in a multi-step analysis, the default amplitude behavior associated with film/radiation thermal loads could result in nonsmooth temperature and heat fluxes across step boundaries. Such discontinuities could also lead to convergence difficulties in nonlinear problems. The new, improved algorithm removes the default modulation of the sink temperature by the step amplitude, ensuring continuity of fluxes, smooth variations of temperature, and better overall convergence. For more information, see the Knowledge Base item QA00000307343 in the Dassault Systèmes Knowledge Base.

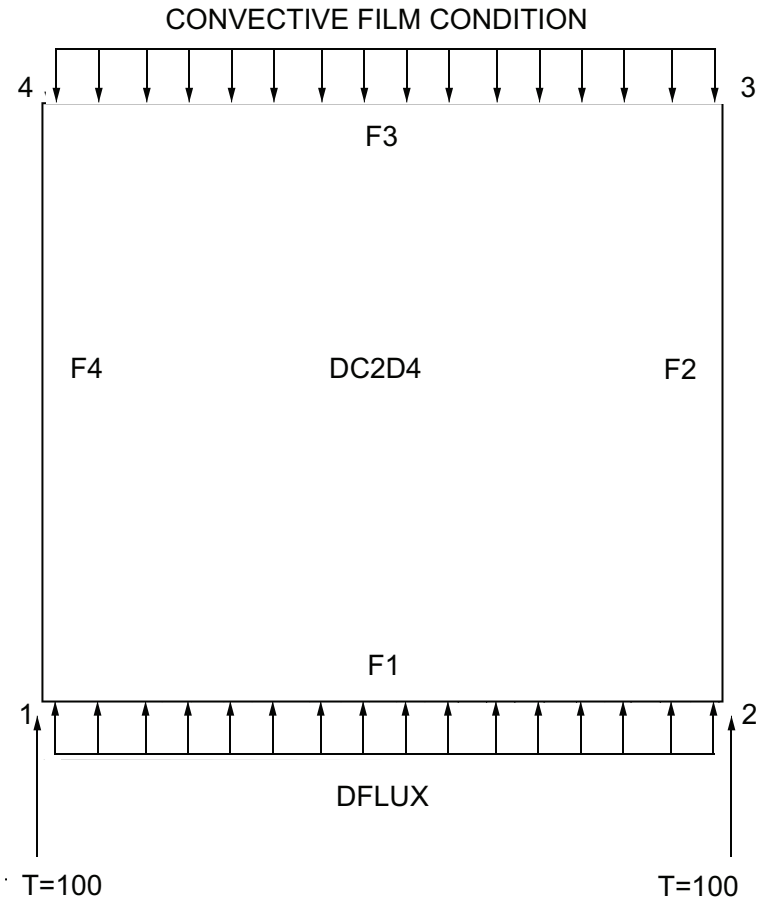
The figure below shows a one-element test with prescribed temperatures at the bottom nodes, a prescribed heat flux along the bottom surface, and a prescribed convective (film) boundary condition on the top surface. There are two steady-state heat transfer steps in the analysis, each with a time period of 1.0 (for a total time period of 2.0).

The plot of temperature versus time below contrasts the temperatures across step transitions using the old and the improved (modified) algorithms.



The plot of convective heat flux versus time below contrasts the convective heat fluxes across step transitions using the old and the improved (modified) algorithms.





In the old formulation, the convective heat flux experienced a discontinuity at the transition between the two steps. The improved formulation leads to a smoother behavior of the convective heat flux over the second step.

For more information, see **FILM*, **CFILM*, **SFILM*, **RADIATE*, **CRADIATE*, **SRADIATE*, and *Thermal Loads*.

Scripting

This page describes recent changes in Abaqus Scripting.

2025 FD01

PDF Guides Available

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

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

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

2024 GA

Abaqus Python Upgrade to Version 3.10.5

Abaqus Python is updated to Version 3.10.5.

Benefits: Customers and partners can leverage the latest functionality provided in Python 3.

AbaqusPython has been updated from Python version 2.7.15 to Python version 3.10.5.

Python 3 introduces a number of changes that break backward-compatibility with existing Python 2 code. As a result, scripts originally written using Python 2 syntax might not run without modification in Python 3. However, in most cases it is possible to modify your scripts so that they will run in both Python 2.7 and Python 3.

The following Abaqus functionality is impacted:

- Abaqus/CAE and Abaqus/Viewer scripting
- Abaqus/CAE and Abaqus/Viewer plugins
- The `abaqus_v6.env` environment file (see *Using the Abaqus environment files*)
- Custom applications (see *Creating a customized application*)
- Standalone Abaqus Python scripts; for example, scripts run with `abaqus python myscript.py` (see *Python Execution*)
- **PARAMETER* (see *Parametric Modeling*)
- Parametric studies—`.psf` files (see *Parametric Studies*)
- Batch scheduling scripts (Dassault Systèmes Knowledge Base, *https://support.3ds.com/knowledge-base*)

A conversion utility is provided to assist with converting Python 2 syntax to Python 3. While it will automatically convert many common syntax changes, not all Python 2 syntax can be converted automatically; therefore, some manual conversion might be required to ensure that Python scripts run properly with the new version. You can invoke the conversion utiility from the command line using the command

```
abaqus python -m abqPy2to3 <script/directory>
```

The utility writes a log file and saves the original versions of the files on which it operates. If you provide a directory to the utility, it recursively traverses through that directory and subdirectories looking for `*.py` files.

You can include `-future False` on the command line if you are not worried about maintaining Python 2.7 forward compatibility. You can also access the conversion utility as part of the script upgrade plug-in utility in Abaqus/CAE (available from **Plug-Ins > Abaqus > Upgrade Scripts**). It is run if you upgrade scripts to 2024.

Additional information related to converting Python code for Abaqus is available in the SIMULIA Community at *go.3ds.com/AbaqusPython*.

Scripting Reference

This page describes recent changes in Abaqus Scripting Reference.

2026 FD01

Python Commands

Revised Python objects are available.

Benefits: The features extend the capabilities of the Python commands.

Table 2: Revisions to Python Command Objects

Command	Object	Revisions
Mesh	*ElemType object*	ElemType now accepts parameters for controlling uniform background grid conversion of continuum elements to particles, for smoothed particle hydrodynamics (SPH) analyses in Abaqus/Explicit.

2026 GA

Python Commands

New Python objects are available.

Benefits: The features extend the capabilities of the Python commands.

Table 3: New Python Command Objects

Command	Object
Interaction	*GapDiffusivity object*
Material	*Electrode object*
	Electrode Particle object
	Electrode Particle Layer object
	Electrolyte object
Step (step)	*CoupledThermalElectrochemicalStructuralStep object*
	CoupledThermalElectrochemStructPorepressureStep object

Table 4: Revisions to Python Command Objects

Command	Object	Revisions
Step (miscellaneous)	*StepControl object*	The optional argument **tolerance** is added.
Interaction	*ContactProperty object*	Member **gapDiffusivity** is added.

2025 FD02

Python Commands

New and revised Python objects are available.

Benefits: The features extend the capabilities of the Python commands.

Table 5: New Python Command Objects

Command	Object
Boundary Condition	*ElectricPotentialElectrolyteBC object*
Boundary Condition	*ElectricPotentialElectrolyteBCState object*
Boundary Condition	*IonConcentrationBC object*
Boundary Condition	*IonConcentrationBCState object*
Load and Load Case	*DistIonConcBodyFlux object*
Load and Load Case	*DistIonConcBodyFluxState object*
Load and Load Case	*FluidSurfaceCurrent object*
Load and Load Case	*FluidSurfaceCurrentState object*
Load and Load Case	*IonConcentrationFluxState object*
Predefined field	*FluidElectricPotential object*
Predefined field	*IonConcentration object*
Step (step)	*CoupledThermalElectrochemicalStep object*
Step (miscellaneous)	*StepControl object*
Step (miscellaneous)	*CoupledThermalElectrochemicalStep object*

Table 6: Revisions to Python Command Objects

Command	Object	Revisions
Constraint	*Coupling object*	The required argument **coupling*Type*** can now take the value of UNIFORM, and the new optional arguments **uniformConstraint** and **temperatureDOFType** are added.

Command	Object	Revisions
Constraint	*Tie object*	Optional arguments **tieTemperature**, **tiePore**, **electricPotential**, **fluidElectricPotential**, and **ionConcentration** are now available.
Interaction	*FilmCondition object*	Optional argument **dynamicSurfaceActivation** is now available.
Interaction	*GapElectricalConductance object*	Optional arguments **Type**, **ionConcentrationDependencyC**, and **ionConcentrationDependencyP** are now available.
Interaction	*GapHeatGeneration object*	Optional arguments **mechanicalInteractionFraction**, **mechanicalInteractionWeightingFactor**, **electricalConductionSolidsFraction**, **electricalConductionSolidsWeightingFactor**, **electricalConductionLiquidsFraction**, and **electricalConductionLiquidsWeightingFactor** are now available.

2025 FD01

Animation File Formats

You can use the new MP4Options and GIFOptions objects in the Animation commands to save and load animations in MP4 and animated GIF formats.

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

For more information, see *GIFOptions object*, *MP4Options object*, *Choosing the animation file format*, and *Using background images and movies*.

PDF Guides Available

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

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

For more information, see *Abaqus Scripting Reference Guide*.

2025 GA

Python Interactions Commands: FluidExchangeActivation Object

The FluidExchangeActivation object is now available in the Interactions profile commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *FluidExchangeActivation object*.

Python Interactions Commands: FluidExchangeActivationState Object

The FluidExchangeActivationState object is now available in the Interactions profile commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *FluidExchangeActivationState object*.

Python Interactions Commands: FluidInflator Object

The required arguments inflationTimeAmplitude and massFlowAmplitude are no longer available in the FluidInflator object.

Benefits: The functionality was extended, with these arguments appropriately available in the FluidInflatorActivation object instead.

For more information, see *FluidInflator object*.

Python Interactions Commands: FluidInflatorActivation Object

The FluidInflatorActivation object is now available in the Interactions profile commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *FluidInflatorActivation object*.

Python Interactions Commands: FluidInflatorActivationState Object

The FluidInflatorActivationState object is now available in the Interactions commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *FluidInflatorActivationState object*.

Python Interactions Commands: WearProperty Object

The WearProperty object is now available in the Interactions commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *WearProperty object*.

Python Interactions Commands: WearSurfacePropertyAssignment Object

The WearSurfacePropertyAssignment object is now available in the Interactions commands.

Benefits: The feature extends the capabilities of the Interactions commands.

For more information, see *WearSurfacePropertyAssignment object*.

Python Load and Load Case Commands: RotationalBodyForce Object

The new optional argument rotorDynamicloads is available in the RotationalBodyForce object.

Benefits: The feature extends the capabilities of the Load and Load Case commands.

For more information, see *RotationalBodyForce object*.

Python Path and Probe Commands: FreeBody Objects

The method getFreeBodyData() is now available in the FreeBody objects.

Benefits: The feature extends the capabilities of the Path and Probe commands.

For more information, see *FreeBody object*.

2024 FD03

New Method getMemberFromAll for Node and Element Sequences

There is a new Python API on node and element sequences to retrieve a member value for each item in the sequence. This applies to MeshNodeArray, MeshElementArray, MeshFaceArray, ODBMeshNodeArray, and ODBMeshElementArray.

Benefits: Retrieving a member from a large sequence (for example, the node label for every node) is much faster using this method than iterating over the sequence.

For more information, see *MeshNodeArray object*, *MeshElementArray object*, *MeshFaceArray object*, *OdbMeshNode object*, and *OdbMeshElement object*.

2024 GA

Python 3

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

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

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

Python Beam Section Profile Commands: ChannelProfile Object

The ChannelProfile object is now available in the Beam Section profile commands.

Benefits: The feature extends the capabilities of the Beam Section profile commands.

For more information, see *ChannelProfile object*.

Python Beam Section Profile Commands: HatProfile Object

The HatProfile object is now available in the Beam Section profile commands.

Benefits: The feature extends the capabilities of the Beam Section profile commands.

For more information, see *HatProfile object*.

Python Constraint Commands: Coupling Object

The Coupling object is now available in the Constraint commands.

Benefits: The feature extends the capabilities of the Constraint commands.

For more information, see *Coupling object*.

Python Engineering Feature Commands: DiscreteFastener Object

The optional argument *rotatonalCoupling* is now available in the DiscreteFastener object.

Benefits: The feature extends the capabilities of the Engineering Feature commands.

For more information, see *DiscreteFastener object*.

Python Interaction Commands: ContactMassScalingExp Object

The ContactMassScalingExp object is now available in the Interaction commands.

Benefits: The feature extends the capabilities of the Interaction commands.

For more information, see *ContactMassScalingExp object*.

Python Job Commands: OptimizationProcess Object

The optional argument licenseType is now available in the OptimizationProcess object.

Benefits: The feature extends the capabilities of the Job commands.

For more information, see *ContactMassScalingExp object*.

Python Odb Commands: OdbAssembly Object

The method getMassProperties is now available in the OdbAssembly object.

Benefits: The feature extends the capabilities of the Odb commands.

For more information, see *OdbAssembly object*.

Python Optimization Commands: SingleTermDesignResponse Object

The optional argument stepOperation is no longer available in the SingleTermDesignResponse object.

Benefits: The functionality was removed from the Optimization commands: SingleTermDesignResponse object.

For more information, see *SingleTermDesignResponse object*.

Python Section Commands: BeamSection Object

The optional argument beamSectionOffset is now available in the BeamSection object.

Benefits: The feature extends the capabilities of the Section commands.

For more information, see *BeamSection object*.

Python Step Commands (Step): StaticLinearPerturbationStep Object

The optional arguments solutionTechnique, gapDistance, and scaleFactor are now available in the StaticLinearPerturbationStep object.

Benefits: The feature extends the capabilities of the Step commands (step).

For more information, see *StaticLinearPerturbationStep object*.

Python XY Commands: Session Object

The optional argument useMaxStressValue is now available in the Session object.

Benefits: The feature extends the capabilities of the XY commands.

For more information, see *Session object*.

Theory

This page describes recent changes in Abaqus Theory.

2025 FD01

PDF Guides Available

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

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

For more information, see *Abaqus Theory Guide*.

2024 FD01

Coupled Acoustic-Structural Medium Analysis

The theory behind the use of secondary output quantities in acoustic analysis is discussed.

Benefits: The discussion expands the definition of the theories used in Abaqus.

For more information, see *Coupled acoustic-structural medium analysis*.

User Subroutines

This page describes recent changes in Abaqus User Subroutines.

2026 FD04

Obtaining the Cap Position for Cap Plasticity Materials

You can use utility subroutines `GETVRM` and `GETVRMAVGATNODE` in Abaqus/Standard and `VGETVRM` in Abaqus/Explicit to obtain the cap position (PBCAP) for materials with cap plasticity.

Benefits: Using the new output variable provides a meaningful label for identifying the cap position in the cap plasticity model.

For more information, see *Obtaining Material Point Information in an Abaqus/Standard Analysis*, *Obtaining Material Point Information Averaged at a Node*, and *Obtaining Material Point Information in an Abaqus/Explicit Analysis*.

Rate-Dependent and Rate-Independent Responses of Rate-Dependent User-Defined Materials

In Abaqus/Explicit, you can use user subroutines `VUCREEPNETWORK`, `VUHARD`, and `VUMAT` to define both the rate-dependent and rate-independent responses of rate-dependent user-defined materials.

Benefits: You can run explicit dynamic simulations with rate-dependent or rate-independent material response without having to modify the material definition.

For more information, see *VUCREEPNETWORK*, *VUHARD*, and *VUMAT*.

2026 FD03

`VUHYPER_STRETCH` (new)

You can use user subroutine `VUHYPER_STRETCH` to define stretch-based hyperelastic models in Abaqus/Explicit.

Benefits: The user subroutine extends the functionality of Abaqus/Explicit, enabling you to define stretch-based hyperelastic models that are not otherwise available in Abaqus.
You can now model hyperelastic material behavior in Abaqus/Explicit using user subroutine `VUHYPER_STRETCH` to define the stretch-based strain energy function. Equivalent functionality is available in Abaqus/Standard with user subroutine `UHYPER_STRETCH`.

`VUHYPER_STRETCH` can be used in combination with other material behaviors, such as linear and nonlinear viscoelasticity. When used with the parallel rheological framework, the subroutine has access to the network identifier (`networkID`), which allows the definition of different hyperelastic models for different nonlinear viscoelastic networks.

For more information, see *VUHYPER_STRETCH*.

User-Defined Output Variables at Beam Elements

You can use user subroutine UVAREL to define output variables at beam elements in Abaqus/Standard. In earlier releases, this functionality was available only for continuum and shell elements.

Benefits: The user subroutine makes it easier for you to compute, create, and elaborate user-defined element output variables and gives you the flexibility to implement a broad variety of data consolidation schemes that can reduce the number of results that must be stored in the output database.
For beam elements, you can also invoke the following new utility routines from user subroutine UVAREL:

- getElMPtVar returns material point information;
- getSecVar returns integration point information;
- getUVarElOrientationName returns the name of the element orientation; and
- getUVarElMaterialName returns the name of the material.

For more information, see *UVAREL*, *Obtaining Material Point Information from User Subroutine UVAREL*, and *Obtaining Element Section Information from User Subroutine UVAREL*.

2026 FD02

Strain Tensors for User-Defined Design Response and Sensitivities in UELEMDRESP

User subroutine UELEMDRESP now allows definition of the user-defined design response and corresponding sensitivities using the strain tensor, the logarithmic strain tensor, or the nominal strain tensor.

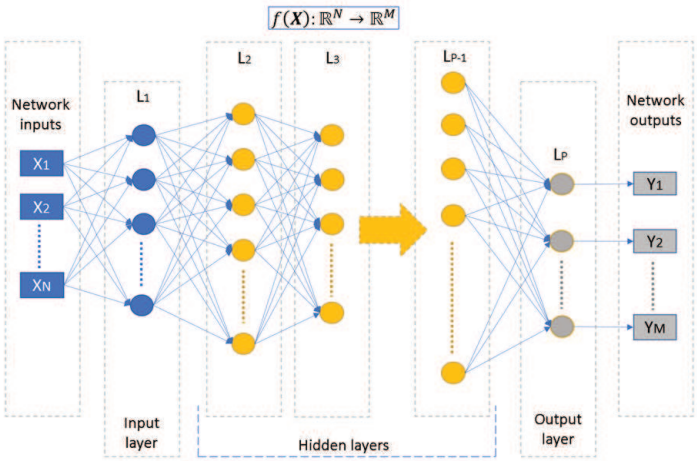
Benefits: User-defined strain-based failure criteria provide improved flexibility for evaluating the sensitivity of allowable responses in design and optimization workflows.
You can now define a user-defined failure criterion (as a design response) using the strain tensor, the logarithmic strain tensor, or the nominal strain tensor using user subroutine UELEMDRESP. This new functionality is similar to the user-defined design responses using the stress tensor or the plastic strain tensor available in earlier releases. You can use UELEMDRESP to calculate the strain-based user-defined design responses for static, quasi-static, and implicit transient dynamic analyses.

For more information, see *UELEMDRESP*.

Evaluating Feed-Forward Neural Networks During Simulations Using Abaqus/Standard

You can now use specialized utility routines in Abaqus/Standard to evaluate feed-forward trained neural networks that were defined in the input file.

Benefits: This new functionality bridges advanced machine learning capabilities with finite element analysis, allowing you to dynamically incorporate trained feed-forward neural network predictions—such as constitutive material responses, complex boundary conditions, or load definitions—into simulations.
You can now define trained feed-forward (dense) neural networks in the input file. In a single input file, you can define and use multiple neural networks. Each neural network can have an arbitrary configuration comprised of different numbers of layers, with each layer having a different size and a different activation function. You can then call the neural network evaluation utility routines to obtain the neural network output and the output derivatives with respect to the input vector. The figure below displays a feed-forward neural network with P layers, mapping an n-dimensional input vector to an m-dimensional output vector.



For more information, see *About Neural Networks* and *Evaluating Abaqus Neural Networks*.

2026 FD01

C/C++ Interfaces to Abaqus Utility Routines

You can now invoke Abaqus utility routines from user subroutines written in C/C++ without any Fortran-CC interoperability issues.

Benefits: This extends the functionality of Abaqus/Standard.

In earlier releases, only Fortran interfaces were available. Now you can invoke these utility routines directly from C/C++ user subroutines without any Fortran-C interoperability issues.

For more information, see *Accessing Abaqus Table Collections*, *Obtaining Material Point Information from User Subroutine UVAREL*, and *Obtaining Material Point Information in an Abaqus/Standard Analysis*.

UHYPER

You can now access the values of the solution-dependent state variables at the beginning of the increment in user subroutine `UHYPER`.

Benefits: This extends the functionality of Abaqus/Standard.

The values of the solution-dependent state variables at the beginning of the increment can now be accessed in user subroutine `UHYPER` in the `STATEV(*,2)` array. This allows you to use those values in the definition of your user-defined isotropic hyperelastic materials. In earlier releases, only the current values of the solution-dependent state variables (now in the `STATEV(*,1)` array) were accessible.

For more information, see *UHYPER*.

2026 GA

VUTRSNETWORK (new)

You can use user subroutine VUTRSNETWORK to define a temperature-time shift for a nonlinear viscoelastic network for models defined using the parallel rheological framework in Abaqus/Explicit.

Benefits: The user subroutine extends the functionality of Abaqus/Explicit, enabling you to define forms of temperature-time shifts that are not otherwise available in Abaqus.
You can now model temperature-time shift in Abaqus/Explicit using user subroutine VUTRSNETWORK to define the form of the temperature-time shift. This functionality is equivalent to user subroutine UTRSNETWORK previously available in Abaqus/Standard. The VUTRSNETWORK subroutine has access to the network identifier (networkID), which allows the definition of different forms of temperature-time shifts for different nonlinear viscoelastic networks.

For more information, see *VUTRSNETWORK*.

2025 FD03

Electrochemical Procedures

You can use new user subroutines in electrochemical analysis.

Benefits: You can simulate a variety of complex electrochemical effects for battery simulations in Abaqus/Standard.

DECURRENT	Define electrical current magnitude as a function of position, time, element number, field variables, etc. for coupled thermal-electrical, coupled temperature-displacement electrical, coupled thermal-electrochemical, and coupled temperature-displacement electrochemical analyses
UCOUPLEDCONSTRESP	Define a fully coupled constitutive response based on the different physics that is being modeled
UECHEMECDBV	Define the Butler-Volmer intercalation current to be used for elements that utilize porous electrode theory in an electrochemical analysis
UECHEMECDDISS	Define the electrode dissolution aging current to be used for elements that utilize porous electrode theory in an electrochemical analysis
UECHEMECDLPL	Define the lithium plating (LPL) aging current to be used for elements that utilize porous electrode theory in an electrochemical analysis
UECHEMECDSEI	Define the Solid Electrolyte Interface (SEI) aging current to be used for elements that utilize porous electrode theory in an electrochemical analysis
UECHEMECDSEIREFORM	Define the Solid Electrolyte Interface (SEI) reformation current to be used for elements that utilize porous electrode theory in an electrochemical analysis
UEIGENSTRAIN	Define the eigenstrain in a material from different physics fields
UEREF	Define the Open Circuit Potential (OCP) to be used for elements that utilize porous electrode theory in an electrochemical analysis

UINTFLUXGEN	Define the internal heat generation in a material from different physics fields

For more information, see *DECURRENT*, *UCOUPLEDCONSTRESP*, *UECHEMECDBV*, *UECHEMECDDISS*, *UECHEMECDLPL*, *UECHEMECDSEI*, *UECHEMECDSEIREFORM*, *UEIGENSTRAIN*, *UEREF*, and *UINTFLUXGEN*.

2025 FD01

UHYPER

You can now use user subroutine UHYPER to define different hyperelastic models for different viscoelastic networks when the subroutine is used with the parallel rheological framework.

Benefits: The user subroutine extends the functionality of Abaqus/Standard.

The network identification number (networkID) is now passed as a parameter to the user subroutine UHYPER when the subroutine is used with the parallel rheological framework. This allows you to define different hyperelastic models for different nonlinear viscoelastic networks. In earlier releases, this capability was available only with element overlay.

For more information, see *UHYPER*.

UHYPER_STRETCH

You can now use user subroutine UHYPER_STRETCH to define different hyperelastic models for different viscoelastic networks when the subroutine is used with the parallel rheological framework.

Benefits: The user subroutine extends the functionality of Abaqus/Standard.

The network identification number (networkID) is now passed as a parameter to user subroutine UHYPER_STRETCH when the subroutine is used with the parallel rheological framework. This allows you to define different hyperelastic models for different nonlinear viscoelastic networks. In earlier releases, this capability was available only with element overlay.

For more information, see *UHYPER_STRETCH*.

VUHYPER (new)

You can use user subroutine VUHYPER to define hyperelastic material models in Abaqus/Explicit.

Benefits: The user subroutine extends the functionality of Abaqus/Explicit, enabling you to define strain energy functions that are not otherwise available in Abaqus.
You can now model hyperelastic material behavior in Abaqus/Explicit using user subroutine VUHYPER to define the strain energy function. This functionality is equivalent to user subroutine UHYPER previously available in Abaqus/Standard. VUHYPER can be used in combination with other material behaviors, such as linear and nonlinear viscoelasticity. When used with the parallel rheological framework, the subroutine has access to the network identifier (networkID), which allows the definition of different hyperelastic models for different nonlinear viscoelastic networks.

For more information, see *VUHYPER*.

PDF Guides Available

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

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

For more information, see *Abaqus User Subroutines Guide*.

2024 FD03

`UVAREL` (new)

You can use the new user subroutine `UVAREL` to define output variables at the element and integration point level in Abaqus/Standard.

Benefits: The user subroutine makes it easier for you to compute, create, and elaborate user-defined element output variables and provides you with the flexibility to implement a broad variety of data consolidation schemes that can reduce the number of results that must be stored in the output database.

User subroutine `UVAREL` gives you a method to define variables at the element level to reduce the amount of output data if the results at every material point are not needed. The reduction of output data can be significant, especially for composite elements with many layers.

The user-defined element output variables that are defined in user subroutine `UVAREL` usually depend on other Abaqus output variables such as stresses, strains, and section forces. You can access these output quantities by calling the new utility routines `getElMPtVar` and `getSecVar`.

You can invoke the following new utility routines from user subroutine `UVAREL`:

- `getElMPtVar` returns material point information
- `getSecVar` returns integration point information
- `getUVarElOrientationName` returns the name of the element orientation
- `getUVarElMaterialName` returns the name of the material

For more information, see *UVAREL*, **ELEMENT USER OUTPUT VARIABLES*, *Obtaining Material Point Information from User Subroutine UVAREL*, *Obtaining the Element Orientation Name from User Subroutine UVAREL*, *Obtaining the Material Name from User Subroutine UVAREL*, *Obtaining Element Section Information from User Subroutine UVAREL*, and *UVAREL*.

User Subroutine and Utilities Assistance Consolidated in this Guide

Conceptual information about user subroutines and utilities available in Abaqus is now consolidated in this guide.

Benefits: Moving the content from the *Abaqus Analysis Guide* makes it easier for you to access the information as you review the reference topics in this guide.

For more information, see *About User Subroutines and Utilities*, *About Table Collections, Parameter Tables, and Property Tables*, and *About Event Series*.

2024 FD02

VDLOAD

You can now access displacement and acceleration information in user subroutine VDLOAD.

Benefits: Accessing the displacements and accelerations information in the user subroutine is useful when the distributed loads are a function of the displacements and accelerations.

The array velocity available in earlier releases in the VDLOAD argument list is now named vua. It is also extended to include velocities, displacements, and accelerations, giving you the option to define distributed loads as function of these quantities with user subroutine VDLOAD. This change does not affect existing VDLOAD implementations based on the interface available in earlier releases.

For more information, see *VDLOAD*.

2024 FD01

UEXPAN

You can now define field expansion strains and, in a porous medium, pore fluid field expansion strains directly in user subroutine UEXPAN.

Benefits: Defining the values in the user subroutine is particularly useful when the field expansion strains are complicated functions of temperature, field variables, or state variables.

The argument list of user subroutine UEXPAN has been enhanced to provide additional information that you can use to discern if the call corresponds to a thermal or a field expansion definition, the field number and, in the case of a porous medium, if the call corresponds to the expansion definition for the pore fluid. This change does not affect existing UEXPAN implementations based on the interface available in earlier releases.

For more information, see *UEXPAN* and *UEXPAN*.

Verification

This page describes recent changes in Abaqus Verification.

2026 FD04

Contact Surface Representation of Threads in Abaqus/Standard

New tests verify the internally generated surface-element representation of threads.

Benefits: The tests verify the capability in Abaqus/Standard.

The new capability improves contact simulation accuracy involving relatively large twisting motion between the external and the internal threads.

For more information, see *Contact surface representation of threads*.

Abaqus/Explicit: Cohesive Contact Involving Large Cohesive Strains

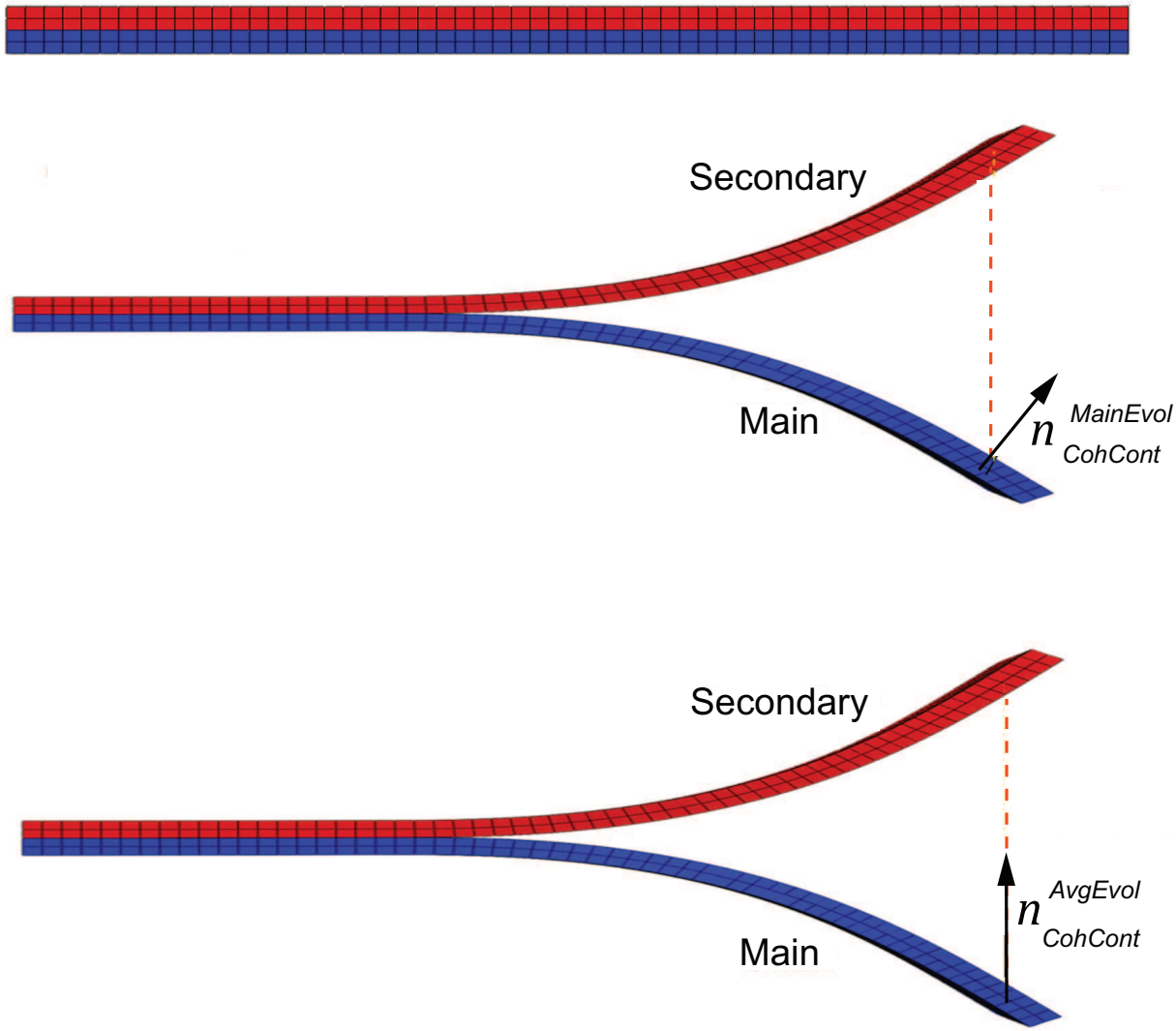
New tests verify cohesive contact behavior involving highly ductile cohesive bonds.

Benefits: The tests verify the capability in Abaqus/Explicit.

The figure below shows the initial and deformed configurations for a peeling test simulation with ductile cohesive contact bonds. The two parts are initially flat (top image) with cohesive contact along the entire length of the interface. Symmetry exists in the mesh refinement and loading conditions.

In the case shown in the middle image, the cohesive contact normal direction evolves with the main surface only, such that the cohesive strain develops normal and shear components. The dotted line in the image connects interacting points of a cohesive constraint.

In the case shown in the bottom image, the cohesive strain is aligned with the cohesive contact normal direction, which evolves taking changes in both the main and secondary surface normal directions into consideration. There is zero cohesive shear strain in this case.



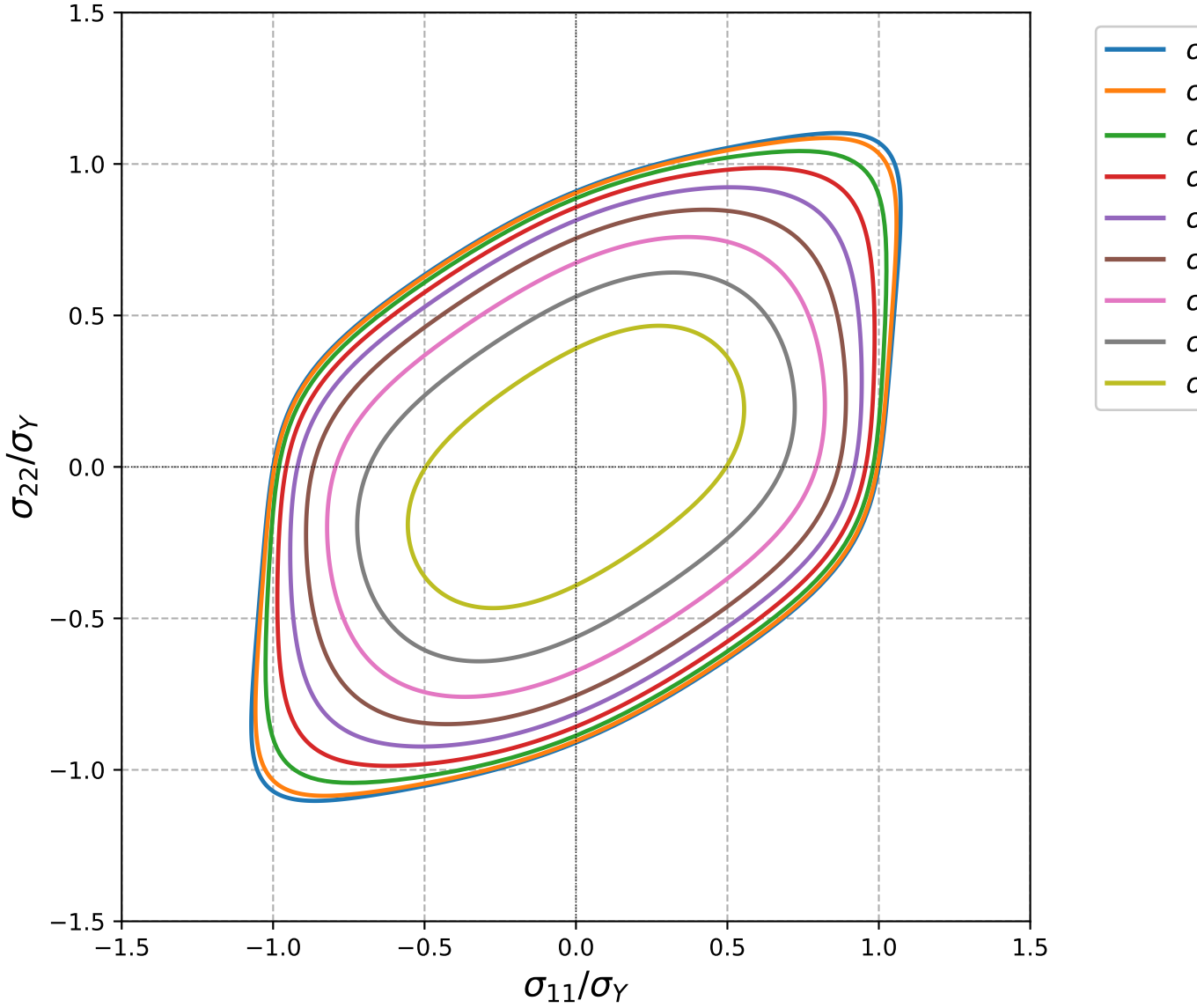
For more information, see *Cohesive surface interaction*.

Barlat Anisotropic Plasticity

New tests verify the Barlat plasticity models.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

New tests are added to verify the modeling of Barlat Yld2004-18p and Barlat Yld91 yield surfaces using plane stress elements. Tests are also added to verify the new Barlat Yld2000-2d model. The plot below shows the yield surface for the 2090-T3 material predicted with the Barlat Yld2000-2d model.



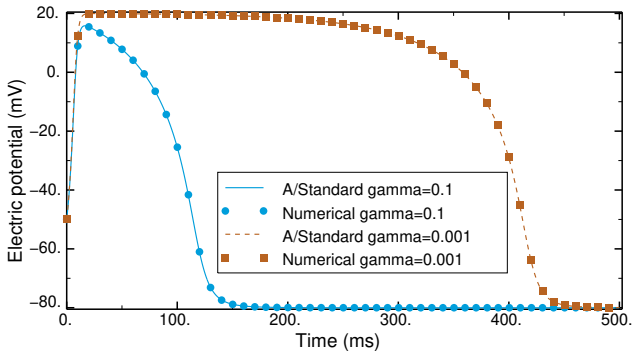
For more information, see *Barlat anisotropic plasticity*.

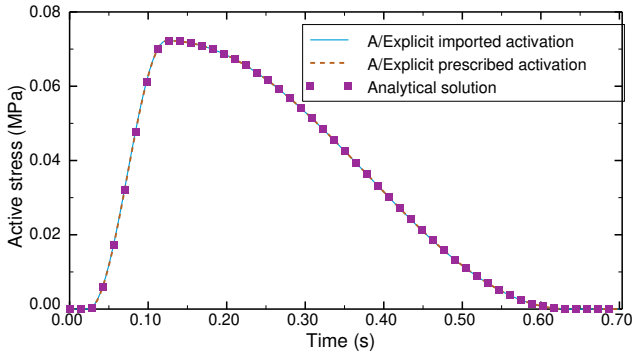
Electrical and Mechanical Modeling of Cardiac Tissue

New tests verify the electrical modeling of cardiac tissue in Abaqus/Standard and its mechanical modeling in Abaqus/Explicit.

Benefits: The tests verify the capabilities in Abaqus/Standard and Abaqus/Explicit.

The figures below show the agreement of the Abaqus/Standard and Abaqus/Explicit results with available numerical or analytical solutions.





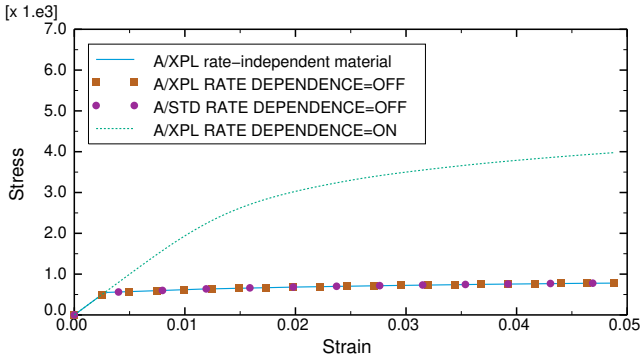
For more information, see *Electrical and mechanical modeling of cardiac tissue*.

Control of Material Rate-Dependence

New tests verify the control of material rate-dependence when running simulations in Abaqus/Standard and Abaqus/Explicit.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The figure below shows the agreement of the simulations ignoring material rate-dependent behaviors with reference simulations where the corresponding material definitions do not contain rate-dependent properties.



For more information, see *Controlling material rate dependence*.

Surface Tension Loads with Eulerian Elements

New tests demonstrate the application of surface tension loads with Eulerian elements.

Benefits: The tests verify the capabilities in Abaqus/Explicit.

For more information, see *Surface tension loads*.

Multiple Linear Load Cases in Submodel Analysis

A new test verifies that you can use multiple linear load cases in a submodel analysis.

Benefits: The test verifies the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Using Multiple Linear Load Cases in a Submodel* and *General submodeling*.

Time Synchronization in General Submodeling

New tests verify the default time synchronization between the global model and submodel.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Referring to the Step and Increment of the Global Model* and *General submodeling*.

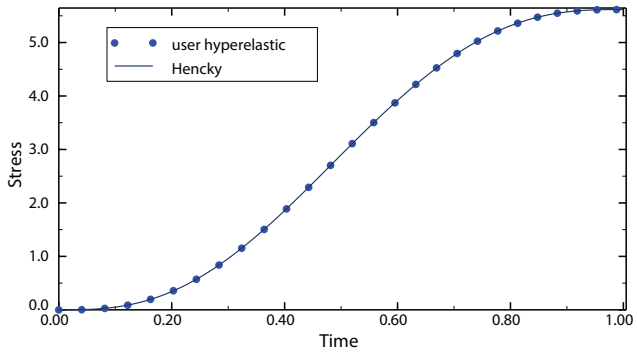
2026 FD03

`VUHYPER_STRETCH`

A new test is added to verify the user-defined stretch-based hyperelastic material model in Abaqus/Explicit.

Benefits: The test verifies the capabilities of `VUHYPER_STRETCH` in Abaqus/Explicit.

Stress results obtained from `VUHYPER_STRETCH` are validated against the solution using the Hencky hyperelastic material.



For more information, see *VUHYPER_STRETCH*.

Trajectory-Based Eigenstrain Technique for Additive Manufacturing Simulations

New verification tests demonstrate the application of eigenstrains during element activation in additive manufacturing–related simulations using the trajectory-based eigenstrain technique.

Benefits: The new tests verify the capabilities in Abaqus/Standard.

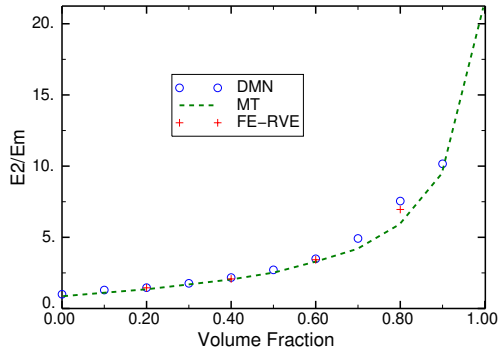
For more information, see *Trajectory-based eigenstrain analysis*.

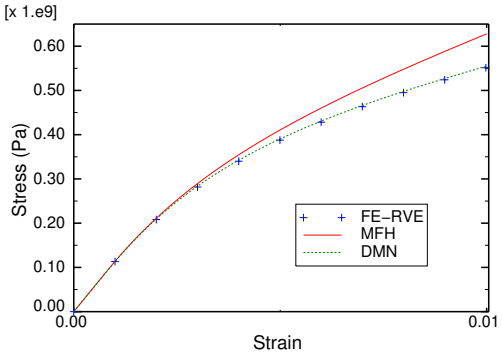
Multiscale Materials

New tests are added to verify the deep material networks (DMNs) for multiscale material modeling.

Benefits: The new tests verify the capabilities in Abaqus/Standard.

The new tests validate the DMN predictions against both FE-RVE simulations and mean-field homogenization models, spanning both linear and nonlinear composite behaviors.





For more information, see *Multiscale linear elastic materials*.

2026 FD02

ABD Stiffness Sensitivities in Adjoint Sensitivity Analyses

A new test verifies the ABD stiffness sensitivity calculations.

Benefits: The test verifies the capability in Abaqus/Standard.
The ABD stiffness sensitivities computed by Abaqus/Standard are compared to corresponding results obtained using the finite difference method.

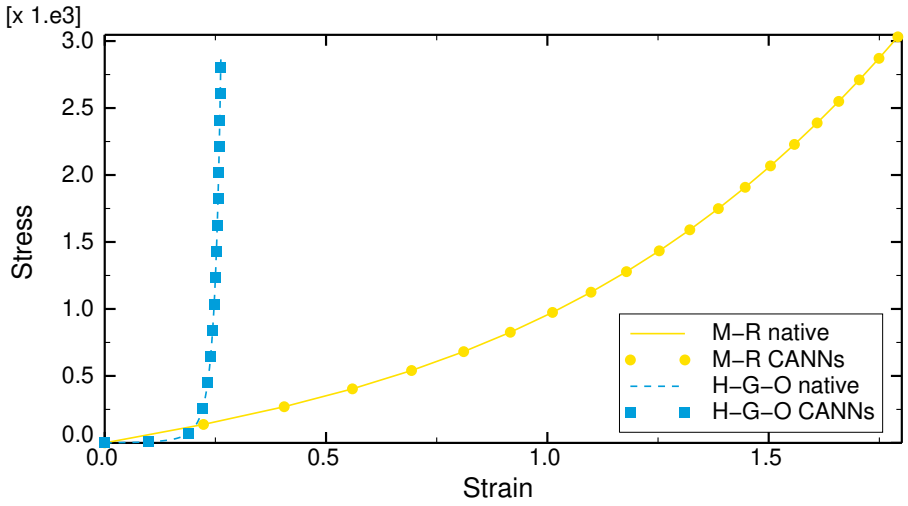
For more information, see *Adjoint Design Sensitivity Analysis Overview*, *Specifying Design Variables*, and *Adjoint design sensitivity analysis for static procedures*.

Constitutive Artificial Neural Networks (CANNs) Models

New tests verify the capability to define anisotropic hyperelastic models using constitutive artificial neural networks (CANNs) models.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The figure below shows the stress-strain response of a Mooney-Rivlin material and a Holzapfel-Gasser-Ogden material. The results for these models implemented as CANNs models agree with the results for their respective native implementations.



For more information, see *Constitutive Artificial Neural Networks (CANNs) models*.

Feed-Forward Neural Networks Evaluation

A new test verifies the capability of defining and evaluating multiple feed-forward neural networks.

Benefits: The test verifies the capability in Abaqus/Standard.
The new test demonstrates how to define and evaluate feed-forward neural networks from Abaqus/Standard user subroutines.

For more information, see *Activation functions used by feed-forward neural networks*.

Discrete Particle Swelling

A new test that verifies the evolution of the particle radius against a first-order model that is fitted to experimental data.

Benefits: The test verifies the capability in Abaqus/Explicit.

For more information, see *Discrete particle swelling*.

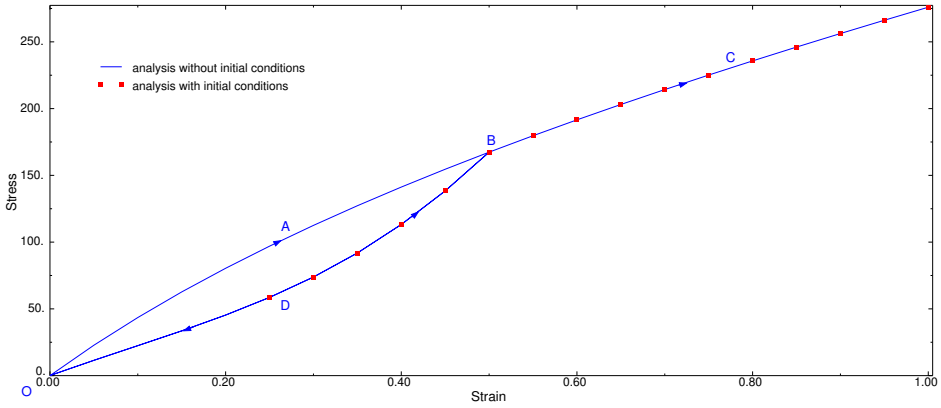
2026 FD01

Initial Stresses with Mullins Effect

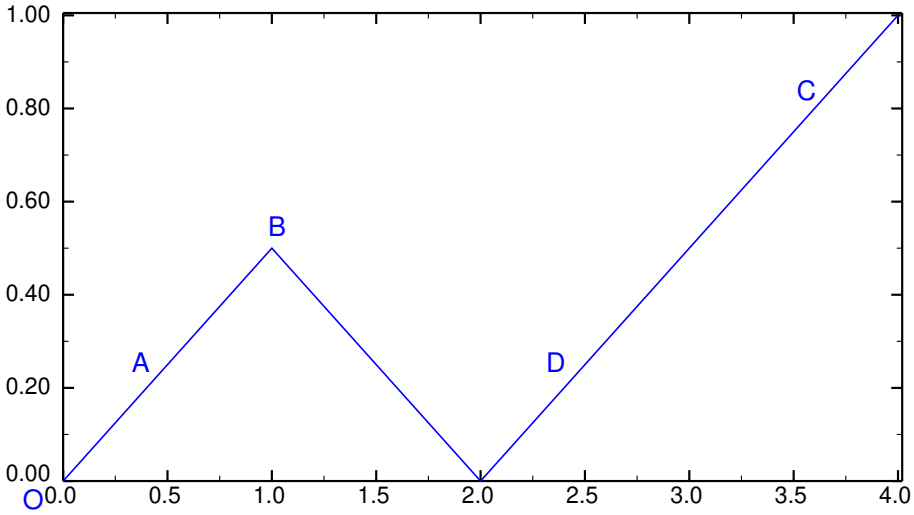
New tests verify the capability of prescribing initial stresses for materials with the Mullins effect model.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The first figure below compares stress-strain responses of two analyses in which specimens are subjected to nonmonotonic uniaxial loading.



The specimens are modeled using a hyperelastic material with the Ogden-Roxburgh Mullins effect model. In the first analysis, the material is subject to a nonmonotonic loading path as shown in the second figure.



In the second analysis, initial values of stress and maximum energy density are specified using results at point D from the first analysis as shown in the figures, and the load amplitude is the same as in the first analysis. The results of both analyses compare well.

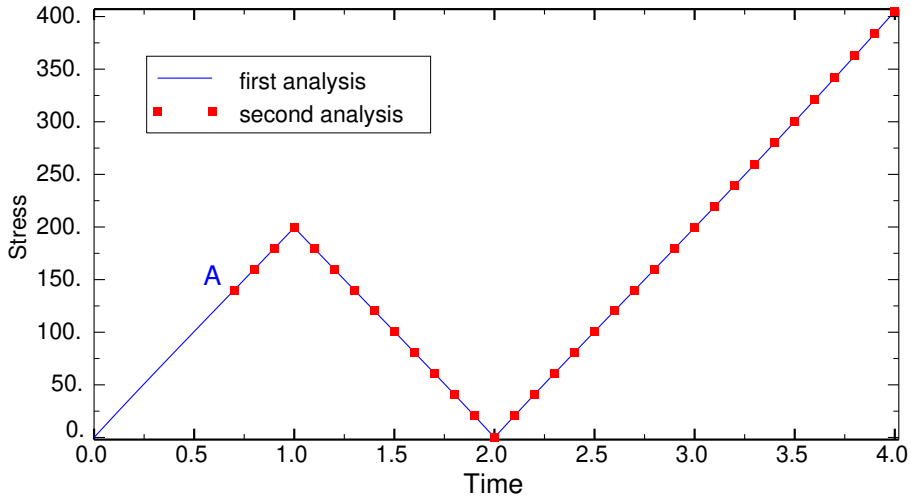
For more information, see *Defining Initial Stresses*.

Long-Term Initial Stresses

New tests verify the capability of prescribing long-term initial stresses.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The figure below compares stress-time histories of two analyses performed using viscoelastic materials.



In both cases small relaxation times are specified for each term in the Prony series that causes the stress to relax immediately. After the first analysis is completed, the second analysis uses the long-term initial stress capability to prescribe initial stresses using results from point A of the first analysis, and then the analysis continues with the same load applied. Both analyses produce identical results, which is expected in this case.

For more information, see *Defining Initial Stresses*.

C/C++ Interfaces to User Utility Routines

A new test with a user subroutine `UVAREL` written in C++ verifies C/C++ interfaces to utility subroutines `getElMPtVarC`, `getSecVarC`, and `getVrmC`.

Benefits: The tests verify the capability in Abaqus/Standard.

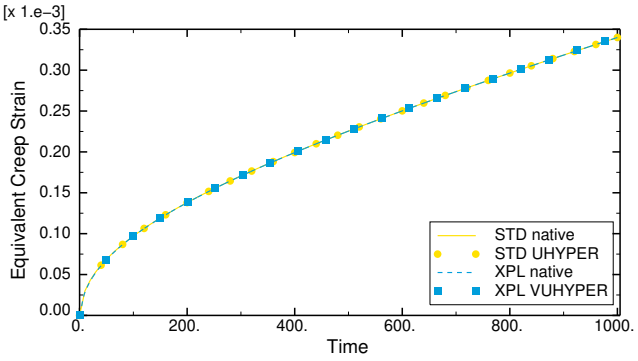
For more information, see *UVAREL*.

Viscous Softening

New tests verify the viscous softening capability in the parallel rheological framework.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The figure below shows the stress-strain response of a nonlinear viscoelastic material subjected to uniaxial loading and unloading and undergoing viscous softening.



Results given by Abaqus/Standard and Abaqus/Explicit are in good agreement with each other and with an equivalent implementation using the `UHYPER` and `VUHYPER` user subroutines.

For more information, see *Nonlinear large-strain viscoelasticity with hyperelasticity*.

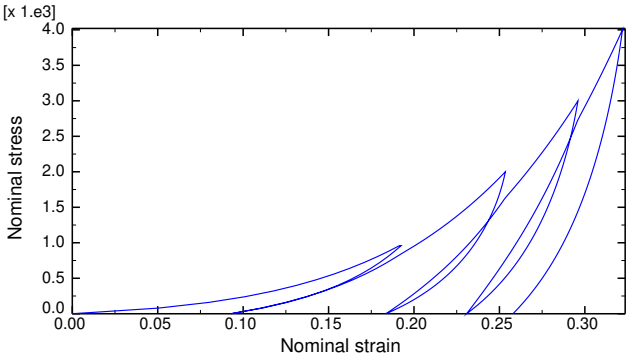
2026 GA

Mullins Effect with Hyperfoam Material and Large-Strain Viscoelasticity

New tests verify the Mullins effect modeling capabilities with large-strain time-domain viscoelasticity and a hyperfoam material.

Benefits: The tests verify the capability in Abaqus/Standard.

The figure below shows the stress-strain response of a specimen subjected to cycling uniaxial loading. The specimen is modeled using the hyperfoam material with large-strain viscoelasticity and the Ogden-Roxburgh Mullins effect model. The response shows that the material can capture the hysteresis as well as the stress-softening on unloading, which are exhibited by many polymers subjected to cyclic loading.



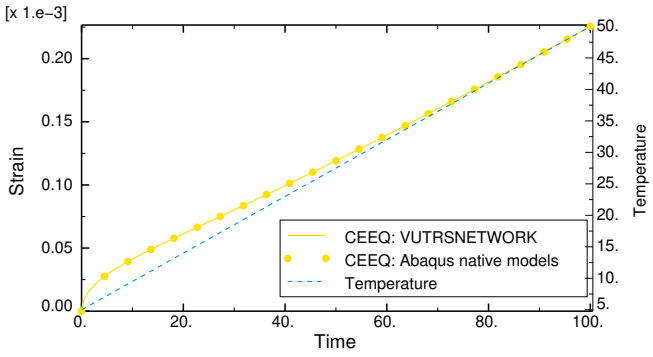
For more information, see *Stress Softening in Rubberlike Materials*.

`VUTRSNETWORK`

New tests verify the variables that are passed into user subroutine VUTRSNETWORK and compare results with those of equivalent temperature-time shifts available in Abaqus/Explicit.

Benefits: The tests verify the capability in Abaqus/Explicit.

The figure below shows the time history of creep strain and prescribed temperature from the new verification test. The results obtained using the VUTRSNETWORK subroutine agree with those obtained using native Abaqus models.



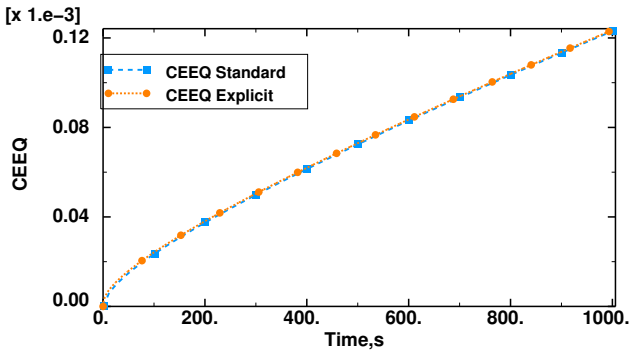
For more information, see *VUTRSNETWORK*.

Anand Model in a Parallel Rheological Framework

New tests verify the Anand creep model used in a parallel rheological framework.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

The figure below shows the history output of creep strain from one of the new uniaxial creep tests. The Anand law is used to model the creep behavior of one of the four viscoelastic networks. Results given by Abaqus/Standard and Abaqus/Explicit are in good agreement.



For more information, see *Parallel Rheological Framework* and *Nonlinear large-strain viscoelasticity with hyperelasticity*.

2025 FD03

Contact Wear Modeling

New tests verify the generation of a 3D model using symmetric model generation.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Contact wear modeling*.

Kamal Cure Kinetics Model

New tests verify the cure modeling capabilities using the Kamal model with field-variable dependent parameters.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Curing Processes in Polymers* and *Cure modeling in Abaqus/Standard*.

Ductile Damage Initiation Criteria

New tests verify the evaluation of ductile damage initiation criteria for hyperelastic materials with isotropic plasticity.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Progressive Damage and Failure* and *Progressive damage and failure of ductile metals*.

Transferring Results with Nodal Temperature and Field Variables

New tests verify the transfer of results with nodal temperature and field variables from an Abaqus/Standard analysis to an Abaqus/Explicit analysis, from an Abaqus/Explicit analysis to an Abaqus/Standard analysis, and from an Abaqus/Explicit analysis to an Abaqus/Explicit analysis.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Transferring results with nodal temperature and field variables*.

Transferring Results of an Analysis Model Multiple Times

New tests verify the transfer of results of solid element sets multiple times from an Abaqus/Standard analysis to an Abaqus/Explicit analysis and from an Abaqus/Explicit analysis to an Abaqus/Explicit analysis.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Transferring results of an analysis model multiple times*.

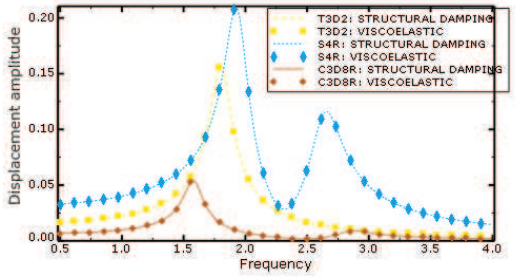
2025 FD02

Structural Damping

New tests verify constant and frequency-dependent structural damping.

Benefits: The tests verify the capability in Abaqus/Standard.

A steady-state dynamic analysis is conducted to determine the steady-state response of 1-element structures subjected to uniaxial harmonic excitation. The following figure shows results for a linear elastic material with structural damping and for a material with structurally equivalent viscoelasticity without structural damping. The results obtained using the two different approaches agree very well with each other.



For more information, see *Structural damping* and *Structural Damping*.

Adjoint Design Sensitivity Analysis for Combined Abaqus and fe-safe–Based Fatigue Procedures

A new test verifies the adjoint fatigue design responses and corresponding sensitivity analysis techniques for shell and solid continuum elements for a combined Abaqus and fe-safe–based fatigue procedure.

Benefits: The test verifies the capability in Abaqus/Standard.

For more information, see *Adjoint design sensitivity analysis for combined Abaqus and fe-safe–based fatigue procedures*.

Mullins Effect

New tests verify the Mullins effect modeling capabilities with the exponential and slow decay forms used to define the evolution of the damage variable.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Mullins effect and permanent set*.

2025 FD01

Paperboard Material Model

New tests verify the modeling of paperboard material.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Paperboard material model*.

UMAT and UHYPER

New tests verify that the correct value of the network identifier is passed into user subroutine UHYPER when the subroutine is used with the parallel rheological framework.

Benefits: The tests verify the capability in Abaqus/Standard

For more information, see *UMAT and UHYPER*.

UHYPER_STRETCH

New tests verify that the correct value of the network identifier is passed into user subroutine UHYPER_STRETCH when the subroutine is used with the parallel rheological framework.

Benefits: The tests verify the capability in Abaqus/Standard

For more information, see *UHYPER_STRETCH*.

VUHYPER

New tests verify the variables that are passed into user subroutine VUHYPER and compare results with those of equivalent hyperelastic material models available in Abaqus/Explicit.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *VUHYPER*.

Normal Definitions of Beams and Shells

New tests verify the use of B21, B31, B32OSH, and B33H elements with user-specified nodal normals.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Normal definitions of beams and shells*.

Elastic Materials

New tests verify the hyperelasticity with Hencky strain energy function in Abaqus/Explicit.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Elastic materials*.

Miscellaneous Results Transfer Tests

New tests verify the results of transfer tests.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.
Tests verify the transfer of the state at the fluid cavity reference node when the material state is imported and the transfer of the state of contact pairs for models that uses part instance.

For more information, see *Miscellaneous results transfer tests*.

Coupling Constraints

New tests verify that appropriate constraints are formed for distributing couplings with collinear cloud nodes with rotational degrees of freedom that are allowed to participate in the coupling.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Coupling constraints*.

Contact Wear Modeling

New tests verify the contact wear capability for general contact.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Contact wear modeling*.

PDF Guides Available

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

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

For more information, see *Abaqus Verification Guide*.

2025 GA

Contact Wear Modeling

New tests verify step cycling controls for general contact.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Contact wear modeling*.

2024 FD04

Contact Wear Modeling

New tests verify the contact wear capability for general contact.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Contact wear modeling*.

2024 FD03

UVAREL

New tests verify variables that are passed into user subroutine `UVAREL` and integration point quantities that are read by utility subroutine `getElMPtVar`.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *UVAREL*.

Piezoresistive Effect

New tests model one-dimensional electrical current flow using axisymmetric and three-dimensional elements.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Piezoresistivity properties: Piezoresistive effect*.

Bending Stabilization

New tests verify the bending stabilization capability.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Bending stabilization for full-integration linear elements* and *Using Section Controls for Bending Stabilization*.

User-Defined Element Output Variables

New tests verify the definition and output of user-defined element variables.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *UVAREL* and **ELEMENT USER OUTPUT VARIABLES*.

Thermal Expansion

New tests verify the tangent and secant thermal expansion capability.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Thermal Expansion*.

Stretching of Solid Body Using SPH Element Conversion

New tests verify the accuracy of Lagrangian element (C3D8R) to SPH particle (PC3D) conversion for a solid body subjected to uniaxial stretching.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Smoothed particle hydrodynamic analysis*.

Progressive Damage and Failure in Fiber-Reinforced Materials

A new Python test models progressive damage and failure in fiber-reinforced materials with multiscale modeling.

Benefits: The test verifies the capability in Abaqus/Standard.

For more information, see *Progressive damage and failure in fiber-reinforced materials*.

2024 FD02

Undrained Pore Fluid Flow and Stress Analysis

New tests verify undrained pore fluid flow and stress analysis.

Benefits: The tests verify the capability in Abaqus/Explicit.
A unit cube of geometry with distorted meshes is used to test three-dimensional solid elements, and a simple rectangular geometry is used to test plane strain elements as well as axisymmetric solid elements. The medium is subject to prescribed displacement boundary conditions, and the numerical results are compared with a reference solution.

For more information, see *Undrained pore fluid flow and stress analysis*.

2024 FD01

Progressive Damage and Failure in Fiber-Reinforced Materials

New tests contain basic cases to model progressive damage and failure in fiber-reinforced materials with multiscale modeling.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Progressive damage and failure in fiber-reinforced materials*.

Generalized Damage and Failure for Anisotropic Materials

New tests contain basic cases with various damage initiation criteria for anisotropic materials.

Benefits: The tests verify the capability in Abaqus/Standard and Abaqus/Explicit.

For more information, see *Generalized damage and failure for anisotropic materials*.

Field Expansion

New tests verify the field expansion capability for both solid and pore fluid phases in a porous medium.

Benefits: The tests verify the capability in Abaqus/Standard.
The solid grains and the pore fluid in a porous medium are subjected to thermal expansion and multiple field expansions. Both built-in capabilities and user subroutine *UEXPAN* are used.

For more information, see *Field expansion test*, *UEXPAN*, and *UEXPAN*.

Band-Limited Damping in Abaqus/Standard

New tests verify the use of band-limited damping with various element types and material models.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Material Damping*.

Updated SLIPRING Contact Angle in Abaqus/Standard

New tests verify the use of the updated contact angle (based on the nodal coordinates) for the SLIPRING connection type in Abaqus/Standard.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Frictional SLIPRING connectors*.

Contact Wear Modeling

New tests verify the contact wear capability for both general contact and contact pairs.

Benefits: The tests verify the capability in Abaqus/Standard.

For more information, see *Contact wear modeling*.

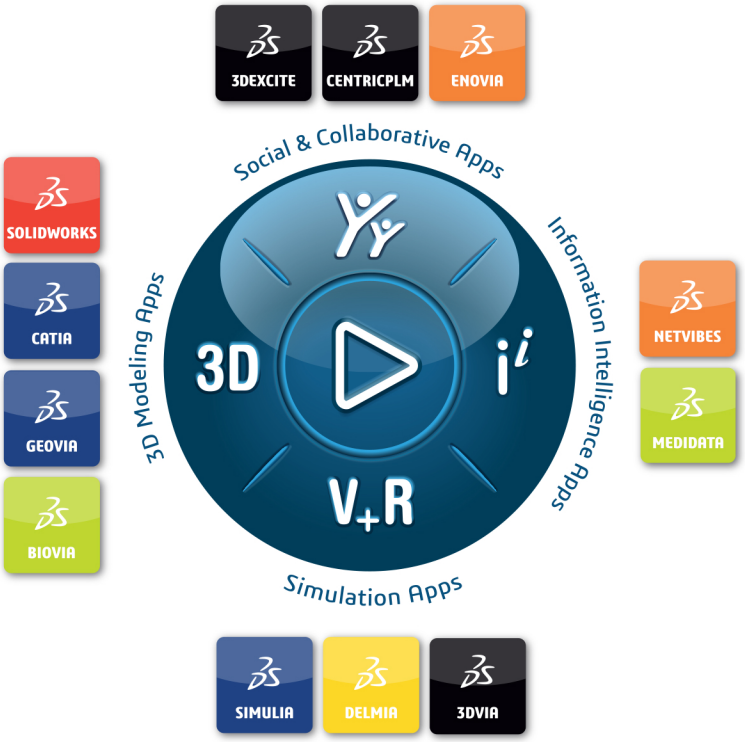
2024 GA

Tie Constraints

New tests verify that the configuration used for the formation of tie constraints in an Abaqus/Explicit import analysis depends on whether or not there are changes to the definitions of the surfaces to be tied.

Benefits: The tests verify the capability in Abaqus/Explicit.

For more information, see *Miscellaneous results transfer tests*.



Our **3D**EXPERIENCE® platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes is a catalyst for human progress. We provide business and people with collaborative virtual environments to imagine sustainable innovations. By creating virtual twin experiences of the real world with our **3D**EXPERIENCE platform and applications, our customers can redefine the creation, production and life-cycle-management processes of their offer and thus have a meaningful impact to make the world more sustainable. The beauty of the Experience Economy is that it is a human-centered economy for the benefit of all – consumers, patients and citizens.

Dassault Systèmes brings value to more than 300,000 customers of all sizes, in all industries, in more than 150 countries. For more information, visit **www.3ds.com**.

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes
17F, Foxconn Building,
No. 1366, Lujiazui Ring Road
Pilot Free Trade Zone, Shanghai 200120
China

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA

DASSAULT SYSTEMES

©2026 Dassault Systèmes. All rights reserved. **3DEXPERIENCE**, the 3DS logo, the Compass icon, IFWE, 3DEXCITE, 3DVIA, BIOVIA, CATIA, CENTRIC PLM, DELMIA, ENOVIA, GEOVIA, MEDIDATA, NETVIBES, OUTSCALE, SIMULIA and SOLIDWORKS are commercial trademarks or registered trademarks of Dassault Systèmes, a European company (Societas Europaea) incorporated under French law, and registered with the Versailles trade and companies registry under number 322 306 440, or its subsidiaries in the United States and/or other countries.