SIMULIA

3DEXPERIENCE

ABAQUS 2026 FD04
MATERIALS GUIDE

DASSAULT SYSTEMES

Contents

Trademarks and Legal Notices....9
Abaqus Materials Guide....10
What's New....11
Introduction....35
About the Material Library....36
Defining Materials....36
Available Material Behaviors....36
Using Material Behaviors with Various Element Types....37
Using Complete Material Definitions....37
Defining Spatially Varying Material Behavior for Homogenous Solid Continuum Elements Using Distributions in Abaqus/Standard....37
Material Data Definition....38
Material Definitions....38
Large-Strain Considerations....39
Specifying Material Data as Functions of Temperature and Independent Field Variables....39
Specifying Material Data as Functions of Solution-Dependent Variables....42
Material Failure and Element Deletion....44
Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit....45
Regularizing User-Defined Data in Abaqus/Explicit....45
Evaluation of Strain Rate–Dependent Data in Abaqus/Explicit....47
Combining Material Behaviors....49
Complete Material Definitions....49
Material Behavior Combination Tables....50
Density....56
Defining Density....56
Units....57
Elements....57
Elastic Mechanical Properties....58
Elastic Behavior....59
Thermal Strain....61
Elastic Strain Magnitude....61
Linear Elasticity....62
Linear Elastic Behavior....63
No Compression or No Tension....77
Plane Stress Orthotropic Failure Measures....81
Plasticity Corrections....86
Elastic Behavior of Porous Materials....93
Logarithmic Porous Elasticity Model....93
Power Law–Based Porous Elasticity Model....95
Material Options....95
Elements....96
Hypoelastic Behavior....97
Defining Hypoelastic Material Behavior....97

Determining the Hypoelastic Material Parameters....97
Plane or Uniaxial Stress....98
Large-Displacement Analysis....98
Material Options....98
Elements....98
Hyperelasticity....99
Hyperelastic Behavior of Rubberlike Materials....100
Hyperelastic Behavior in Elastomeric Foams....125
Anisotropic Hyperelastic Behavior....136
Stress Softening in Elastomers....146
Stress Softening in Rubberlike Materials....147
Stress Softening in Elastomeric Foams....161
Linear Viscoelasticity....167
Time Domain Viscoelasticity....168
Frequency Domain Viscoelasticity....186
Structural Relaxation in Glass....196
Nonlinear Viscoelasticity....200
Hysteresis in Elastomers....201
Parallel Rheological Framework....204
Low-Density Foams....219
Mechanical Response....219
Tension Cutoff and Failure in Abaqus/Explicit....222
Thermal Expansion....223
Material Stability....223
Elements....224
Procedures....224
Superelasticity....225
Material Behavior....225
Defining the Elastic Constants of Austenite....227
Defining the Properties of Martensite and the Transformation Plateau....227
Shape Setting....228
Elements....228
Output....229
References....229
Inelastic Mechanical Properties....230
Inelastic Behavior....231
Plasticity Theories....235
Elastic Response....235
Stress and Strain Measures....236
Metal Plasticity....239
Classical Metal Plasticity....240
Models for Metals Subjected to Cyclic Loading....248
Rate-Dependent Yield....265
Rate-Dependent Plasticity: Creep and Swelling....269
Annealing or Melting....284
Hill Anisotropic Yield/Creep....287
Nonquadratic Yield....294
Johnson-Cook Plasticity....301

Dynamic Failure Models....307
Porous Metal Plasticity....313
Cast Iron Plasticity....320
Two-Layer Viscoplasticity....324
ORNL – Oak Ridge National Laboratory Constitutive Model....329
Deformation Plasticity....332
Other Plasticity Models....335
Extended Drucker-Prager Models....336
Modified Drucker-Prager/Cap Model....362
Soft Rock Plasticity Model....376
Mohr-Coulomb Plasticity....383
Critical State (Clay) Plasticity Model....391
Crushable Foam Plasticity Models....400
Fabric Material Behavior....412
Fabric Material Behavior....413
Test Data–Based Fabric Materials....414
User-Defined Fabric Materials....424
Thermal Expansion....425
Fabric Thickness....425
Defining a Reference Mesh (Initial Metric)....425
Defining Yarn Directions in the Reference Configuration....426
Picture-Frame Shear Fabric Test....426
Material Options....427
Elements....427
Procedures....428
Output....428
Jointed Material Model....429
Joint Opening/Closing....429
Compressive Joint Sliding....430
Bulk Failure....431
Nonassociated Flow....432
Elements....433
Output....433
Concrete....434
Concrete Smeared Cracking....435
Cracking Model for Concrete....448
Concrete Damaged Plasticity....457
Permanent Set in Rubberlike Materials....476
Material Behavior....476
Specifying Permanent Set....477
Limitations....478
Elements....478
Procedures....478
Output....478
References....478
Plasticity Model for Superelastic Materials....479
Material Behavior....479
Elements....481

Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials....482
Shear Plasticity Behavior....482
Elements....483
Paperboard Material Model....484
Local Directions....484
In-plane Response....485
ZD Response....489
Out-of-Plane Shear Response....489
Damage and Failure....490
Elements....490
Output....490
References....491
Progressive Damage and Failure....492
About Progressive Damage and Failure....493
General Framework for Modeling Damage and Failure....494
Mesh Dependency....495
Damage and Failure for Ductile Metals....497
About Damage and Failure for Ductile Metals....498
Damage Initiation for Ductile Metals....501
Damage Evolution and Element Removal for Ductile Metals....518
Damage and Failure for Fiber-Reinforced Composites....529
About Damage and Failure for Fiber-Reinforced Composites....530
Damage Initiation for Fiber-Reinforced Composites....534
Damage Evolution and Element Removal for Fiber-Reinforced Composites....544
Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling....555
About Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling....556
Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling....559
Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling....564
Generalized Damage and Failure for Anisotropic Materials....571
About Generalized Stress- and Strain-Based Damage and Failure for Anisotropic Materials....572
Generalized Damage Initiation Criteria for Anisotropic Materials....573
Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis....577
About Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis....578
Damage Initiation for Ductile Materials in Low-Cycle Fatigue....580
Damage Evolution for Ductile Materials in Low-Cycle Fatigue....582
Hydrodynamic Properties....585
Hydrodynamic Behavior....586
Deviatoric Behavior....587
Thermal Strain....587
Equation of State....588
Energy Equation and Hugoniot Curve....589
Mie-Grüneisen Equations of State....589
Tabulated Equation of State....591
User-Defined Equation of State....592
P–alpha Equation of State....593
JWL High Explosive Equation of State....597
Ignition and Growth Equation of State....599
Ideal Gas Equation of State....601

Deviatoric Behavior....604
Use with the Mises or the Johnson-Cook Plasticity Models....605
Use with the Extended Drucker-Prager Plasticity Models....606
Use with the Tensile Failure Model....607
Adiabatic Assumption....607
Modeling Fluids....607
Elements....608
Output....608
References....608
Multiscale Properties....610
Multiscale Material Modeling....611
Mean-Field Homogenization....612
Specifying Constituent Properties for a Composite Material....612
Mean-Field Homogenization for Linear Elastic Composites....612
Mean-Field Homogenization for Thermal Properties....615
Incremental Mean-Field Homogenization for Nonlinear Composites....616
Multistep Homogenization....618
Specifying the Microstructure of the Composite....619
Composites with Thermal Expansion....624
Composites with Viscoelasticity....625
Composites Curing....626
Composites with Damage....626
Example....627
Elements....628
Output....628
References....629
Other Material Properties....630
Mechanical Properties....631
Fatigue Crack Growth Laws for Linear Elastic Materials....632
Material Damping....639
Thermal Expansion....648
Field Expansion....657
Pore Fluid Pressure....663
Viscosity....665
CZone Crush Stress....670
Heat Transfer Properties....673
About Thermal Properties....674
Conductivity....675
Specific Heat....677
Latent Heat....679
Curing Processes in Polymers....681
Modeling Curing Processes....681
Cure Shrinkage Strain....685
Example: Defining the Cure Properties of a Material....686
Initial Conditions....687
Material Options....687
Elements....688
Procedures....688

Output....688
References....689
Acoustic Medium....690
Defining an Acoustic Medium....690
Volumetric Drag....691
Porous Acoustic Material Models....691
Fluid Cavitation....695
Defining a Steady Flow Field....696
Elements....696
Output....696
References....696
Poroelastic Acoustic Medium....698
Frequency Response of a Poroelastic Acoustic Medium....698
Poroelastic Acoustic Material Behavior....699
Elements....700
Output....700
Mass Diffusion Properties....701
Diffusivity....702
Solubility....707
Electromagnetic Properties....708
Electrical Conductivity....709
Electrical Resistivity....711
Piezoelectric Behavior....713
Piezoresistive Effect....718
Magnetic Permeability....721
Pore Fluid Flow Properties....726
About Pore Fluid Flow Properties....727
Permeability....729
Porous Bulk Moduli....735
Sorption....736
Swelling Gel....740
Moisture Swelling....742
User Materials....745
User-Defined Mechanical Material Behavior....746
User-Defined Thermal Material Behavior....753
User-Defined Material Properties....756
Special-Purpose Material Modeling Techniques....757
About Special-Purpose Material Modeling Techniques....758
Material User Type Definition....758
Reserved Material Name Definition....759
Modeling the Cure Process in Thermosetting Polymers....760
Reaction Kinetics....761
Cure Heat Generation....762
Cure Shrinkage Strain....762
Mechanical Response of the Material....764
Solution-Dependent State Variables....764
Example: Defining the Cure Modeling Capabilities....765
Elements....766

Procedures....766
Output....766
References....766
Modeling Metallurgical Phase Transformations....767
Material Physical Raw-Liquid-Solid (RLS) State....768
Solid-Phase Transformation....768
Grain Morphology....772
Grain Size....773
Specifying Material Properties as Functions of Metallurgical State....774
Solution-Dependent State Variables....775
References....776
Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)....777
CANNs Model Architecture....778
CANNs Model Definition....778
Examples....782
Elements....784
References....784
Modeling Multiscale Materials with Deep Material Networks (DMNs)....785
DMN Model Architecture....786
Microstructure Characteristics....788
Constituent Properties....789
Example: Defining a Short Fiber-Reinforced Composite with DMN....792
Elements....792
Procedures....792
Output....792
References....793
Modeling The Electrical and Mechanical Response Of Cardiac Tissue....794
Modeling the Electrical Response of Cardiac Tissue....795
Modeling the Mechanical Response of Cardiac Tissue....801
Parameter Table Type Reference....808
ABQ_CANN_Coeff....809
ABQ_CANN_DerivedInvariant....810
ABQ_CARDIAC_ELEC_ACTIVATION....812
ABQ_CARDIAC_ELEC_PARAMETERS_AP....813
ABQ_Cure_HeatGeneration....814
ABQ_Cure_ReactionKinetics_Kamal....815
ABQ_DMN_Controls....816
ABQ_DMN_Level....817
ABQ_MULTSCALE_Prony....818
ABQ_MULTSCALE_Shape....819
ABQ_PHASE_TRANS_Diffusional_TTT_Constants....820
ABQ_PHASE_TRANS_FieldVariables....821
ABQ_PHASE_TRANS_GrainSize_Definitions....822
ABQ_PHASE_TRANS_Martensitic_TTT_Constants....823
ABQ_PHASE_TRANS_MeltingTemperature....824
ABQ_PHASE_TRANS_SolidPhases....825
ABQ_PHASE_TRANS_SolidificationMap_Constants....826
ABQ_PHASE_TRANS_Transformations....827

Property Table Type Reference....830
ABQ_Cure_ReactionKinetics_MaxConversion....831
ABQ_Cure_ReactionKinetics_Tabular....832
ABQ_Cure_ShrinkageCoeff_Aniso....833
ABQ_Cure_ShrinkageCoeff_Iso....834
ABQ_Cure_ShrinkageCoeff_Ortho....835
ABQ_Cure_ShrinkageCoeff_Vol....836
ABQ_MULTSCALE_AspectRatio....837
ABQ_MULTSCALE_Chaboche....838
ABQ_MULTSCALE_ExpLinearHard....839
ABQ_MULTSCALE_IsoCTE....840
ABQ_MULTSCALE_IsoElas....841
ABQ_MULTSCALE_IsoHard....842
ABQ_MULTSCALE_Oritens....843
ABQ_MULTSCALE_PowerHard....844
ABQ_MULTSCALE_PowerLaw....845
ABQ_MULTSCALE_Prandtl....846
ABQ_MULTSCALE_TransIsoCTE....847
ABQ_MULTSCALE_TransIsoElas....848
ABQ_MULTSCALE_Volfrac....849
ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients....850
ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac....851
ABQ_PHASE_TRANS_Diffusional_TTT_Diagram....852
ABQ_PHASE_TRANS_GrainSize_Coefficients....853
ABQ_PHASE_TRANS_Martensitic_KM_Coefficients....854
ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac....855
ABQ_PHASE_TRANS_Martensitic_TTT_Diagram....856
ABQ_PHASE_TRANS_SolidificationMap_Diagram....857

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 Materials Guide

The Abaqus Materials Guide describes the Abaqus material modeling options and how to calibrate some of the more advanced material models.

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.

Materials are defined by selecting material behaviors and defining them and combining complementary material behaviors such as elasticity and plasticity. A local coordinate system can be used for material calculations. Any anisotropic properties must be given in this local system.

The material library in Abaqus is intended to provide comprehensive coverage of both linear and nonlinear, isotropic and anisotropic material behaviors. The use of numerical integration in the elements, including numerical integration across the cross-sections of shells and beams, provides the flexibility to analyze the most complex composite structures.

Material behaviors fall into the following general categories:

- general properties (material damping, density, thermal expansion);
- elastic mechanical properties;
- inelastic mechanical properties;
- multiscale properties;
- thermal properties;
- acoustic properties;
- hydrostatic fluid properties;
- equations of state;
- mass diffusion properties;
- electrical properties; and
- pore fluid flow properties.

Some of the mechanical behaviors offered are mutually exclusive: such behaviors cannot appear together in a single material definition. Some behaviors require the presence of other behaviors; for example, plasticity requires linear elasticity. Such requirements are discussed at the end of each material behavior description, as well as in *Combining Material Behaviors*.

What's New

This page describes recent changes in Abaqus Materials.

2026 FD04

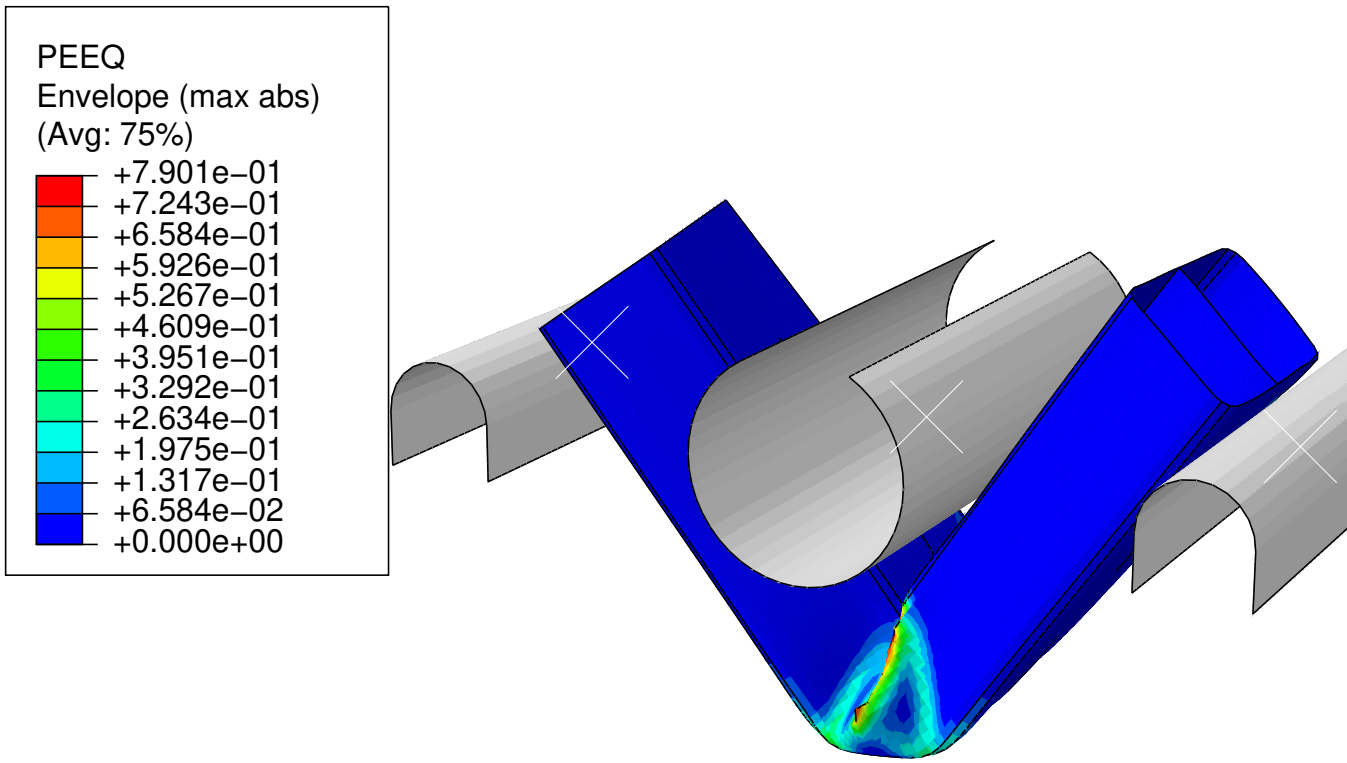
Modeling Nonquadratic Yield Using Plane Stress, Shell, and Membrane Elements

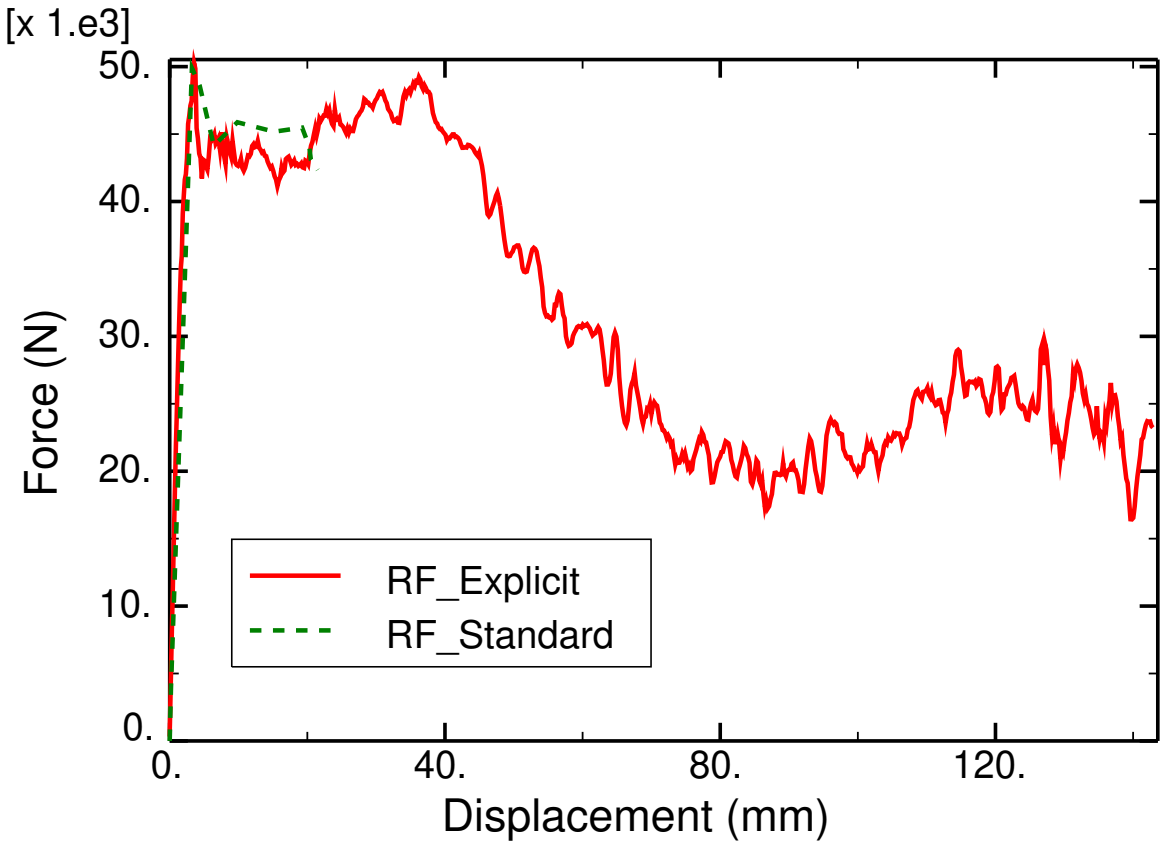
You can now model nonquadratic yield using plane stress, shell, and membrane elements in Abaqus/Standard and Abaqus/Explicit. You can also use a new yield criterion (Barlat Yld2000-2d) to model sheet metal forming.

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



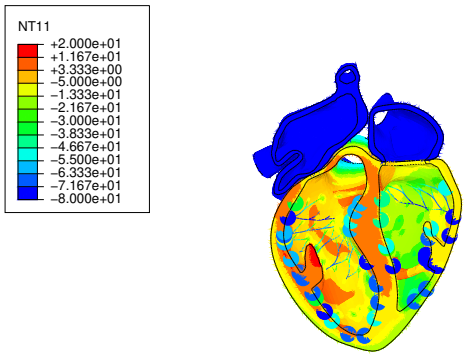
For more information, see *Nonquadratic Yield*.

Modeling the Electrical and Mechanical Behaviors of Cardiac Tissue

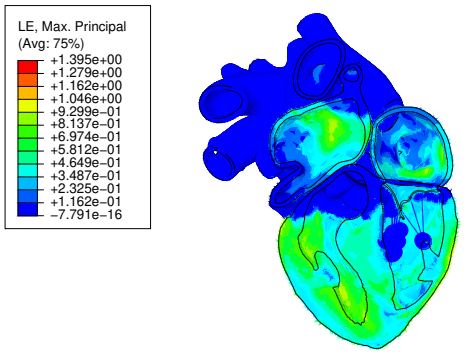
You can now model the electrical behavior of cardiac tissue in Abaqus/Standard and its mechanical behavior in Abaqus/Explicit.

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

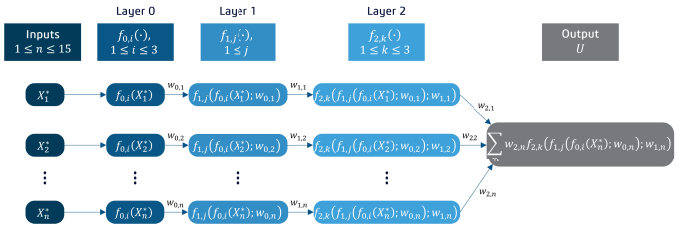
Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs) Models

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

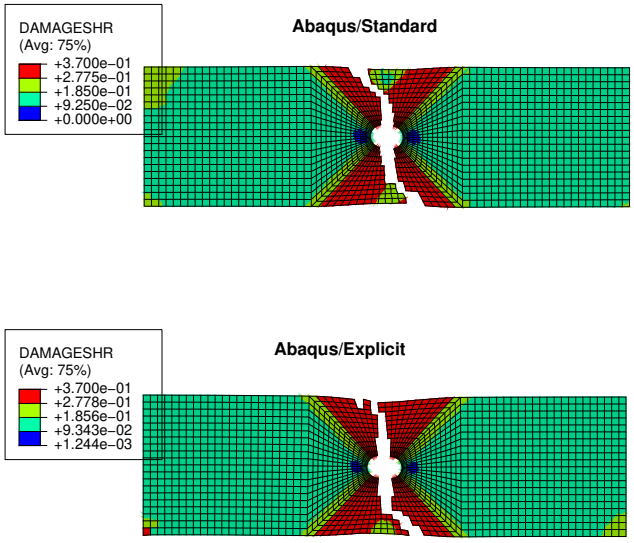
Ply Fabric Material Model in Abaqus/Standard

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*.

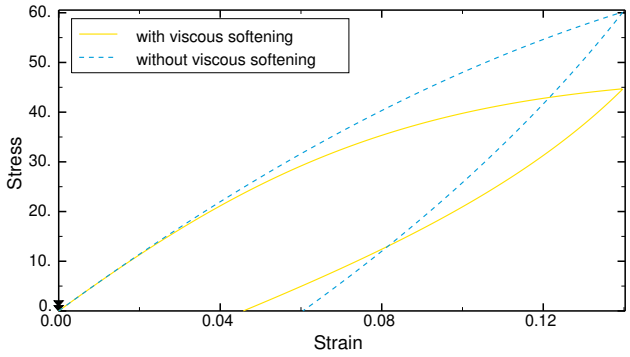
Viscous Softening

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.



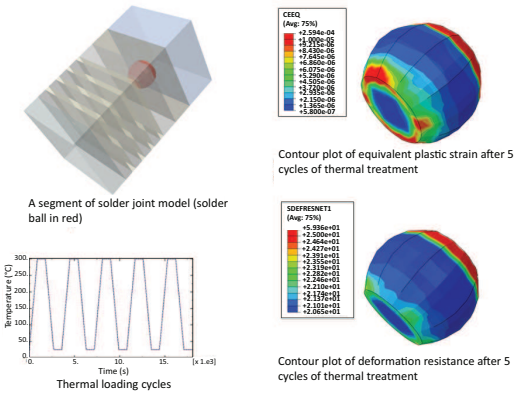
For more information, see *Parallel Rheological Framework*.

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*.

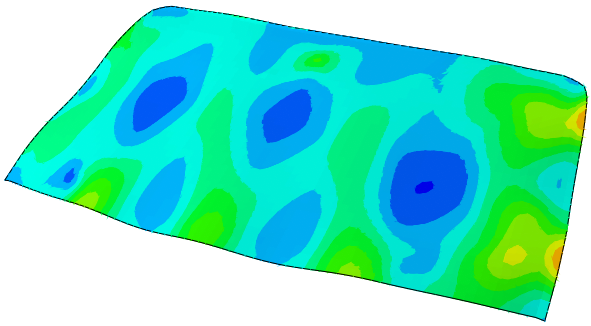
2025 FD02

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.



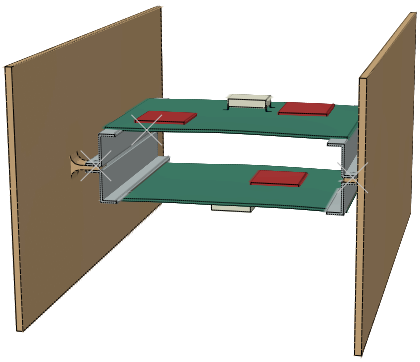
For more information, see *Defining Variable Frequency-Dependent Structural Damping in Abaqus/Standard* and *Structural damping*.

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.

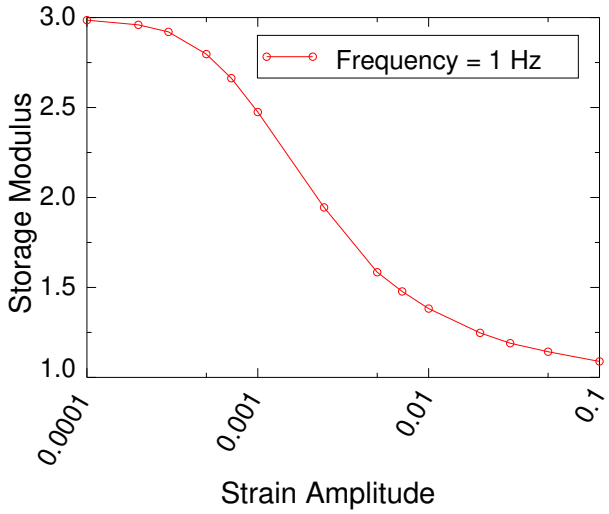
Benefits: You can now model amplitude-dependent material softening in rubber under dynamic loading accurately.

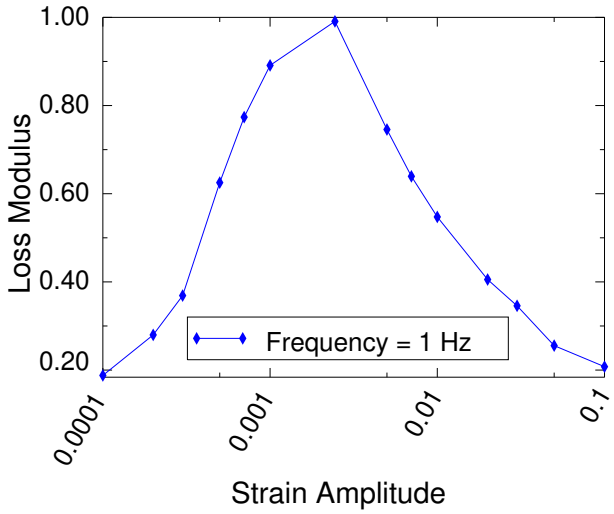
In earlier releases, Abaqus offered the Ogden-Roxburgh model to predict the softening in rubber materials subjected to quasi-static cyclic loading. This phenomenon is known as the Mullins effect and is characterized by irreversible softening of the stress-strain response that occurs as the material is 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

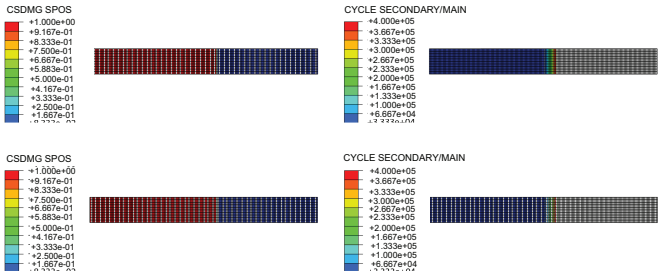
PRX-Fatigue Criterion

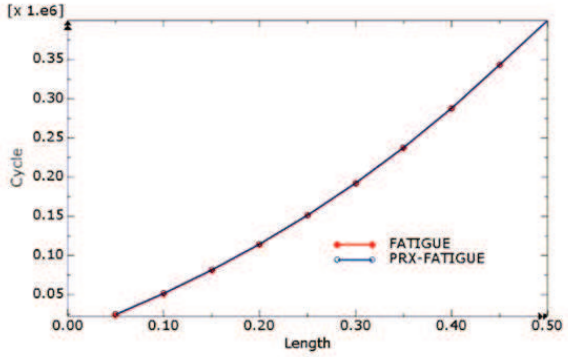
You can use the PRX-Fatigue criterion to model delamination propagation under cyclic loading in Abaqus/Standard.

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.

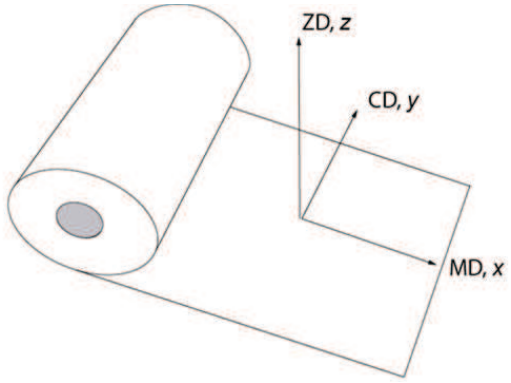
For more information, see *Hyperelastic Behavior of Rubberlike Materials*.

Paperboard Material Model

You can now model paperboard material in Abaqus/Explicit and Abaqus/Standard.

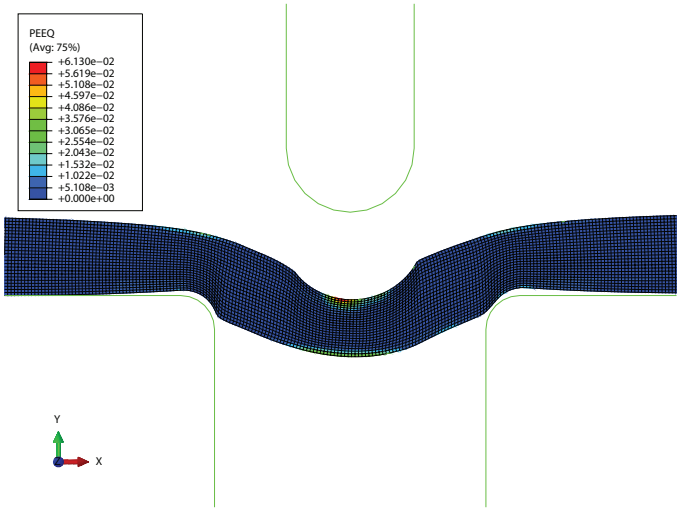
Benefits: You can now simulate workflows in industries that use paperboard material widely, such as in the packaging industry.

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.

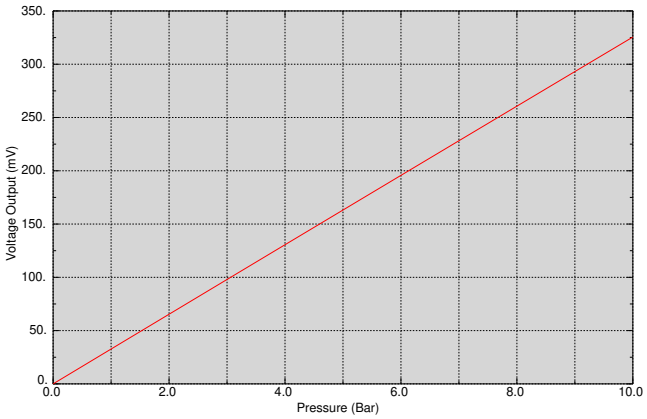
For more information, see *Abaqus Materials Guide*.

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

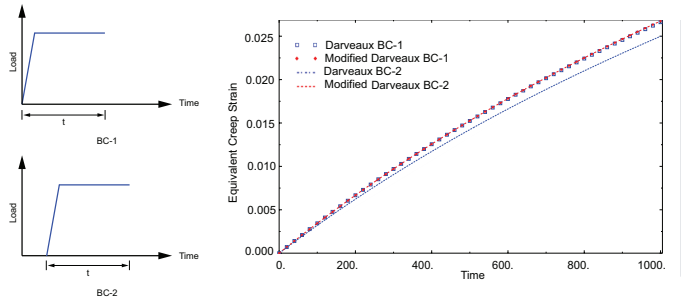
You can now use the modified Darveaux model to define creep response in the parallel rheological framework (PRF).

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

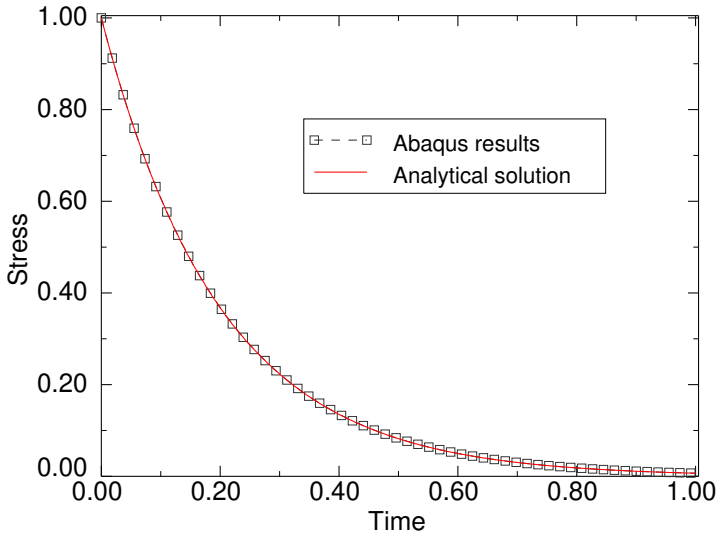
You can now combine nonlinear viscoelastic shear behavior with equation of state (EOS) materials in Abaqus/Explicit to model the response of viscoelastic fluids.

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*.

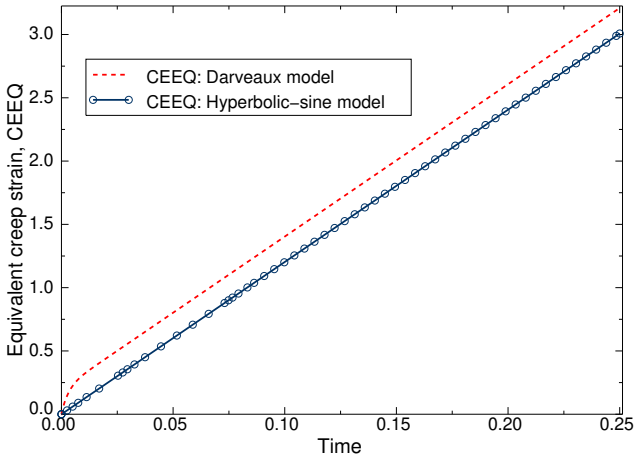
Darveaux Viscous Behavior for Parallel Rheological Framework

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

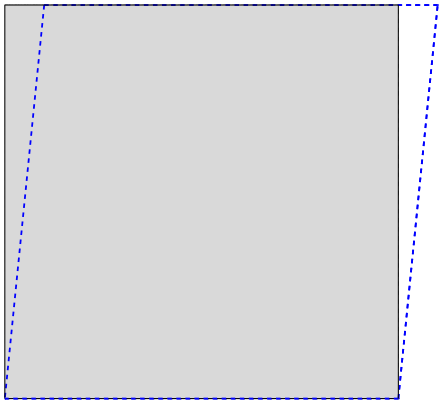
You can now use the Hencky model to define hyperelastic material response accurately.

Benefits: You can now accurately predict elastic material response for moderately large deformations.

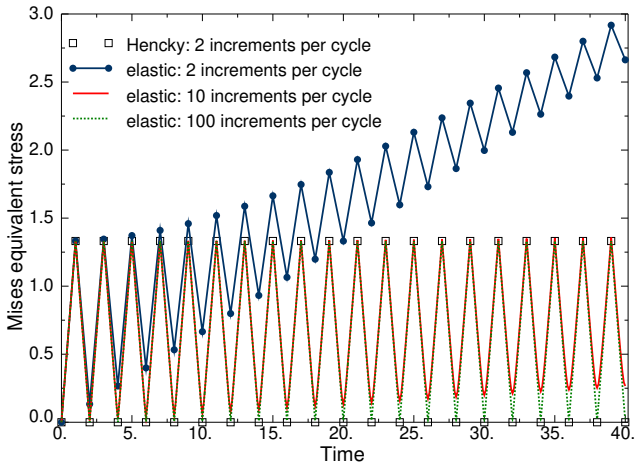
The Hencky model is an isotropic hyperelastic model that assumes a linear relation between the Kirchhoff stress tensor and the Hencky or logarithmic strain tensor. It is a generalization of the classical Hooke's law for isotropic infinitesimal elasticity. You define the model by providing two material 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

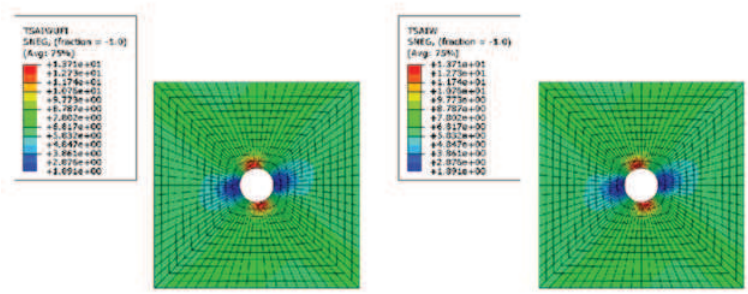
Generalized Damage and Failure for Anisotropic Materials

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.

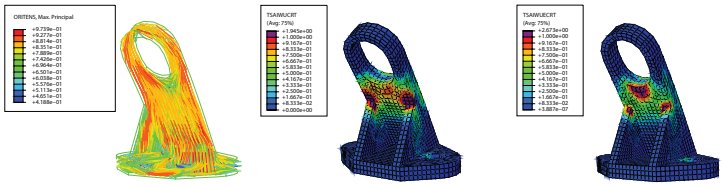
Benefits: The new stress/strain-based failure criteria are available with any material type and all stress/displacement elements, which gives you more options to evaluate the strength of materials.

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*.

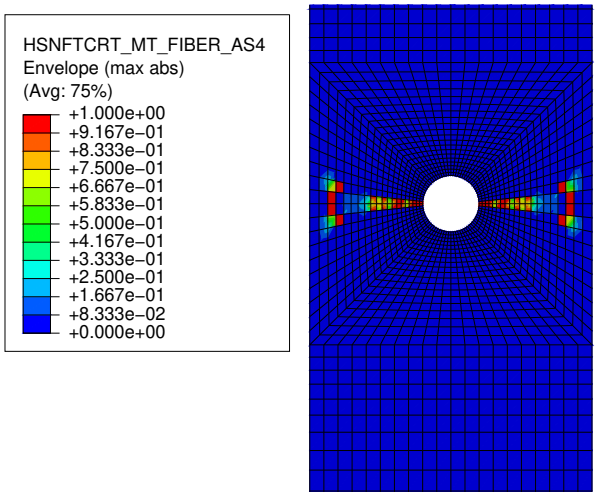
Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling

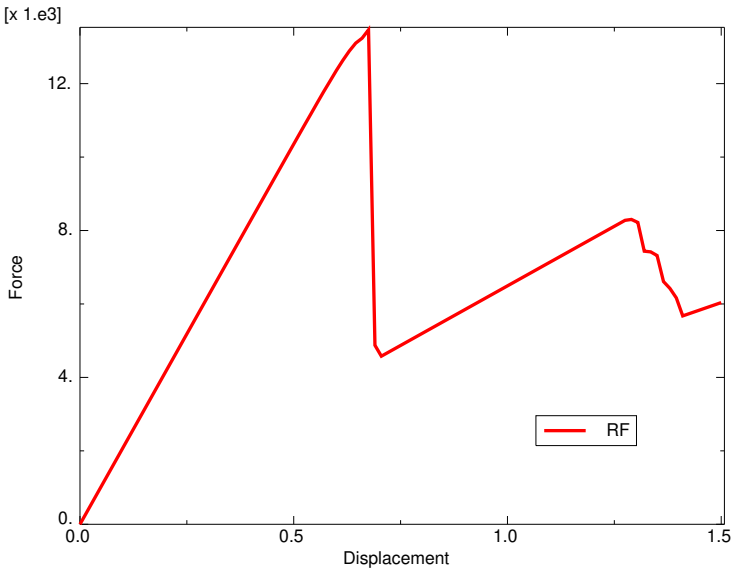
You can now specify new failure criteria for fiber-reinforced composite materials to evaluate the strength of the material based on stress invariants at the micro-level. You can also specify damage evolution laws for these damage criteria.

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.

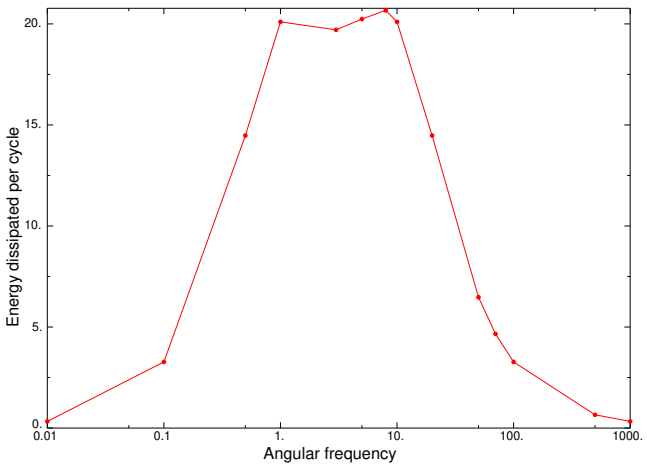
Benefits: The band-limited damping allows you to specify a damping ratio that remains approximately constant within a frequency range specified.

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

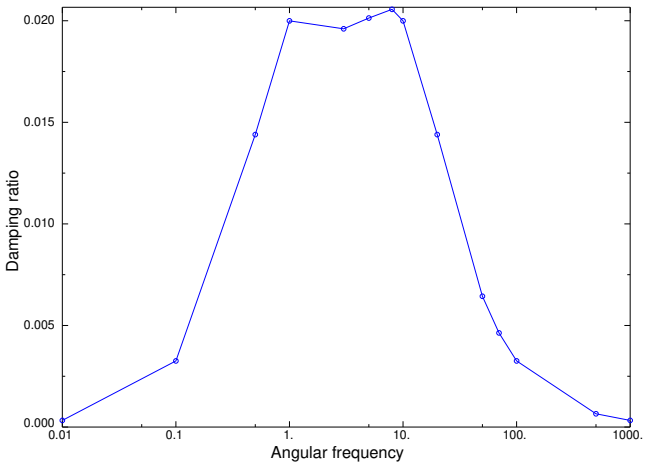
value of the damping ratio (and energy dissipated per cycle) within this range. In Abaqus/Standard, the effect of band-limited damping is considered only with the implicit dynamic procedure.

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*.

Introduction

The material library in Abaqus is intended to provide comprehensive coverage of both linear and nonlinear, isotropic, and anisotropic material behaviors. Materials are defined by selecting a combination of complementary material behaviors such as elasticity and plasticity.

In this section:

- *About the Material Library*
- *Material Data Definition*
- *Combining Material Behaviors*

About the Material Library

Overview

The material library in Abaqus is intended to provide comprehensive coverage of both linear and nonlinear, isotropic and anisotropic material behaviors.
The use of numerical integration in the elements, including numerical integration across the cross-sections of shells and beams, provides the flexibility to analyze the most complex composite structures.

This section describes how to define materials in Abaqus and contains brief descriptions of each of the material behaviors provided. Further details of the more advanced behaviors are provided in the *Abaqus Theory Guide*.

References:

Defining Materials

Materials are defined by:

- selecting material behaviors and defining them (*Material Data Definition*); and
- combining complementary material behaviors such as elasticity and plasticity (*Combining Material Behaviors*).

A local coordinate system can be used for material calculations (*Orientations*). Any anisotropic properties must be given in this local system.

Available Material Behaviors

Material behaviors fall into the following general categories:

- general properties (material damping, density, thermal expansion);
- elastic mechanical properties;
- inelastic mechanical properties;
- thermal properties;
- acoustic properties;
- hydrostatic fluid properties;
- equations of state;
- mass diffusion properties;
- electrical properties; and
- pore fluid flow properties.

Some of the mechanical behaviors offered are mutually exclusive: such behaviors cannot appear together in a single material definition. Some behaviors require the presence of other behaviors; for example, plasticity requires linear elasticity. Such requirements are discussed at the end of each material behavior description, as well as in *Combining Material Behaviors*.

Using Material Behaviors with Various Element Types

There are no general restrictions on the use of particular material behaviors with solid, shell, beam, and pipe elements. Any combination that makes sense is acceptable. The few restrictions that do exist are mentioned when that particular behavior is described in the pages that follow. A section on the elements available for use with a material behavior appears at the end of each material behavior description.

Using Complete Material Definitions

A material definition can include behaviors that are not meaningful for the elements or analysis in which the material is being used. Such behaviors will be ignored. For example, a material definition can include heat transfer properties (conductivity, specific heat) as well as stress-strain properties (elastic moduli, yield stress, etc). When this material definition is used with uncoupled stress/displacement elements, the heat transfer properties are ignored by Abaqus; when it is used with heat transfer elements, the mechanical strength properties are ignored. This capability allows you to develop complete material definitions and use them in any analysis.

Defining Spatially Varying Material Behavior for Homogenous Solid Continuum Elements Using Distributions in Abaqus/Standard

In Abaqus/Standard spatially varying mass density (*Density*), linear elastic behavior (*Linear Elastic Behavior*), and thermal expansion (*Thermal Expansion*) can be defined for homogeneous solid continuum elements using distributions (*Distribution Definition*). Using distributions in a model with significant variation in material behavior can greatly simplify pre- and postprocessing and improve performance during the analysis by allowing a single material definition to define the spatially varying material behavior. Without distributions such a model may require many material definitions and associated section assignments.

Material Data Definition

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Combining Material Behaviors*
- **MATERIAL*
- *Creating materials*

Overview

Each material definition can contain multiple material behaviors, as required, to specify the complete material behavior.

A material definition in Abaqus:

- specifies the behavior of a material and supplies all the relevant property data;
- is assigned a name, which is used to refer to those parts of the model that are made of that material;
- can have temperature and field variable dependence;
- can have solution variable dependence; and
- can be specified in a local coordinate system (*Orientations*), which is required if the material is not isotropic.

Material Definitions

Any number of materials can be defined in an analysis. Each material definition can contain any number of material behaviors, as required, to specify the complete material behavior. For example, in a linear static stress analysis only elastic material behavior might be needed, while in a more complicated analysis several material behaviors might be required.

A name must be assigned to each material definition. This name allows the material to be referenced from the section definitions used to assign this material to regions in the model.

Input File Usage: **MATERIAL*, NAME=*name*

Each material definition is specified in a data block, which is initiated by a **MATERIAL* option. The material definition continues until an option that does not define a material behavior (such as another **MATERIAL* option) is introduced, at which point the material definition is assumed to be complete. The order of the material behavior options is not important. All material behavior options within the data block are assumed to define the same material.

Abaqus/CAE Usage: Property module: material editor: **Name**
Use the menu bar under the **Material Options** list to add behaviors to a material.

Large-Strain Considerations

When giving material properties for finite-strain calculations, “stress” means “true” (Cauchy) stress (force per current area) and “strain” means logarithmic strain. For example, unless otherwise indicated, for uniaxial behavior

$$\varepsilon = \int \frac{dl}{l} = \ln\left(\frac{l}{l_0}\right).$$

Specifying Material Data as Functions of Temperature and Independent Field Variables

Material data are often specified as functions of independent variables such as temperature. Material properties are made temperature dependent by specifying them at several different temperatures.

In some cases a material property can be defined as a function of variables calculated by Abaqus; for example, to define a work-hardening curve, stress must be given as a function of equivalent plastic strain.

Material properties can also depend on “field variables” (user-defined variables that can represent any independent quantity and are defined at the nodes, as functions of time). For example, material moduli can be functions of weave density in a composite or of phase fraction in an alloy. See *Specifying Field Variable Dependence* for details. The initial values of field variables are given as initial conditions (see *Initial Conditions*) and can be modified as functions of time during an analysis (see *Predefined Fields*). This capability is useful if, for example, material properties change with time because of irradiation or some other precalculated environmental effect.

Any material behaviors defined using a distribution in Abaqus/Standard (mass density, linear elastic behavior, and thermal expansion) cannot be defined with temperature and/or field dependence. However, material behaviors defined with distributions can be included in a material definition with other material behaviors that have temperature and/or field dependence. See *Density*, *Linear Elastic Behavior*, and *Thermal Expansion*.

Interpolation of Material Data

In the simplest case of a constant property, only the constant value is entered. When the material data are functions of only one variable, the data must be given in order of increasing values of the independent variable. Abaqus then interpolates linearly for values between those given. Thus, you can give as many or as few input values as are necessary for the material model. If the material data depend on the independent variable in a strongly nonlinear manner, you must specify enough data points so that a linear interpolation captures the nonlinear behavior accurately.

When material properties depend on several variables, the variation of the properties with respect to the first variable must be given at fixed values of the other variables, in ascending values of the second variable, then of the third variable, and so on. The data must always be ordered so that the independent variables are given increasing values. This process ensures that the value of the material property is completely and uniquely defined at any values of the independent variables on which the property depends. See *Input Syntax Rules* for further explanation and an example.

Example: Temperature-Dependent Linear Isotropic Elasticity

Figure 1 shows a simple, isotropic, linear elastic material, giving the Young's modulus and the Poisson's ratio as functions of temperature.

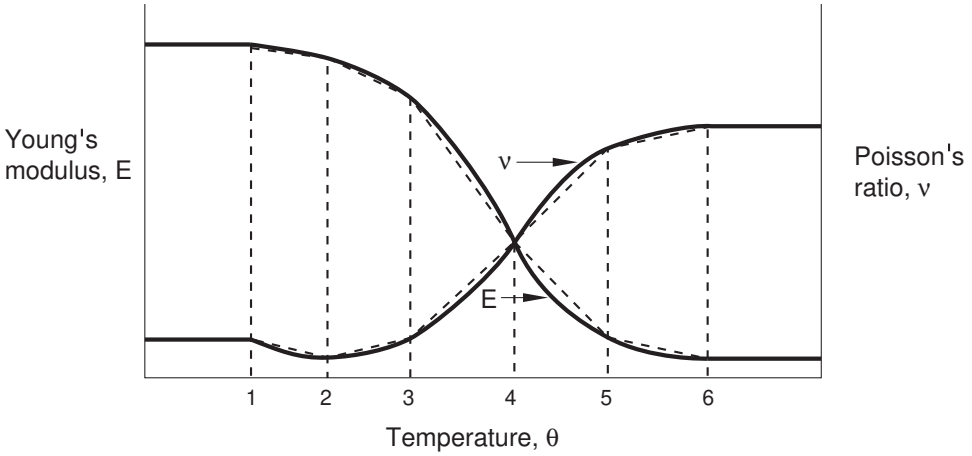


Figure 1: Example of material definition.

In this case six sets of values are used to specify the material description, as shown in the following table:

Elastic Modulus	Poisson's Ratio	Temperature
E_1	ν_1	θ_1
E_2	ν_2	θ_2
E_3	ν_3	θ_3
E_4	ν_4	θ_4
E_5	ν_5	θ_5
E_6	ν_6	θ_6

For temperatures that are outside the range defined by θ_1 and θ_6, Abaqus assumes constant values for E and ν. The dotted lines on the graph represent the straight-line approximations that will be used for this model. In this example only one value of the thermal expansion coefficient is given, α_1, and it is independent of temperature.

Example: Elastic-Plastic Material

Figure 2 shows an elastic-plastic material for which the yield stress depends on the equivalent plastic strain and temperature.

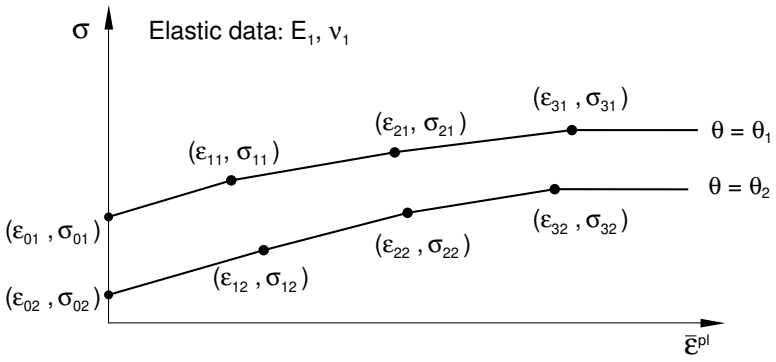


Figure 2: Example of material definition with two independent variables.

In this case the second independent variable (temperature) must be held constant, while the yield stress is described as a function of the first independent variable (equivalent plastic strain). Then, a higher value of temperature is chosen and the dependence on equivalent plastic strain is given at this temperature. This process, as shown in the following table, is repeated as often as necessary to describe the property variations in as much detail as required:

Yield Stress	Equivalent Plastic Strain	Temperature
σ_{01}	ε_{01}	θ_1
σ_{11}	ε_{11}	θ_1
σ_{21}	ε_{21}	θ_1
σ_{31}	ε_{31}	θ_1
σ_{02}	ε_{02}	θ_2
σ_{12}	ε_{12}	θ_2
σ_{22}	ε_{22}	θ_2
σ_{32}	ε_{32}	θ_2

Extrapolation of Material Data

By default, the property is assumed to be constant outside the range of independent variables given (except for fabric materials, where the property is extrapolated linearly outside the specified range). For some properties, Abaqus allows you to choose the extrapolation method outside the specified range. In these cases, you can specify if the value of the property is constant or is extrapolated linearly outside the range. If the latter method is selected, Abaqus extrapolates the property linearly using the slope at the last specified data point.

Example: Extrapolation for Elastic-Plastic Material with Isotropic Hardening

In *Table 1* the Mises isotropic yield surface is defined by providing three data points.

Table 1: Isotropic hardening data.

Yield Stress	Equivalent Plastic Strain
σ_0^Y	0
σ_1^Y	$\bar{\varepsilon}_1^{pl}$
σ_2^Y	$\bar{\varepsilon}_2^{pl}$

The response of the material outside the range of equivalent plastic strain ($\bar{\varepsilon}^{pl} > \bar{\varepsilon}_2^{pl}$) depends on the extrapolation method specified, as shown in *Figure 3*.

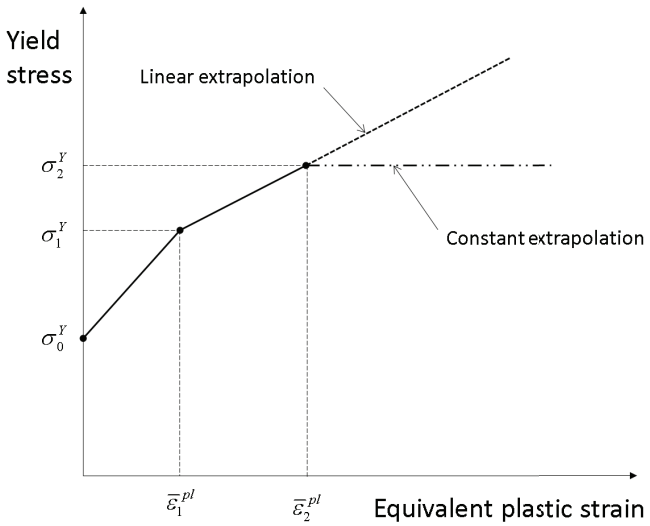


Figure 3: Material response for constant and linear extrapolations.

Specifying Field Variable Dependence

You can specify the number of user-defined field variable dependencies required for many material behaviors (see *Predefined Fields*). If you do not specify a number of field variable dependencies for a material behavior with which field variable dependence is available, the material data are assumed not to depend on field variables.

Input File Usage: **MATERIAL BEHAVIOR OPTION*, DEPENDENCIES=*n*

**MATERIAL BEHAVIOR OPTION* refers to any material behavior option for which field dependence can be specified. Each data line can hold up to eight data items. If more field variable dependencies are required than fit on a single data line, more data lines can be added. For example, a linear, isotropic elastic material can be defined as a function of temperature and seven field variables ($\boldsymbol{fv}$) as follows:

**ELASTIC*, TYPE=ISOTROPIC, DEPENDENCIES=7

E, ν, θ, $\boldsymbol{fv}_1$, $\boldsymbol{fv}_2$, $\boldsymbol{fv}_3$, $\boldsymbol{fv}_4$, $\boldsymbol{fv}_5$

$\boldsymbol{fv}_6$, $\boldsymbol{fv}_7$

This pair of data lines would be repeated as often as necessary to define the material as a function of the temperature and field variables.

Abaqus/CAE Usage: Property module: material editor: *material behavior*: **Number of field variables:** *n*

material behavior refers to any material behavior for which field dependence can be specified.

Specifying Material Data as Functions of Solution-Dependent Variables

In Abaqus you can introduce dependence on solution variables through a user subroutine or by direct specification.

User Subroutine Approach

User subroutines *USDFLD* in Abaqus/Standard and *VUSDFLD* in Abaqus/Explicit allow you to define field variables at a material point as functions of time, material directions, and any of the available material point quantities. These quantities are listed in *Abaqus/Standard Output Variable Identifiers* for *USDFLD* and in *Obtaining Material Point Information in an Abaqus/Explicit Analysis* for *VUSDFLD*. Material properties defined as functions of these field variables, therefore, can depend on the solution.

User subroutines *USDFLD* and *VUSDFLD* are called at each material point for which the material definition includes a reference to the user subroutine.

For general analysis steps the values of variables provided in user subroutines *USDFLD* and *VUSDFLD* are those corresponding to the start of the increment. Hence, the solution dependence introduced in this way is explicit; the material properties for a given increment are not influenced by the results obtained during the increment. Consequently, the accuracy of the results generally depends on the time increment size. This is not typically a concern in Abaqus/Explicit because the stable time increment is usually sufficiently small to ensure good accuracy. In Abaqus/Standard you can control the time increment from inside user subroutine *USDFLD*. For linear perturbation steps the solution variables in the base state are available. For a discussion of general and linear perturbation steps, see *General and Perturbation Procedures*.

Input File Usage: **USER DEFINED FIELD*, TYPE=USER (default)

Abaqus/CAE Usage: User subroutines *USDFLD* and *VUSDFLD* are not supported in Abaqus/CAE.

Direct Specification Approach

There are applications where it is desirable to redefine a field variable to be equal to the value of a scalar output variable. In such cases, you can avoid the overhead of writing and maintaining a user subroutine and, instead, introduce solution dependence directly by associating a field variable with an existing scalar output variable in Abaqus. For scalar output variables, you only need to specify the output variable identifier to build this association. However, for a vector or a tensor output variable, you must specify the appropriate component of the variable (for example, specify PE11 instead of PE to associate a field variable with the -11- component of the plastic strain tensor).

Any output variable that is available with the `GETVRM` or the `VGETVRM` utility routine, used to access material point information in Abaqus/Standard and Abaqus/Explicit, respectively, can be associated with a field variable.

For general analysis steps the values of the output variables are those corresponding to the start of the increment. Hence, the solution dependence introduced in this way is explicit; the material properties for a given increment are not influenced by the results obtained during the increment. Consequently, the accuracy of the results typically depends on the time increment size. This is usually not a concern in Abaqus/Explicit because the stable time increment is usually sufficiently small to ensure good accuracy. For linear perturbation steps the output variables in the base state are used. For a discussion of general and linear perturbation steps, see *General and Perturbation Procedures*.

Input File Usage: **USER DEFINED FIELD*, TYPE=SPECIFIED
field_variable_number, *output_variable_identifier (scalar values only)*

Repeat the data line as often as necessary to define more field variables.

Abaqus/CAE Usage: The direct specification approach is not supported in Abaqus/CAE.

Material Failure and Element Deletion

Abaqus offers a general framework for material failure modeling that allows the combination of multiple failure mechanisms acting simultaneously on the same material. Material failure refers to the complete loss of load-carrying capacity at a material point that results from progressive degradation of the material stiffness or instantaneous rupture based on material constitutive models or by user specification. An active material point is considered to turn inactive (deleted) on material failure. For material behaviors defined by user subroutines, you can control the status of a material point inside user subroutines using solution-dependent state variables (see *Deleting Elements from a Mesh Using State Variables*). In Abaqus/Explicit output variable STATUSMP can be used to request the status at each material point. This variable is equal to one if the material point is active and equal to zero if the material point fails.

Unless specified otherwise for particular material behaviors, an element is deleted (or removed) from a mesh on material failure. Detailed criteria for element deletion driven by material failure in Abaqus/Standard and Abaqus/Explicit are described below. Deleted elements have no ability to carry stresses and, therefore, do not contribute to the stiffness of the model. In a heat transfer analysis the thermal contribution of the element is also removed when the element deletion criterion is reached.

The status of an element can be determined by requesting output variable STATUS. This variable is equal to one if the element is active and equal to zero if the element is deleted. Elements that have been removed are not displayed when you view the deformed model in the Visualization module of Abaqus/CAE (Abaqus/Viewer). However, the elements still remain in the Abaqus model. You can choose to display removed elements by suppressing the use of the STATUS variable (see *Selecting the status field output variable*).

Alternatively, several material failure mechanisms allow you to retain fully damaged elements (along with a residual stiffness) in the computations even after material failure. For example, for progressive damage and failure modeling, you can specify the value of the maximum degradation and whether element deletion occurs when the degradation reaches this value (see *Controlling Element Deletion and Maximum Degradation for Materials with Damage Evolution*).

Element Deletion Driven by Material Failure in Abaqus/Standard

In Abaqus/Standard an element is removed from the mesh if the material fails at all the section points at all the integration locations of the element. In addition, for cohesive elements with traction-separation response, elements are deleted only if none of the integration points are in compression.

Element Deletion Driven by Material Failure in Abaqus/Explicit

In Abaqus/Explicit a general rule for element deletion from the mesh is that the material fails at all of the section points at any one integration location of an element. It is not necessary for all material points in the element to fail. The following examples describe default behavior that results in element removal:

- Solid elements: material failure at any one integration point.
- Second-order reduced-integration beam elements: material failure at all section points through the thickness at either of the two element integration locations along the beam axis.
- Modified triangular and tetrahedral solid elements and fully integrated membrane elements: material failure at any one integration point.

The general rule has two exceptions. First, the criterion for element deletion for cohesive elements is that the material fails at all integration points. In addition, for cohesive elements with traction-separation response, elements are deleted only if none of the integration points are in compression.

Second, a shell element is removed from the mesh by default when all of the active section points (that is, section points where the material has not failed) at any one integration location share the same through-the-thickness z-location. In this condition, the shell element cannot sustain any bending moment. Specifically, a shell element

is deleted when any one integration location has at most one active section point or at least two active section points and all of them share the same z-location. For example, in the case of a two-layer composite with three section points per layer, the third section point of the first layer and the first point of the second layer share the same z-location (on the interface between the two layers). The element deletion criterion is triggered if only one of the six section points is left active or if only the two points mentioned above (that share the same z-location) remain active.

Difficulties Associated with Element Removal in Abaqus/Standard

When elements are removed from the model, their nodes still remain in the model even if they are not attached to any active elements. When the solution progresses, these nodes can undergo nonphysical displacements due to the extrapolation scheme used in Abaqus/Standard to speed up the solution (see *Convergence Criteria for Nonlinear Problems*). You can prevent these nonphysical displacements by turning off the extrapolation. In addition, applying a point load to a node that is not attached to an active element causes convergence difficulties since there is no stiffness to resist the load. You should take precautions to avoid these situations.

Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit

The characteristic element length is used by Abaqus for the regularization of models that exhibit strain softening or is passed to user subroutines that are called at a material point. By default, Abaqus computes the characteristic element length using the geometric mean–based definition.

The default value for a first-order element is the typical length of a line across an element, and the default value for a second-order element is half of the same typical length. For trusses the default value is a characteristic length along the element axis. For membranes and shells the default value is a characteristic length in the reference surface. For axisymmetric elements the default value is a characteristic length in the r–z plane only.

In Abaqus/Explicit you can redefine the value of the characteristic element length based on the element topology and geometry in user subroutine *VUCHARLENGTH*.

Input File Usage: Use the following option for the geometric mean–based definition of the characteristic element length (default):

**CHARACTERISTIC LENGTH*, DEFINITION=GEOMETRIC MEAN

Use the following option to specify the characteristic element length in user subroutine *VUCHARLENGTH*:

**CHARACTERISTIC LENGTH*, DEFINITION=USER, COMPONENTS=*n*, PROPERTIES=*n*

Abaqus/CAE Usage: User subroutine *VUCHARLENGTH* is not supported in Abaqus/CAE.

Regularizing User-Defined Data in Abaqus/Explicit

Interpolating material data as functions of independent variables requires table lookups of the material data values during the analysis. The table lookups occur frequently in Abaqus/Explicit and are most economical if the interpolation is from regular intervals of the independent variables. For example, the data shown in *Figure 1* are not regular because the intervals in temperature (the independent variable) between adjacent data points vary. You are not required to specify regular material data. Abaqus/Explicit automatically regularizes user-defined data. For example, the temperature values in *Figure 1* can be defined at 10°, 20°, 25°, 28°, 30°, and 35° C. In this case Abaqus/Explicit can regularize the data by defining the data over 25 increments of 1° C and your piecewise linear data will be reproduced exactly. This regularization requires the expansion of your data from

values at 6 temperature points to values at 26 temperature points. This example is a case where a simple regularization can reproduce your data exactly.

If there are multiple independent variables, the concept of regular data also requires that the minimum and maximum values (the range) be constant for each independent variable while specifying the other independent variables. The material definition in *Figure 2* illustrates a case where the material data are not regular since $\epsilon_{11} \neq \epsilon_{12}$, $\epsilon_{21} \neq \epsilon_{22}$, and $\epsilon_{31} \neq \epsilon_{32}$. Abaqus/Explicit will also regularize data involving multiple independent variables, although the data provided must satisfy the rules specified in *Input Syntax Rules*.

Error Tolerance Used in Regularizing User-Defined Data

It is not always desirable to regularize the input data so that they are reproduced exactly in a piecewise linear manner. Suppose that the yield stress is defined as a function of plastic strain in Abaqus/Explicit as follows:

Yield Stress	Plastic Strain
50000	.0
75000	.001
80000	.003
85000	.010
86000	1.0

It is possible to regularize the data exactly but it is not very economical, since it requires the subdivision of the data into 1000 regular intervals. Regularization is more difficult if the smallest interval you defined is small compared to the range of the independent variable.

Abaqus/Explicit uses an error tolerance to regularize the input data. The number of intervals in the range of each independent variable is chosen such that the error between the piecewise linear regularized data and each of your defined points is less than the tolerance times the range of the dependent variable. In some cases the number of intervals becomes excessive and Abaqus/Explicit cannot regularize the data using a reasonable number of intervals. The number of intervals considered reasonable depends on the number of intervals that you define. If you defined 50 or less intervals, the maximum number of intervals used by Abaqus/Explicit for regularization is equal to 100 times the number of user-defined intervals. If you defined more than 50 intervals, the maximum number of intervals used for regularization is equal to 5000 plus 10 times the number of user-defined intervals above 50. If the number of intervals becomes excessive, the program stops during the data checking phase and issues an error message. You can either redefine the material data or change the tolerance value. The default tolerance is 0.03.

The yield stress data in the example above are a typical case where such an error message might be issued. In this case you can simply remove the last data point since it produces only a small difference in the ultimate yield value.

Input File Usage: **MATERIAL*, RTOL=*tolerance*

Abaqus/CAE Usage: Property module: material editor: **General->Regularization**: **Rtol:** *tolerance*

Interpolation of Strain Rate–Dependent Data in Abaqus/Explicit

Because strain rate dependence of data is usually measured at logarithmic intervals, Abaqus/Explicit interpolates strain rate data using logarithmic intervals rather than uniformly spaced intervals by default. This generally provides a better match to typical strain rate–dependent curves. You can specify linear strain rate interpolation to use uniform intervals for interpolation of strain rate data. The use of linear strain rate interpolation affects

only the interpolation of strain rate as an independent variable and is relevant only if one of the following behaviors is used to define the material data:

- low-density foams (*Low-Density Foams*)
- rate-dependent metal plasticity (*Classical Metal Plasticity*)
- rate-dependent viscoplasticity defined by yield stress ratios (*Rate-Dependent Yield*)
- shear failure defined using direct tabular data (*Dynamic Failure Models*)
- rate-dependent Drucker-Prager hardening (*Extended Drucker-Prager Models*)
- rate-dependent concrete damaged plasticity (*Concrete Damaged Plasticity*)
- rate-dependent damage initiation criterion (*Damage Initiation for Ductile Metals*)

Input File Usage: Use the following option to specify logarithmic interpolation (default):

**MATERIAL*, STRAIN RATE INTERPOLATION=LOGARITHMIC

Use the following option to specify linear interpolation:

**MATERIAL*, STRAIN RATE INTERPOLATION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **General->Regularization**: **Strain rate regularization**: **Logarithmic** or **Linear**

Evaluation of Strain Rate–Dependent Data in Abaqus/Explicit

Rate-sensitive material constitutive behavior can introduce nonphysical high-frequency oscillations in an explicit dynamic analysis. To overcome this problem, Abaqus/Explicit computes the equivalent plastic strain rate used for the evaluation of strain rate–dependent data as

$$\dot{\bar{\varepsilon}}^{pl}\Big|_{t+\Delta t} = \omega\ \frac{\Delta\bar{\varepsilon}^{pl}}{\Delta t} + (1-\omega)\ \dot{\bar{\varepsilon}}^{pl}\Big|_{t}.$$

Here $\Delta\bar{\varepsilon}^{pl}$ is the incremental change in equivalent plastic strain during the time increment Δt, and $\dot{\bar{\varepsilon}}^{pl}\big|_{t}$ and $\dot{\bar{\varepsilon}}^{pl}\big|_{t+\Delta t}$ are the strain rates at the beginning and end of the increment, respectively. The factor ω ($0 < \omega \le 1$) facilitates filtering high-frequency oscillations associated with strain rate–dependent material behavior.

Abaqus/Explicit provides two options to specify the value of the strain rate factor, ω:

- You can specify a constant value of ω. The default value is 0.9. A value of $\omega = 1$ does not provide the desired filtering effect; therefore, you should avoid this setting.
- You can define a time-based filter by specifying a time constant, μ. In this case, the value of ω is computed as $\omega = 1/(1 + \mu/\Delta t)$. This method produces filtered results that are less sensitive to the changes in time incrementation used for the analysis.

Input File Usage: To define a constant filter factor:

**MATERIAL*, SRATE FACTOR=ω

To define a time-based filter:

**MATERIAL*, SRATE TIME=μ

Abaqus/CAE Usage: You cannot specify the value of the strain rate factor in Abaqus/CAE.

Combining Material Behaviors

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Material Data Definition*
- *Creating materials*

Overview

Some general rules apply for combining material behaviors in Abaqus.

Abaqus provides a broad range of possible material behaviors. A material is defined by choosing the appropriate behaviors for the purpose of an analysis. This section describes the general rules for combining material behaviors. Specific information for each material behavior is also summarized at the end of each material behavior description section in this chapter.

Some of the material behaviors in Abaqus are completely unrestricted: they can be used alone or together with other behaviors. For example, thermal properties such as conductivity can be used in any material definition. They are used in an analysis if the material is associated with elements that can solve heat transfer problems and if the analysis procedure allows for the thermal equilibrium equation to be solved.

Some material behaviors in Abaqus require the presence of other material behaviors, and some exclude the use of other material behaviors. For example, metal plasticity requires the definition of elastic material behavior or an equation of state and excludes all other rate-independent plasticity behaviors.

Complete Material Definitions

Abaqus requires that the material be sufficiently defined to provide suitable properties for those elements with which the material is associated and for all analysis procedures through which the model will be run. Thus, a material associated with displacement or structural elements must include either a “Complete mechanical” category behavior or an “Elasticity” category behavior, as discussed below. In Abaqus/Explicit density (*Density*) is required for all materials except hydrostatic fluids.

It is impossible to modify or add to material definitions once an analysis is started. However, material definitions can be modified in an import analysis. For example, a static analysis can be run in Abaqus/Standard using a material definition that does not include a density specification. Density can be added to the material definition when the analysis is imported into Abaqus/Explicit.

All aspects of a material's behavior need not be fully defined; any behavior that is omitted is assumed not to exist in that part of the model. For example, if elastic material behavior is defined for a metal but metal plasticity is not defined, the material is assumed not to have a yield stress. You must ensure that the material is adequately defined for the purpose of the analysis. The material can include behaviors that are not relevant for the analysis, as described in *About the Material Library*. Thus, you can include general material behavior libraries, without having to delete those behaviors that are not required for a particular application. This generality offers great flexibility in material modeling.

In Abaqus/Standard any material behaviors defined using a distribution (*Distribution Definition*) can be combined with almost all material behaviors in a manner identical to how they are combined when no distributions are used. For example, if the linear elastic material behavior is defined using a distribution, it can be combined with

metal plasticity or any other material behavior that can normally be combined with linear elastic behavior. In addition, more than one material behavior defined with a distribution (linear elastic behavior and thermal expansion, for example) can be included in the same material definition. The only exception is that a material defined with concrete damaged plasticity (*Concrete Damaged Plasticity*) cannot have any material behaviors defined with a distribution.

Material Behavior Combination Tables

The material behavior combination tables that follow explain which behaviors must be used together. The tables also show the material behaviors that cannot be combined. Behaviors designated with an (S) are available only in Abaqus/Standard; behaviors designated with an (E) are available only in Abaqus/Explicit.

The behaviors are assigned to categories because exclusions are best described in terms of those categories. Some of the categories require explanation:

- "Complete mechanical behaviors" are those behaviors in Abaqus that, individually, completely define a material's mechanical (stress-strain) behavior. A behavior in this category, therefore, excludes any other such behavior and also excludes any behavior that defines part of a material's mechanical behavior: those behaviors that belong to the elasticity and plasticity categories.
- "Elasticity, fabric, and equation of state behaviors" contains all the basic elasticity behaviors in Abaqus. If a behavior from the "Complete mechanical behaviors" category is not used and mechanical behavior is required, a behavior must be selected from this category. This selection then excludes any other elasticity behavior.
- "Enhancements for elasticity behaviors" contains behaviors that extend the modeling provided by the elasticity behaviors in Abaqus.
- "Rate-independent plasticity behaviors" contains all the basic plasticity behaviors in Abaqus except deformation plasticity, which is in the "Complete mechanical behaviors" category because it completely defines the material's mechanical behavior.
- "Rate-dependent plasticity behaviors" contains behaviors that extend the modeling provided by the rate-independent plasticity behaviors and by the linear elastic material behavior.

If elastic-plastic behavior must be modeled, you should select an appropriate plasticity behavior from one of the plasticity behaviors categories and an elasticity behavior from one of the elasticity behaviors categories.

General Behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Material damping	**DAMPING*	Elasticity, fabric, equation of state (except ignition and growth equation of state), hyperelasticity, hyperfoam, low-density foam, or anisotropic hyperelasticity (except when used with beam or shell general sections or substructures)
Density	**DENSITY*	Required in Abaqus/Explicit, except for hydrostatic fluid elements
Solution-dependent state variables	**DEPVAR*	
Thermal expansion	**EXPANSION*	

Complete Mechanical Behaviors:

These behaviors are mutually exclusive and exclude all behaviors listed for elasticity, plasticity, and hydrostatic fluid behaviors, including all related enhancements. However, when an element is enriched (*Modeling*

Discontinuities as an Enriched Feature Using the Extended Finite Element Method), the mechanical user material behavior can be used with the progressive damage and failure behavior, which is in the enhancement for elasticity or plasticity behaviors category.

Behavior	Keyword	Requires
Acoustic medium	**ACOUSTIC MEDIUM*	Density
Deformation plasticity(S)	**DEFORMATION PLASTICITY*	
Mechanical user material	**USER MATERIAL*, TYPE=MECHANICAL	

Elasticity, Fabric, and Equation of State Behaviors:

These behaviors are mutually exclusive.

Behavior	Keyword	Requires
Elasticity	**ELASTIC*	
Equation of state(E)	**EOS*	
Fabric(E)	**FABRIC*	
Hyperelasticity	**HYPERELASTIC*	
Hyperfoam	**HYPERFOAM*	
Anisotropic hyperelasticity	**ANISOTROPIC HYPERELASTIC*	
Hypoelasticity(S)	**HYPOELASTIC*	
Porous elasticity (S)	**POROUS ELASTIC*	
Low-density foam (E)	**LOW DENSITY FOAM*	

Enhancements for Elasticity Behaviors:

Behavior	Keyword	Requires
Elastic shear behavior for an equation of state(E)	**ELASTIC*, TYPE=SHEAR	Equation of state
Strain-based failure measures	**FAIL STRAIN*	Elasticity
Stress-based failure measures	**FAIL STRESS*	Elasticity
Hysteresis(S)	**HYSTERESIS*	Hyperelasticity (excludes all plasticity behaviors and Mullins effect)
Mullins effect	**MULLINS EFFECT*	Hyperelasticity (excludes hysteresis), hyperfoam, or anisotropic hyperelasticity
Compressive failure theory(S)	**NO COMPRESSION*	Elasticity
Tension failure theory(S)	**NO TENSION*	Elasticity
Paperboard ZD compression elasticity	**PAPERBOARD THICKNESS COMPRESSION ELASTIC*	Orthotropic elasticity with decoupled in-plane and out-of-plane responses
Viscoelasticity	**VISCOELASTIC*	Elasticity, hyperelasticity, or hyperfoam (excludes all plasticity behaviors and all associated plasticity enhancements); or anisotropic hyperelasticity
Shear viscosity for an equation of state(E)	**VISCOSITY*	Equation of state

Rate-Independent Plasticity Behaviors:

These behaviors are mutually exclusive.

Behavior	Keyword	Requires
Brittle cracking(E)	**BRITTLE CRACKING*	Isotropic elasticity and brittle shear
Modified Drucker-Prager/Cap plasticity	**CAP PLASTICITY*	Drucker-Prager/Cap plasticity hardening and isotropic elasticity or porous elasticity
Cast iron plasticity	**CAST IRON PLASTICITY*	Cast iron compression hardening, cast iron tension hardening, and isotropic elasticity
Cam-clay plasticity	**CLAY PLASTICITY*	Isotropic elasticity, orthotropic elasticity, or porous elasticity (in Abaqus/Standard)
		Isotropic elasticity or orthotropic elasticity (in Abaqus/Explicit)
Concrete(S)	**CONCRETE*	Isotropic elasticity
Concrete damaged plasticity	**CONCRETE DAMAGED PLASTICITY*	Concrete compression hardening, concrete tension stiffening, and isotropic elasticity
Crushable foam plasticity	**CRUSHABLE FOAM*	Crushable foam hardening and isotropic elasticity or orthotropic elasticity
Drucker-Prager plasticity	**DRUCKER PRAGER*	Drucker-Prager hardening and isotropic elasticity, orthotropic elasticity, or porous elasticity (in Abaqus/Standard)
		Drucker-Prager hardening and isotropic elasticity, orthotropic elasticity, or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)
Plastic compaction behavior for an equation of state(E)	**EOS COMPACTION*	Linear $U_s - U_p$ equation of state
Jointed material(S)	**JOINTED MATERIAL*	Isotropic elasticity and a local orientation
Mohr-Coulomb plasticity	**MOHR COULOMB*	Mohr-Coulomb hardening and isotropic elasticity
Paperboard in-plane plasticity	**PAPERBOARD PLASTICITY*	Orthotropic elasticity with decoupled in-plane and out-of-plane responses
Paperboard thickness compression plasticity	**PAPERBOARD THICKNESS COMPRESSION PLASTICITY*	Orthotropic elasticity with decoupled in-plane and out-of-plane responses, and paperboard thickness compression elasticity
Paperboard out-of-plane shear plasticity	**PAPERBOARD TRANSVERSE SHEAR PLASTICITY*	Orthotropic elasticity with decoupled in-plane and out-of-plane responses
Metal plasticity	**PLASTIC*	Elasticity or hyperelasticity (in Abaqus/Standard)
		Isotropic elasticity, orthotropic elasticity (requires anisotropic yield), hyperelasticity, or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)

Rate-Dependent Plasticity Behaviors:

These behaviors are mutually exclusive, except metal creep and time-dependent volumetric swelling.

Behavior	Keyword	Requires
Cap creep(S)	**CAP CREEP*	Elasticity, modified Drucker-Prager/Cap plasticity, and Drucker-Prager/Cap plasticity hardening

Behavior	Keyword	Requires
Metal creep[(S)]	**CREEP*	Elasticity (except when used to define rate-dependent gasket behavior; excludes all rate-independent plasticity behaviors except metal plasticity)
Drucker-Prager creep[(S)]	**DRUCKER PRAGER CREEP*	Elasticity, Drucker-Prager plasticity, and Drucker-Prager hardening
Metal plasticity	**PLASTIC*, RATE	Elasticity or hyperelasticity (in Abaqus/Standard)
		Isotropic elasticity, orthotropic elasticity (requires anisotropic yield), hyperelasticity, or the combination of an equation of state and isotropic linear elastic shear behavior for an equation of state (in Abaqus/Explicit)
Nonlinear viscoelasticity	**VISCOELASTIC*, NONLINEAR	Hyperelasticity
Rate-dependent viscoplasticity	**RATE DEPENDENT*	Drucker-Prager plasticity, crushable foam plasticity, or metal plasticity
Time-dependent volumetric swelling[(S)]	**SWELLING*	Elasticity (excludes all rate-independent plasticity behaviors except metal plasticity)
Two-layer viscoplasticity[(S)]	**VISCOUS*	Elasticity and metal plasticity

Enhancements for Plasticity Behaviors:

Behavior	Keyword	Requires
Annealing temperature	**ANNEAL TEMPERATURE*	Metal plasticity
Brittle failure[(E)]	**BRITTLE FAILURE*	Brittle cracking and brittle shear
Cyclic hardening	**CYCLIC HARDENING*	Metal plasticity with nonlinear isotropic/kinematic hardening
Inelastic heat fraction	**INELASTIC HEAT FRACTION*	Metal plasticity and specific heat
Oak Ridge National Laboratory constitutive model[(S)]	**ORNL*	Metal plasticity, cycled yield stress data, and, usually, metal creep
Porous material failure criteria[(E)]	**POROUS FAILURE CRITERIA*	Porous metal plasticity
Porous metal plasticity	**POROUS METAL PLASTICITY*	Metal plasticity
Hill anisotropic yield/creep	**POTENTIAL*, TYPE=HILL	Metal plasticity, metal creep, Cam-clay plasticity, crushable foam plasticity, Drucker-Prager plasticity, or two-layer viscoplasticity
Nonquadratic yield[(E)]	**POTENTIAL* (except TYPE=HILL)	Metal plasticity
Shear failure[(E)]	**SHEAR FAILURE*	Metal plasticity
Tension cutoff	**TENSION CUTOFF*	Mohr-Coulomb plasticity

Enhancement for Elasticity or Plasticity Behaviors:

Behavior	Keyword	Requires
Tensile failure[(E)]	**TENSILE FAILURE*	Metal plasticity or equation of state
Damage initiation	**DAMAGE INITIATION*	For elasticity behaviors: elasticity based on a traction-separation description for cohesive elements or elasticity model for fiber-reinforced composites For plasticity behaviors: elasticity and metal plasticity or Drucker-Prager plasticity
Damage evolution	**DAMAGE EVOLUTION*	Damage initiation

Behavior	Keyword	Requires
Damage stabilization	**DAMAGE STABILIZATION*	Damage evolution
Transverse shear	**TRANSVERSE SHEAR*	Mechanical behaviors associated with shell or beam elements

Thermal Behaviors:

The following behavior is unrestricted:

Behavior	Keyword	Requires
Volumetric heat generation	**HEAT GENERATION*	

The following behaviors are unrestricted but exclude thermal user materials:

Behavior	Keyword	Requires
Thermal conductivity	**CONDUCTIVITY*	
Latent heat	**LATENT HEAT*	Density
Specific heat	**SPECIFIC HEAT*	Density

Complete Thermal Behavior:

This behavior is unrestricted but excludes the thermal behaviors in the previous table.

Behavior	Keyword	Requires
Thermal user material	**USER MATERIAL*, TYPE=THERMAL	Density

Pore Fluid Flow Behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Swelling gel$^{(S)}$	**GEL*	Permeability, porous bulk moduli, and absorption/exsorption behavior
Moisture-driven swelling$^{(S)}$	**MOISTURE SWELLING*	Permeability and absorption/exsorption behavior
Permeability$^{(S)}$	**PERMEABILITY*	
Porous bulk moduli$^{(S)}$	**POROUS BULK MODULI*	Permeability and either elasticity or porous elasticity
Absorption/exsorption behavior$^{(S)}$	**SORPTION*	Permeability

Electrical Behaviors:

These behaviors are unrestricted.

Behavior	Keyword	Requires
Dielectricity$^{(S)}$	**DIELECTRIC*	
Fraction of electric energy released as heat$^{(S)}$	**JOULE HEAT FRACTION*	

Behavior	Keyword	Requires
Piezoelectricity(S)	*PIEZOELECTRIC	

These behaviors are mutually exclusive.

Behavior	Keyword	Requires
Electrical conductivity(S)	*ELECTRICAL CONDUCTIVITY	
Electrical resistivity(S)	*ELECTRICAL RESISTIVITY	

Enhancements for Electrical Behaviors:

Behavior	Keyword	Requires
Piezoelectricity(S)	*PIEZOELECTRIC	*ELECTRICAL RESISTIVITY

Mass Diffusion Behaviors:

These behaviors exclude all other behaviors.

Behavior	Keyword	Requires
Mass diffusivity(S)	*DIFFUSIVITY	Solubility
Solubility(S)	*SOLUBILITY	Mass diffusivity

Hydrostatic Fluid Behaviors:

Behavior	Keyword	Requires
Fluid bulk modulus(S)	*FLUID BULK MODULUS	Hydraulic fluid
Hydrostatic fluid density	*FLUID DENSITY	
Fluid thermal expansion coefficient(S)	*FLUID EXPANSION	Hydraulic fluid

Density

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- **DENSITY*
- *Specifying material mass density*

Overview

A material's mass density:

- must be defined in Abaqus/Standard for eigenfrequency and transient dynamic analysis, transient heat transfer analysis, adiabatic stress analysis, and acoustic analysis;
- must be defined in Abaqus/Standard for gravity, centrifugal, and rotary acceleration loading;
- must be defined in Abaqus/Explicit for all materials except hydrostatic fluids;
- can be specified as a function of temperature and predefined variables;
- can be distributed from nonstructural features (such as paint on sheet metal panels in a car) to the underlying elements using a nonstructural mass definition; and
- can be defined with a distribution for solid continuum elements in Abaqus/Standard.

Defining Density

Density can be defined as a function of temperature and field variables. Based on user-defined data Abaqus internally estimates the material density as follows:

- For acoustic, heat transfer, and coupled thermal-electrical elements in Abaqus/Standard and acoustic elements in Abaqus/Explicit, the density is continually updated to the value corresponding to the current temperature and field variables.
- For coupled temperature-displacement elements in Abaqus/Standard, the density used in both heat transfer and structural body force computations is assumed to be only a function of the initial temperature and field variables and changes in volume. This ensures mass conservation during the structural body force computations.
- For all other elements in Abaqus/Standard and Abaqus/Explicit, the density is taken to be a function of the initial temperature and field variables and changes in volume only. It is not updated if temperatures and field variables change during the analysis.

In an Abaqus/Standard analysis, a spatially varying mass density can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include a default value for the density. If a distribution is used, no dependencies on temperature or field variables for the density can be defined.

Input File Usage: Use either of the following options:

**DENSITY*
**DENSITY*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **General->Density**

You can select **Use temperature-dependent data** to define the density as a function of temperature or select the **Number of field variables** to define the density as a function of field variables.

Units

Since Abaqus has no built-in dimensions, you must ensure that the density is given in consistent units. The use of consistent units, and density in particular, is discussed in *Conventions*. If American or English units are used, you must be particularly careful that the density used is in units of $\mathbf{ML^{-3}}$, where mass is defined in units of $\mathbf{FT^2L^{-1}}$.

Elements

The density behavior described in this section is used to specify mass density for all elements, except rigid elements. Mass density for rigid elements is specified as part of the rigid body definition (see *Rigid Elements*).

In Abaqus/Explicit, you must define a nonzero mass density for all elements that are not part of a rigid body. In an undrained pore fluid flow and stress analysis, you must specify the density of the pore fluid for the porous medium if the elements are used with a permeating fluid.

In Abaqus/Standard density must be defined for heat transfer elements and acoustic elements; mass density can be defined for stress/displacement elements, coupled temperature-displacement elements, and elements including pore pressure. For elements that include pore pressure as a degree of freedom, the density of the dry material should be given for the porous medium in a coupled pore fluid flow/stress analysis.

If you have a complex density for an acoustic medium, you should enter its real part here and convert the imaginary part into a volumetric drag, as discussed in *Acoustic Medium*.

The mass contribution from features that have negligible structural stiffness can be added to the model by smearing the mass over an element set that is typically adjacent to the nonstructural feature. The nonstructural mass can be specified in the form of a total mass value, a mass per unit volume, a mass per unit area, or a mass per unit length (see *Nonstructural Mass Definition*). A nonstructural mass definition contributes additional mass to the specified element set and does not alter the underlying material density.

Elastic Mechanical Properties

Abaqus provides several models to define the elastic behavior of the material.

In this section:

- *Elastic Behavior*
- *Linear Elasticity*
- *Elastic Behavior of Porous Materials*
- *Hypoelastic Behavior*
- *Hyperelasticity*
- *Stress Softening in Elastomers*
- *Linear Viscoelasticity*
- *Nonlinear Viscoelasticity*
- *Low-Density Foams*
- *Superelasticity*

Elastic Behavior

Overview

The material library in Abaqus includes several models of elastic behavior.

Linear elasticity

Linear elasticity (*Linear Elastic Behavior*) is the simplest form of elasticity available in Abaqus. The linear elastic model can define isotropic, orthotropic, or anisotropic material behavior and is valid for small elastic strains.

Plane stress orthotropic failure

Failure measures (*Plane Stress Orthotropic Failure Measures*) are provided for use with linear elasticity as indicators of material failure. They can be used to obtain postprocessed output requests based on the evaluation of common failure theories for fiber-reinforced composite materials.

Porous elasticity

The porous elastic model in Abaqus/Standard (*Elastic Behavior of Porous Materials*) is used to study porous materials with nonlinear pressure-dependent elastic behavior (including logarithmic or power laws). This form of nonlinear elasticity is valid for small elastic strains.

Hypoelasticity

The hypoelastic model in Abaqus/Standard (*Hypoelastic Behavior*) is used for materials in which the rate of change of stress is defined by an elasticity matrix multiplying the rate of change of elastic strain, where the elasticity matrix is a function of the total elastic strain. This general, nonlinear elasticity is valid for small elastic strains.

Rubberlike hyperelasticity

The hyperelastic model for rubberlike materials (*Hyperelastic Behavior of Rubberlike Materials*) provides a general capability for modeling the behavior of nearly incompressible elastomers under large elastic deformations.

Foam hyperelasticity

The hyperfoam model (*Hyperelastic Behavior in Elastomeric Foams*) provides a general capability for elastomeric compressible foams at finite strains. This nonlinear elasticity model is valid for large strains (especially large volumetric changes). The low-density foam model (*Low-Density Foams*) is a nonlinear viscoelastic model suitable for specifying strain-rate sensitive behavior of low-density elastomeric foams such as used in crash and impact applications. The crushable foam plasticity model (*Crushable Foam Plasticity Models*) is used to model compressible foam materials that undergo permanent deformation.

Anisotropic hyperelasticity

The anisotropic hyperelastic model (*Anisotropic Hyperelastic Behavior*) provides a general capability for modeling materials that exhibit highly anisotropic and nonlinear elastic behavior (such as biomedical soft tissues and fiber-reinforced elastomers). The model is valid for large elastic strains and captures the changes in the preferred material directions (or fiber directions) with deformation.

Fabric materials

The fabric model in Abaqus/Explicit (*Fabric Material Behavior*) for woven fabrics captures the directional nature of the stiffness along the fill and the warp yarn directions. It also captures the shear response as the yarn directions rotate relative to each other. The model takes into account finite strains including large shear rotations. It captures the highly nonlinear elastic response of fabrics through the use of test data or a user subroutine, *VFABRIC* (see *VFABRIC*) for the material characterization. The test data based fabric behavior can include nonlinear elasticity, permanent deformation, rate-dependent response, and damage accumulation.

Viscoelasticity

The viscoelastic model is used to specify time-dependent material behavior (*Time Domain Viscoelasticity*). In Abaqus/Standard it is also used to specify frequency-dependent material behavior (*Frequency Domain Viscoelasticity*). It must be combined with linear elasticity, rubberlike hyperelasticity, or foam hyperelasticity.

Parallel rheological framework

The parallel rheological framework (*Parallel Rheological Framework*) is intended for modeling nonlinear behavior for materials subjected to large strains, such as elastomers and polymers. The models defined within this framework consist of multiple parallel viscoelastic networks and, optionally, an elastoplastic network to allow modeling permanent set and material softening using the Mullins effect. The elastic response is defined using the hyperelastic material model; the plastic response is based on the theory of incompressible isotropic hardening plasticity; and the viscous response is specified using the flow rule derived from a creep potential.

Hysteresis

The hysteresis model in Abaqus/Standard (*Hysteresis in Elastomers*) is used to specify rate-dependent behavior of elastomers. It is used in conjunction with hyperelasticity.

Mullins effect

The Mullins effect model (*Stress Softening in Rubberlike Materials*) is used to specify stress softening of filled rubber elastomers due to damage, a phenomenon referred to as Mullins effect. The model can also be used to include permanent energy dissipation and stress softening effects in elastomeric foams (*Stress Softening in Elastomeric Foams*). It is used in conjunction with rubberlike hyperelasticity or foam hyperelasticity.

No compression or no tension elasticity

The no compression and no tension models (*No Compression or No Tension*) can be used when compressive or tensile principal stresses should not be generated. These options can be used only with linear elasticity.

References:

Thermal Strain

Thermal expansion can be introduced for any of the elasticity or fabric models (*Thermal Expansion*).

Elastic Strain Magnitude

Except in the hyperelasticity and fabric material models, the stresses are always assumed to be small compared to the tangent modulus of the elasticity relationship; that is, the elastic strain must be small (less than 5%). The total strain can be arbitrarily large if inelastic response such as metal plasticity is included in the material definition.

For finite-strain calculations where the large strains are purely elastic, the fabric model (for woven fabrics), the hyperelastic model (for rubberlike behavior), or the foam hyperelasticity model (for elastomeric foams) should be used. The hyperelasticity and fabric models are the only models that give realistic predictions of actual material behavior at large elastic strains. The linear or, in Abaqus/Standard, porous elasticity models are appropriate in other cases where the large strains are inelastic.

In Abaqus/Standard the linear elastic, porous elastic, and hypoelastic models will exhibit poor convergence characteristics if the stresses reach levels of 50% or more of the elastic moduli; this limitation is not serious in practical cases because these material models are not valid for the resulting large strains.

Linear Elasticity

Linear elasticity is the simplest form of elasticity available in Abaqus. The linear elastic model can define isotropic, orthotropic, or anisotropic material behavior and is valid for small elastic strains.

In this section:

- *Linear Elastic Behavior*
- *No Compression or No Tension*
- *Plane Stress Orthotropic Failure Measures*
- *Plasticity Corrections*

Linear Elastic Behavior

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- **ELASTIC*
- *Creating a linear elastic material model*

Overview

A linear elastic material model:

- is valid for small elastic strains (normally less than 5%);
- can be isotropic, transversely isotropic, orthotropic, or fully anisotropic;
- can have properties that depend on temperature and other field variables; and
- can be defined with a distribution for solid continuum elements in Abaqus/Standard.

Defining Linear Elastic Material Behavior

The total stress is defined from the total elastic strain as

$$\boldsymbol{\sigma} = \mathbf{D}^{el}\boldsymbol{\varepsilon}^{el},$$

where $\boldsymbol{\sigma}$ is the total stress ("true," or Cauchy stress in finite-strain problems), $\mathbf{D}^{el}$ is the fourth-order elasticity tensor, and $\boldsymbol{\varepsilon}^{el}$ is the total elastic strain (log strain in finite-strain problems). Do not use the linear elastic material definition when the elastic strains might become large; use a hyperelastic model instead. Even in finite-strain problems the elastic strains should still be small (less than 5%).

Defining Linear Elastic Response for Viscoelastic Materials

The elastic response of a viscoelastic material (*Time Domain Viscoelasticity*) can be specified by defining either the instantaneous response or the long-term response of the material.

Instantaneous Response

To define the instantaneous response, experiments to determine the elastic constants must be performed within time spans much shorter than the characteristic relaxation time of the material.

Input File Usage: **ELASTIC*, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Moduli time scale (for viscoelasticity)**: **Instantaneous**

Long-Term Response

However, if the long-term elastic response is used, data from experiments must be collected after time spans much longer than the characteristic relaxation time of the viscoelastic material. Long-term elastic response is the default elastic material behavior.

Input File Usage: **ELASTIC*, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Moduli time scale (for viscoelasticity)**: **Long-term**

Directional Dependence of Linear Elasticity

Depending on the number of symmetry planes for the elastic properties, a material can be classified as either *isotropic* (an infinite number of symmetry planes passing through every point) or *anisotropic* (no symmetry planes). Some materials have a restricted number of symmetry planes passing through every point; for example, *orthotropic* materials have two orthogonal symmetry planes for the elastic properties. The number of independent components of the elasticity tensor $\mathbf{D}^{el}$ depends on such symmetry properties. You define the level of anisotropy and method of defining the elastic properties, as described below. If the material is anisotropic, a local orientation (*Orientations*) must be used to define the direction of anisotropy.

Stability of a Linear Elastic Material

Linear elastic materials must satisfy the conditions of material or Drucker stability (see the discussion on material stability in *Hyperelastic Behavior of Rubberlike Materials*). Stability requires that the tensor $\mathbf{D}^{el}$ be positive definite, which leads to certain restrictions on the values of the elastic constants. The stress-strain relations for several different classes of material symmetries are given below. The appropriate restrictions on the elastic constants stemming from the stability criterion are also given.

Defining Isotropic Elasticity

The simplest form of linear elasticity is the isotropic case, and the stress-strain relationship is given by

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \begin{bmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

The elastic properties are completely defined by giving the Young's modulus, E, and the Poisson's ratio, ν. The shear modulus, G, can be expressed in terms of E and ν as $G = E/2(1+\nu)$. These parameters can be given as functions of temperature and of other predefined fields, if necessary.

In Abaqus/Standard spatially varying isotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for E and ν. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

For orthotropic elasticity under plane stress conditions, Abaqus/Standard computes the out-of-plane strain based on the incompressibility assumption.

Input File Usage: **ELASTIC*, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Isotropic**

Stability

The stability criterion requires that $E > 0$, $G > 0$, and $-1 < \nu < 0.5$. Values of Poisson's ratio approaching 0.5 result in nearly incompressible behavior. With the exception of plane stress cases (including membranes and shells) or beams and trusses, such values require the use of "hybrid" elements in Abaqus/Standard and generate high frequency noise and result in excessively small stable time increments in Abaqus/Explicit.

In Abaqus/Standard it is recommended that you use solid continuum hybrid elements for linear elastic materials with Poisson's ratio greater than 0.495 (that is, the ratio of K/μ greater than 100) to avoid potential convergence problems. Otherwise, the analysis preprocessor will issue an error. You can use the "nonhybrid incompressible" diagnostics control to downgrade this error to a warning message.

Input File Usage: Use the following option to downgrade an error message to a warning message:

**DIAGNOSTICS*, NONHYBRID INCOMPRESSIBLE=WARNING

Defining Orthotropic Elasticity by Specifying the Engineering Constants

Linear elasticity in an orthotropic material is most easily defined by giving the "engineering constants": the three moduli E_1, E_2, E_3; Poisson's ratios ν_{12}, ν_{13}, ν_{23}; and the shear moduli G_{12}, G_{13}, and G_{23} associated with the material's principal directions. These moduli define the elastic compliance according to

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{21}/E_2 & -\nu_{31}/E_3 & 0 & 0 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & -\nu_{32}/E_3 & 0 & 0 & 0 \\ -\nu_{13}/E_1 & -\nu_{23}/E_2 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{23} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

The quantity ν_{ij} has the physical interpretation of the Poisson's ratio that characterizes the transverse strain in the *j*-direction, when the material is stressed in the *i*-direction. In general, ν_{ij} is not equal to ν_{ji}: they are related by $\nu_{ij}/E_i = \nu_{ji}/E_j$. The engineering constants can also be given as functions of temperature and other predefined fields, if necessary.

In Abaqus/Standard spatially varying orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for the elastic moduli and Poisson's ratios. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: **ELASTIC*, TYPE=ENGINEERING CONSTANTS

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Engineering Constants**

Stability

Material stability requires

$$
\begin{aligned}
&E_1, E_2, E_3, G_{12}, G_{13}, G_{23} > 0,\\
&|\nu_{12}| < (E_1/E_2)^{1/2},\\
&|\nu_{13}| < (E_1/E_3)^{1/2},\\
&|\nu_{23}| < (E_2/E_3)^{1/2},\\
&1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13} > 0.
\end{aligned}
$$

When the left-hand side of the inequality approaches zero, the material exhibits incompressible behavior. Using the relations $\nu_{ij}/E_i = \nu_{ji}/E_j$, the second, third, and fourth restrictions in the above set can also be expressed as

$$
\begin{aligned}
&|\nu_{21}| < (E_2/E_1)^{1/2},\\
&|\nu_{31}| < (E_3/E_1)^{1/2},\\
&|\nu_{32}| < (E_3/E_2)^{1/2}.
\end{aligned}
$$

Defining Transversely Isotropic Elasticity

A special subclass of orthotropy is *transverse isotropy*, which is characterized by a plane of isotropy at every point in the material. Abaqus assumes the 2-3 plane to be the plane of isotropy at every point; therefore, transverse isotropy requires that $E_2 = E_3$, $\nu_{12} = \nu_{13}$, $\nu_{21} = \nu_{31}$, $\nu_{23} = \nu_{32}$, and $G_{12} = G_{13}$, and $G_{23} = E_2/2(1+\nu_{23})$. The stress-strain laws reduce to

$$
\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} =
\begin{bmatrix}
1/E_1 & -\nu_{21}/E_2 & -\nu_{21}/E_2 & 0 & 0 & 0 \\
-\nu_{12}/E_1 & 1/E_2 & -\nu_{32}/E_2 & 0 & 0 & 0 \\
-\nu_{12}/E_1 & -\nu_{23}/E_2 & 1/E_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/G_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & 1/G_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & 1/G_{23}
\end{bmatrix}
\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix},
$$

where the total number of independent constants is only five.

Input File Usage: **ELASTIC*, TYPE=TRANSVERSELY ISOTROPIC

Abaqus/CAE Usage: Specifying the elasticity of a transversely isotropic material is not supported in Abaqus/CAE.

Stability

In the transversely isotropic case the stability relations for orthotropic elasticity simplify to

$$E_1, E_2, G_{12}, G_{23} > 0,$$
$$|\nu_{12}| < (E_1/E_2)^{1/2},$$
$$|\nu_{21}| < (E_2/E_1)^{1/2},$$
$$|\nu_{23}| < 1.0,$$
$$1 - {\nu_{23}}^2 - 2\nu_{12}\nu_{21} - 2\nu_{21}\nu_{32}\nu_{12} > 0.$$

Defining Orthotropic Elasticity in Plane Stress

Under plane stress conditions, such as in a shell element, only the values of E_1, E_2, ν_{12}, G_{12}, G_{13}, and G_{23} are required to define an orthotropic material. (In all of the plane stress elements in Abaqus the $(1, 2)$ surface is the surface of plane stress, so that the plane stress condition is $\sigma_{33} = 0$.) The shear moduli G_{13} and G_{23} are included because they might be required for modeling transverse shear deformation in a shell. The Poisson's ratio ν_{21} is implicitly given as $\nu_{21} = (E_2/E_1)\,\nu_{12}$. In this case the stress-strain relations for the in-plane components of the stress and strain are of the form

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix}.$$

In Abaqus/Standard spatially varying plane stress orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for the elastic moduli and Poisson's ratio. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Note: For orthotropic elasticity under plane stress conditions, the solver computes the out of plane strain on the incompressibility assumption.

Input File Usage: **ELASTIC*, TYPE=LAMINA

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Elastic**: **Type**: **Lamina**

Stability

Material stability for plane stress requires

$$E_1, E_2, G_{12}, G_{13}, G_{23} > 0,$$
$$|\nu_{12}| < (E_1/E_2)^{1/2}.$$

Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression

Under plane stress conditions, Abaqus/Explicit allows you to define different orthotropic elastic properties in tension and compression. This type of elasticity definition is called bilamina elasticity. The stress-strain relations for the in-plane components of the stress and strain are of the form

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix},$$

where the values of the elastic constants E_1, E_2, and ν_{12} are interpolated between their tensile values (E_{1+}, E_{2+}, and ν_{12+}) and compressive values (E_{1-}, E_{2-}, and ν_{12-}) based on a criterion depending on the invariants of the strain tensor.

Input File Usage: **ELASTIC*, TYPE=BILAMINA

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Elastic**: **Type**: **Bilamina**

Stability

Material stability for plane stress requires

$$E_{1+}, E_{2+}, G_{12}, E_{1-}, E_{2-} > 0,$$
$$|\nu_{12+}| < (E_{1+}/E_{2+})^{1/2},$$
$$|\nu_{12-}| < (E_{1-}/E_{2-})^{1/2}.$$

Defining Orthotropic Elasticity by Specifying the Terms in the Elastic Stiffness Matrix

Linear elasticity in an orthotropic material can also be defined by giving the nine independent elastic stiffness parameters, as functions of temperature and other predefined fields, if necessary. In this case the stress-strain relations are of the form

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} D_{1111} & D_{1122} & D_{1133} & 0 & 0 & 0 \\ & D_{2222} & D_{2233} & 0 & 0 & 0 \\ & & D_{3333} & 0 & 0 & 0 \\ & & & D_{1212} & 0 & 0 \\ & sym & & & D_{1313} & 0 \\ & & & & & D_{2323} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \left[D^{el}\right] \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix}.$$

For an orthotropic material the engineering constants define the $\mathbf{D}$ matrix as

$$D_{1111} = E_1 \left(1 - \nu_{23}\nu_{32}\right) \Upsilon,$$
$$D_{2222} = E_2 \left(1 - \nu_{13}\nu_{31}\right) \Upsilon,$$
$$D_{3333} = E_3 \left(1 - \nu_{12}\nu_{21}\right) \Upsilon,$$
$$D_{1122} = E_1 \left(\nu_{21} + \nu_{31}\nu_{23}\right) \Upsilon = E_2 \left(\nu_{12} + \nu_{32}\nu_{13}\right) \Upsilon,$$
$$D_{1133} = E_1 \left(\nu_{31} + \nu_{21}\nu_{32}\right) \Upsilon = E_3 \left(\nu_{13} + \nu_{12}\nu_{23}\right) \Upsilon,$$
$$D_{2233} = E_2 \left(\nu_{32} + \nu_{12}\nu_{31}\right) \Upsilon = E_3 \left(\nu_{23} + \nu_{21}\nu_{13}\right) \Upsilon,$$
$$D_{1212} = G_{12},$$
$$D_{1313} = G_{13},$$
$$D_{2323} = G_{23},$$

where

$$\Upsilon = \frac{1}{1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}}.$$

When the material stiffness parameters (the D_{ijkl}) are given directly, Abaqus imposes the constraint $\sigma_{33} = 0$ for the plane stress case to reduce the material's stiffness matrix as required.

In Abaqus/Standard spatially varying orthotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for the elastic moduli. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: **ELASTIC*, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Elastic**: **Type**: **Orthotropic**

Stability

The restrictions on the elastic constants due to material stability are

$$D_{1111}, D_{2222}, D_{3333}, D_{1212}, D_{1313}, D_{2323} > 0,$$
$$|D_{1122}| < (D_{1111} D_{2222})^{1/2},$$
$$|D_{1133}| < (D_{1111} D_{3333})^{1/2},$$
$$|D_{2233}| < (D_{2222} D_{3333})^{1/2},$$
$$det\left(D^{el}\right) > 0.$$

The last relation leads to

$$D_{1111} D_{2222} D_{3333} + 2D_{1122} D_{1133} D_{2233} - D_{2222} {D_{1133}}^2 - D_{1111} {D_{2233}}^2 - D_{3333} {D_{1122}}^2 > 0.$$

These restrictions in terms of the elastic stiffness parameters are equivalent to the restrictions in terms of the "engineering constants." Incompressible behavior results when the left-hand side of the inequality approaches zero.

Defining Fully Anisotropic Elasticity

For fully anisotropic elasticity 21 independent elastic stiffness parameters are needed. The stress-strain relations are as follows:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} D_{1111} & D_{1122} & D_{1133} & D_{1112} & D_{1113} & D_{1123} \\ & D_{2222} & D_{2233} & D_{2212} & D_{2213} & D_{2223} \\ & & D_{3333} & D_{3312} & D_{3313} & D_{3323} \\ & & & D_{1212} & D_{1213} & D_{1223} \\ & sym & & & D_{1313} & D_{1323} \\ & & & & & D_{2323} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \left[D^{el}\right] \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix}.$$

When the material stiffness parameters (the D_{ijkl}) are given directly, Abaqus imposes the constraint $\sigma_{33} = 0$ for the plane stress case to reduce the material's stiffness matrix as required.

In Abaqus/Standard spatially varying anisotropic elastic behavior can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for the elastic moduli. If a distribution is used, no dependencies on temperature and/or field variables for the elastic constants can be defined.

Input File Usage: **ELASTIC*, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Anisotropic**

Stability

The restrictions imposed on the elastic constants by stability requirements are too complex to express in terms of simple equations. However, the requirement that $\mathbf{D}^{el}$ is positive definite requires that all of the eigenvalues of the elasticity matrix $\left[D^{el}\right]$ be positive.

Defining Orthotropic Elasticity for 1-DOF Warping Elements

For two-dimensional meshed models of solid cross-section Timoshenko beam elements modeled with 1-DOF warping elements (see *Meshed Beam Cross-Sections*), Abaqus offers a linear elastic material definition that can have two different shear moduli in the user-specified material directions. In the user-specified directions the stress-strain relations are as follows:

$$\begin{Bmatrix} \sigma \\ \tau_1 \\ \tau_2 \end{Bmatrix} = \begin{bmatrix} E & & \\ & G_1 & \\ & & G_2 \end{bmatrix} \begin{Bmatrix} \varepsilon \\ \gamma_1 \\ \gamma_2 \end{Bmatrix}.$$

A local orientation is used to define the angle α between the global directions and the user-specified material directions. In the cross-section directions the stress-strain relations are as follows:

$$\begin{Bmatrix} \sigma \\ \tau_1 \\ \tau_2 \end{Bmatrix} = \begin{bmatrix} E & 0 & 0 \\ & G_1(\cos\alpha)^2 + G_2(\sin\alpha)^2 & (G_1 - G_2)\cos\alpha\sin\alpha \\ & sym & G_1(\sin\alpha)^2 + G_2(\cos\alpha)^2 \end{bmatrix} \begin{Bmatrix} \varepsilon \\ \gamma_1 \\ \gamma_2 \end{Bmatrix},$$

where σ represents the beam's axial stress and τ_1 and τ_2 represent two shear stresses.

Input File Usage: **ELASTIC*, TYPE=TRACTION

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Traction**

Stability

The stability criterion requires that $E > 0$, $G_1 > 0$, and $G_2 > 0$.

Defining Elasticity in Terms of Tractions and Separations for Cohesive Elements

For cohesive elements used to model bonded interfaces (see *Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*) Abaqus offers an elasticity definition that can be written directly in terms of the nominal tractions and the nominal strains. Both uncoupled and coupled behaviors are supported. For uncoupled behavior each traction component depends only on its conjugate nominal strain, while for coupled behavior the response is more general (as shown below). In the local element directions the stress-strain relations for uncoupled behavior are as follows:

$$\begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{bmatrix} E_{nn} & & \\ & E_{ss} & \\ & & E_{tt} \end{bmatrix} \begin{Bmatrix} \varepsilon_n \\ \varepsilon_s \\ \varepsilon_t \end{Bmatrix}.$$

The quantities t_n, t_s, and t_t represent the nominal tractions in the normal and the two local shear directions, respectively; while the quantities ε_n, ε_s, and ε_t represent the corresponding nominal strains. For coupled traction separation behavior the stress-strain relations are as follows:

$$\begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{bmatrix} E_{nn} & E_{ns} & E_{nt} \\ E_{ns} & E_{ss} & E_{st} \\ E_{nt} & E_{st} & E_{tt} \end{bmatrix} \begin{Bmatrix} \varepsilon_n \\ \varepsilon_s \\ \varepsilon_t \end{Bmatrix}.$$

Input File Usage: Use the following option to define uncoupled elastic behavior for cohesive elements:

**ELASTIC*, TYPE=TRACTION

Use the following option to define coupled elastic behavior for cohesive elements:

**ELASTIC*, TYPE=COUPLED TRACTION

Abaqus/CAE Usage: Use the following option to define uncoupled elastic behavior for cohesive elements:

Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Traction**

Use the following option to define coupled elastic behavior for cohesive elements:

Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Coupled Traction**

Stability

The stability criterion for uncoupled behavior requires that $E_{nn} > 0$, $E_{ss} > 0$, and $E_{tt} > 0$. For coupled behavior the stability criterion requires that:

$$\begin{gathered} E_{nn} > 0\,, \quad E_{ss} > 0\,, \quad E_{tt} > 0; \\ E_{ns} < \sqrt{E_{nn} E_{ss}}; \\ E_{st} < \sqrt{E_{ss} E_{tt}}; \\ E_{nt} < \sqrt{E_{nn} E_{tt}}; \\ det \begin{bmatrix} E_{nn} & E_{ns} & E_{nt} \\ E_{ns} & E_{ss} & E_{st} \\ E_{nt} & E_{st} & E_{tt} \end{bmatrix} > 0. \end{gathered}$$

Cohesive Offset

A cohesive offset (*Figure 1*) refers to a value of the separation in the normal direction, $\delta_n = \delta_n^{off}$, below which the resistance to additional closure $\left(\delta_n > 0,\ \dot{\delta}_n < 0\right)$ of the cohesive surfaces increases significantly. This capability is useful in preventing the closure of a fracture when changes in external loading conditions might otherwise result in closure. A practical application of this capability occurs during hydraulic fracturing with slurry, where solid proppant particles (the solid part of the slurry) help "prop open" the fracture when external hydraulic pressure loads are removed.

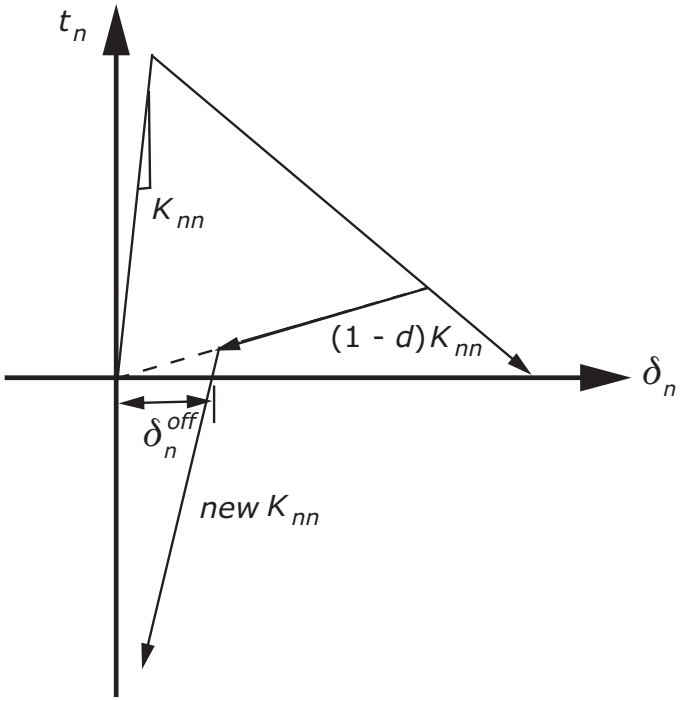


Figure 1: Traction-separation linear elastic response with a cohesive offset.

You can specify an offset to the separation in the normal direction for the following classes of cohesive elements:

- pore pressure;
- coupled temperature-pore pressure;
- coupled slurry concentration-pore pressure; and
- coupled temperature-slurry concentration-pore pressure.

You must specify the offset through user subroutine *UCOHESIVEOFFSET*. In addition to defining an offset, you can also define the stress or the Jacobian matrix for the traction-separation behavior.

If you do not define the stress in user subroutine *UCOHESIVEOFFSET*, Abaqus/Standard computes the normal component of the effective traction as $t_n = E_{nn}\left(\varepsilon_n - \varepsilon_n^{off}\right)$ for an uncoupled traction-separation response and as $t_n = E_{nn}\left(\varepsilon_n - \varepsilon_n^{off}\right) + E_{ns}\varepsilon_s + E_{nt}\varepsilon_t$ for coupled traction behavior. In the expressions above for t_n, $\varepsilon_n^{off} = \frac{\delta_n^{off}}{T_0}$, where T_0 is the original thickness of the cohesive element. The cohesive offset is considered only in the event of crack "closing back" ($\delta_n < \delta_n^{off} \le \delta_n^{\max}$, where $\delta_n^{\max}$ is the historical maximum opening.) . Subsequently, the effective traction is degraded due to progressive damage.

Input File Usage: Use the following option to define the cohesive offset for uncoupled elastic traction-separation behavior:

**ELASTIC*, TYPE=TRACTION, COHESIVE OFFSET=USER

Use the following option to define the cohesive offset for coupled elastic traction-separation behavior:

*ELASTIC, TYPE=COUPLED TRACTION, COHESIVE OFFSET=USER

COHESIVE OFFSET=USER must be used in conjunction with either TYPE=COUPLED TRACTION or TYPE=TRACTION. And also a user subroutine *UCOHESIVEOFFSET* must be associated with this option.

Abaqus/CAE Usage: Specifying the cohesive offset is not supported in Abaqus/CAE.

Defining Isotropic Shear Elasticity for Equations of State in Abaqus/Explicit

Abaqus/Explicit allows you to define isotropic shear elasticity to describe the deviatoric response of materials whose volumetric response is governed by an equation of state (*Elastic Shear Behavior*). In this case the deviatoric stress-strain relationship is given by

$$\mathbf{S} = 2\mu\mathbf{e}^{el},$$

where $\mathbf{S}$ is the deviatoric stress and $\mathbf{e}^{el}$ is the deviatoric elastic strain. You must provide the elastic shear modulus, μ, when you define the elastic deviatoric behavior.

Input File Usage: *ELASTIC, TYPE=SHEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **Type**: **Shear**

Defining the Elastic Transverse Shear Modulus

Abaqus allows you to specify the initial elastic transverse shear moduli for computing the transverse shear stiffness of shells and beams. You can define isotropic, orthotropic, and anisotropic transverse shear moduli.

For beam elements, the material transverse shear stress-relations are as follows:

- Isotropic:

$$\begin{Bmatrix} \sigma_{12} \\ \sigma_{13} \end{Bmatrix} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \gamma_{12} \\ \gamma_{13} \end{Bmatrix}$$

- Orthotropic:

$$\begin{Bmatrix} \sigma_{12} \\ \sigma_{13} \end{Bmatrix} = \begin{bmatrix} G_{12} & 0 \\ 0 & G_{13} \end{bmatrix} \begin{Bmatrix} \gamma_{12} \\ \gamma_{13} \end{Bmatrix}$$

- Anisotropic:

$$\begin{Bmatrix} \sigma_{12} \\ \sigma_{13} \end{Bmatrix} = \begin{bmatrix} D_{1212} & D_{1213} \\ D_{1213} & D_{1313} \end{bmatrix} \begin{Bmatrix} \gamma_{12} \\ \gamma_{13} \end{Bmatrix}$$

For shell elements, the material transverse shear stress-strain relations are as follows:

- Isotropic:

$$\begin{Bmatrix} \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix}$$

- Orthotropic:

$$\begin{Bmatrix} \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} G_{13} & 0 \\ 0 & G_{23} \end{bmatrix} \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix}$$

- Anisotropic:

$$\begin{Bmatrix} \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} D_{1313} & D_{1323} \\ D_{1323} & D_{2323} \end{bmatrix} \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix}$$

You can define the transverse shear moduli for materials for which estimates of the shear moduli are unavailable during the preprocessing stage of input; for example, when the material behavior is defined by user subroutines *UMAT*, *UHYPEL*, *UHYPER*, or *VUMAT*. You can also define the transverse shear moduli to override the default values computed by a material behavior; for example, you can define anisotropic transverse shear moduli for an isotropic elastic material.

The transverse shear moduli that you define are used only to compute the transverse shear stiffness at the beginning of the analysis. Any change to the transverse shear moduli during the analysis is ignored. The transverse shear moduli of a material are ignored if nondefault values of the transverse shear stiffness are specified for a section that is associated with the material (see *Defining the Transverse Shear Stiffness* for shell sections, *Defining the Transverse Shear Stiffness* for general shell sections, and *Transverse Shear Stiffness Definition* for beams).

Input File Usage: Use the following option to specify an isotropic transverse shear modulus:

**TRANSVERSE SHEAR*, TYPE=ISOTROPIC (default)

Use the following option to specify orthotropic transverse shear moduli:

**TRANSVERSE SHEAR*, TYPE=ORTHOTROPIC

Use the following option to specify anisotropic transverse shear moduli:

**TRANSVERSE SHEAR*, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Specifying the elastic transverse shear modulus is not supported in Abaqus/CAE.

Defining the Elasticity of a Short-Fiber Reinforced Composite

You can define the elasticity of a short-fiber reinforced composite (for example, an injection molded composite) by specifying an orientation tensor (see *Specifying a Second-Order Orientation Tensor in Abaqus/Standard* and *Specifying a Second-Order Orientation Tensor in Abaqus/Explicit*) and the elastic modulus of a unidirectional (UD) composite. If the elasticity of the unidirectional composite is transversely isotropic (see *Defining Transversely Isotropic Elasticity*), the elasticity can then be computed using orientation averaging (*Advani and Tucker, 1987*):

$$\left[D^{el}\right] = D_1 a_{ijkl} + D_2 \left(a_{ij}\delta_{kl} + a_{kl}\delta_{ij}\right) + D_3 \left(a_{ik}\delta_{jl} + a_{il}\delta_{jk} + a_{jl}\delta_{ik} + a_{jk}\delta_{il}\right) + D_4 \delta_{ij}\delta_{kl} + D_5 \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right),$$

where D_1, D_2, D_3, D_4, and D_5 are the five constants of the stiffness matrix of the unidirectional composite (the 1-direction is the fiber direction of the unidirectional composite):

$$D_1 = D^{UD}_{1111} + D^{UD}_{2222} - 2D^{UD}_{1122} - 4D^{UD}_{1212},$$

$$D_2 = D^{UD}_{1122} - D^{UD}_{2233},$$

$$D_3 = D^{UD}_{1212} + \tfrac{1}{2}\left(D^{UD}_{2233} - D^{UD}_{2222}\right),$$

$$D_4 = D^{UD}_{2233},$$

$$D_5 = \frac{1}{2}\left(D_{2222}^{UD} - D_{2233}^{UD}\right);$$

a_{ij} is the second-order orientation tensor; and δ_{ij} is the Kronecker delta. The fourth-order tensor a_{ijkl} is computed using the second-order orientation tensor with closure approximations (*Cintra and Tucker, 1995*). Predictions of linear elastic properties are sensitive to the choice of closure approximation. Although there are many closure formulations available in Abaqus, it is recommended to use the orthotropic fitted (ORF) closure (*Cintra and Tucker, 1995*) or orthotropic equaivalent of the natural (ORE) closure (*VerWeyst, 1998*) formulation when predicting mechanical properties.

Abaqus uses the closure approximations that result in an orthotropic stiffness with the principal directions aligned with those of the second-order orientation tensor. Since the elasticity is orthotropic, you must define the material directions with local orientations (see *Orientations*), and the axes of the local system must align with the principal directions of the second-order orientation tensor.

Input File Usage: Use one of the following options to specify the elastic stiffness of the transversely isotropic unidirectional composite:

**ELASTIC*, TYPE=ENGINEERING CONSTANTS
**ELASTIC*, TYPE=ORTHOTROPIC

Use the following option to specify the second-order orientation tensor in the orientation definition in Abaqus/Explicit:

**ORIENTATION*, DISPERSION

Use the following option to specify the second-order orientation tensor with the quadratic closure approximation:

**FIBER DISPERSION*, CLOSURE=QUADRATIC

Use the following option to specify the second-order orientation tensor with the hybrid closure approximation:

**FIBER DISPERSION*, CLOSURE=HYBRID

Use the following option to specify the second-order orientation tensor with the orthotropic fitted (ORF) closure approximation:

**FIBER DISPERSION*, CLOSURE=FITTED

Use the following option to specify the second-order orientation tensor with the orthotropic equaivalent of the natural (ORE) closure approximation:

**FIBER DISPERSION*, CLOSURE=ORE

Use the following option to specify the second-order orientation tensor with the orthotropic smooth (ORS) closure approximation:

**FIBER DISPERSION*, CLOSURE=SMOOTH

Abaqus/CAE Usage: Specifying the elasticity of a short-fiber reinforced composite is not supported in Abaqus/CAE.

Elements

Linear elasticity can be used with any stress/displacement element or coupled temperature-displacement element in Abaqus. The exceptions are traction elasticity, which can be used only with 1-DOF warping elements and cohesive elements; coupled traction elasticity, which can be used only with cohesive elements; shear elasticity, which can be used only with solid (continuum) elements except plane stress elements; and, in Abaqus/Explicit, anisotropic elasticity, which is not supported for truss, rebar, pipe, and beam elements.

If the material is (almost) incompressible (Poisson's ratio $\nu > 0.49$ for isotropic elasticity), hybrid elements should be used in Abaqus/Standard. Compressible anisotropic elasticity should not be used with second-order hybrid continuum elements: inaccurate results and/or convergence problems might occur.

References

- Advani, S., and C. Tucker, "The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites," *Journal of Rheology*, vol. 31 (8), pp. 751–784, 1987.
- Cintra, J., and C. Tucker, "Orthotropic Closure Approximations for Flow-induced Fiber Orientation," *Journal of Rheology*, vol. 39, pp. 1095–1122, 1995.
- VerWeyst, B.E., "Numerical Predictions of Flow-Induced Fiber Orientation in 3-D Geometries," *Ph.D. thesis, University of Illinois at Urbana-Champaign*, 1998.

No Compression or No Tension

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Linear Elastic Behavior*
- **NO COMPRESSION*
- **NO TENSION*
- *Specifying elastic material properties*

Overview



Warning:

Except when used with truss or beam elements, Abaqus/Standard does not form an exact material stiffness for this option. Therefore, the convergence can sometimes be slow.

The no compression and no tension elasticity models:

- are used to modify the linear elasticity of the material so that compressive stress or tensile stress cannot be generated; and
- can be used only in conjunction with an elasticity definition.

Defining the Modified Elastic Behavior

The modified elastic behavior is obtained by first solving for the principal stresses assuming linear elasticity and then setting the appropriate principal stress values to zero. The associated stiffness matrix components will also be set to zero. These models are not history dependent: the directions in which the principal stresses are set to zero are recalculated at every iteration.

The no compression effect for a one-dimensional stress case such as a truss or a layer of a beam in a plane is illustrated in *Figure 1*. No compression and no tension definitions modify only the elastic response of the material.

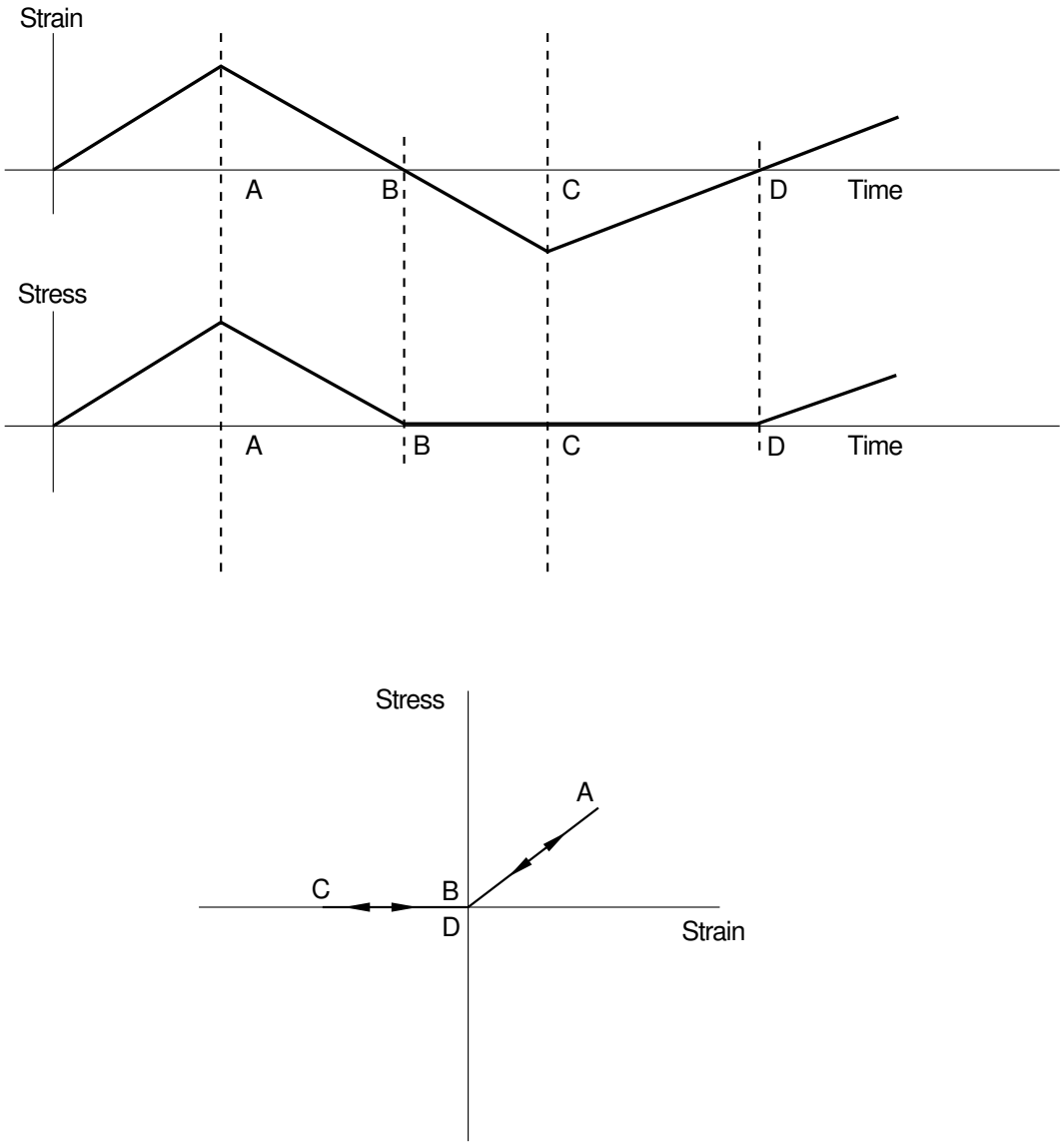


Figure 1: A no compression elastic case with an imposed strain cycle.

Input File Usage: Use one of the following options:

**NO COMPRESSION*
**NO TENSION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**: **No compression** or **No tension**

Cutoff for Principal Stress Values in Abaqus/Explicit

In Abaqus/Explicit, you can define a cutoff for principal stress values to remove the influences of low-level oscillating stresses. If a cutoff stress $\sigma_t^{cutoff} \geq 0$ is defined for a no tension elasticity material, the principal stress values larger than the cutoff stress are set to σ_t^{cutoff}, and principal stress values smaller than or equal to the cutoff

stress are not affected. If a cutoff stress $\sigma_c^{cutoff} \leq 0$ is defined for a no compression elasticity material, the principal stress values smaller than the cutoff stress are set to σ_t^{cutoff}, and principal stress values larger than or equal to the cutoff stress are not affected (an example is shown in *Figure 2*). The default value is based on a small fraction (10^{-4}) of the dilatational wave modulus of the material.

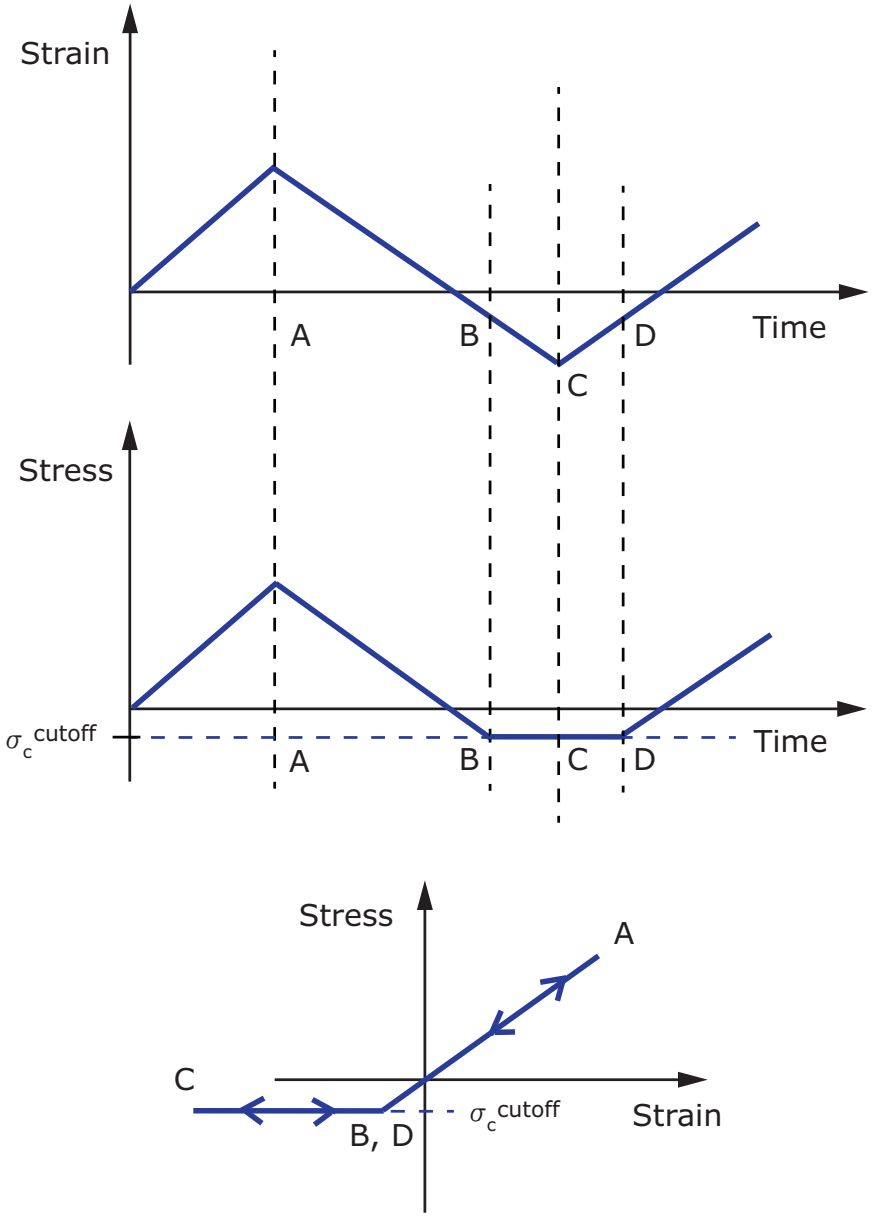


Figure 2: Cutoff for principal stress values.

Input File Usage: Use one of the following options:

**NO COMPRESSION*, CUT OFF STRESS=σ_c^{cutoff}

**NO TENSION*, CUT OFF STRESS=σ_t^{cutoff}

Abaqus/CAE Usage: Defining a cutoff stress is not supported in Abaqus/CAE.

Stability

Using no compression or no tension elasticity can make a model unstable: convergence difficulties might occur. Sometimes these difficulties can be overcome by overlaying each element that uses the no compression (or no tension) model with another element that uses a small value of Young's modulus (small in comparison with the Young's modulus of the element using modified elasticity). This technique creates a small “artificial” stiffness, which can stabilize the model.

Material Options

No compression and no tension definitions can be used only in conjunction with an elasticity definition, which can optionally include thermal expansion and material damping. These definitions cannot be used with any other material option.

Elements

The no compression and no tension elasticity models can be used with any stress/displacement element. However, they cannot be used with shell elements or beam elements if section properties are preintegrated using a general section definition.

Plane Stress Orthotropic Failure Measures

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Linear Elastic Behavior*
- **FAIL STRAIN*
- **FAIL STRESS*
- **ELASTIC*
- *Defining stress-based failure measures for an elastic model*
- *Defining strain-based failure measures for an elastic model*

Overview

The orthotropic plane stress failure measures:

- are indications of material failure (normally used for fiber-reinforced composite materials; for alternative damage and failure models for fiber-reinforced composite materials, see *About Damage and Failure for Fiber-Reinforced Composites*);
- can be used only in conjunction with a linear elastic material model (with or without local material orientations);
- can be used for any element that uses a plane stress formulation; that is, for plane stress continuum elements, shell elements, and membrane elements;
- are postprocessed output requests and do not cause any material degradation; and
- take values that are greater than or equal to 0.0, with values that are greater than or equal to 1.0 implying failure.

Failure Theories

Five different failure theories are provided: four stress-based theories and one strain-based theory.

We denote orthotropic material directions by 1 and 2, with the 1-material direction aligned with the fibers and the 2-material direction transverse to the fibers. For the failure theories to work correctly, the 1- and 2-directions of the user-defined elastic material constants must align with the fiber and the transverse-to-fiber directions, respectively. For applications other than fiber-reinforced composites, the 1- and 2-material directions should represent the strong and weak orthotropic-material directions, respectively.

In all cases tensile values must be positive and compressive values must be negative.

Stress-Based Failure Theories

The input data for the stress-based failure theories are tensile and compressive stress limits, $\boldsymbol{X_t}$ and $\boldsymbol{X_c}$, in the 1-direction; tensile and compressive stress limits, $\boldsymbol{Y_t}$ and $\boldsymbol{Y_c}$, in the 2-direction; and shear strength (maximum shear stress), S, in the X–Y plane.

All four stress-based theories are defined and available with a single definition in Abaqus; the desired output is chosen by the output variables described at the end of this section.

Input File Usage: **FAIL STRESS*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Elastic**: **Suboptions**->**Fail Stress**

Maximum Stress Theory

If $\sigma_{11} > 0$, $X = X_t$; otherwise, $X = X_c$. If $\sigma_{22} > 0$, $Y = Y_t$; otherwise, $Y = Y_c$. The maximum stress failure criterion requires that

$$I_F = \max\left(\frac{\sigma_{11}}{X}, \frac{\sigma_{22}}{Y}, \left|\frac{\sigma_{12}}{S}\right|\right) < 1.0\,.$$

Tsai-Hill Theory

If $\sigma_{11} > 0$, $X = X_t$; otherwise, $X = X_c$. If $\sigma_{22} > 0$, $Y = Y_t$; otherwise, $Y = Y_c$. The Tsai-Hill failure criterion requires that

$$I_F = \frac{\sigma_{11}^2}{X^2} - \frac{\sigma_{11}\sigma_{22}}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0.$$

Tsai-Wu Theory

The Tsai-Wu failure criterion requires that

$$I_F = F_1\sigma_{11} + F_2\sigma_{22} + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{66}\sigma_{12}^2 + 2F_{12}\sigma_{11}\sigma_{22} < 1.0\,.$$

The Tsai-Wu coefficients are defined as follows:

$$F_1 = \frac{1}{X_t} + \frac{1}{X_c}\ , \quad F_2 = \frac{1}{Y_t} + \frac{1}{Y_c}\ , \quad F_{11} = -\frac{1}{X_tX_c}\ , \quad F_{22} = -\frac{1}{Y_tY_c}\ , \quad F_{66} = \frac{1}{S^2}\,.$$

σ_{biax} is the equibiaxial stress at failure. If it is known, then

$$F_{12} = \frac{1}{2\sigma_{biax}^2}\left[1 - \left(\frac{1}{X_t} + \frac{1}{X_c} + \frac{1}{Y_t} + \frac{1}{Y_c}\right)\sigma_{biax} + \left(\frac{1}{X_tX_c} + \frac{1}{Y_tY_c}\right)\sigma_{biax}^2\right];$$

otherwise,

$$F_{12} = \overset{*}{f}\sqrt{F_{11}F_{22}}\,,$$

where $-1.0 \le \overset{*}{f} \le 1.0$. The default value of $\overset{*}{f}$ is zero. For the Tsai-Wu failure criterion either $\overset{*}{f}$ or σ_{biax} must be given as input data. The coefficient $\overset{*}{f}$ is ignored if σ_{biax} is given.

Azzi-Tsai-Hill Theory

The Azzi-Tsai-Hill failure theory is the same as the Tsai-Hill theory, except that the absolute value of the cross product term is taken:

$$I_F = \frac{\sigma_{11}^2}{X^2} - \frac{|\sigma_{11}\sigma_{22}|}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0.$$

This difference between the two failure criteria shows up only when σ_{11} and σ_{22} have opposite signs.

Yamada-Sun Theory

If $\sigma_{11} > 0$, $X = X_t$; otherwise, $X = X_c$. The Yamada-Sun failure criterion requires that:

$$I_F = \frac{\sigma_{11}^2}{X^2} + \frac{\sigma_{12}^2}{S^2} < 1.0.$$

Stress-Based Failure Measures—Failure Envelopes

To illustrate the four stress-based failure measures, *Figure 1*, *Figure 2*, and *Figure 3* show each failure envelope (i.e., $I_F = 1.0$) in (σ_{11}–σ_{22}) stress space compared to the Tsai-Hill envelope for a given value of in-plane shear stress. In each case the Tsai-Hill surface is the piecewise continuous elliptical surface with each quadrant of the surface defined by an ellipse centered at the origin. The parallelogram in *Figure 1* defines the maximum stress surface. In *Figure 2* the Tsai-Wu surface appears as the ellipse. In *Figure 3* the Azzi-Tsai-Hill surface differs from the Tsai-Hill surface only in the second and fourth quadrants, where it is the outside bounding surface (i.e., further from the origin). Since all of the failure theories are calibrated by tensile and compressive failure under uniaxial stress, they all give the same values on the stress axes.

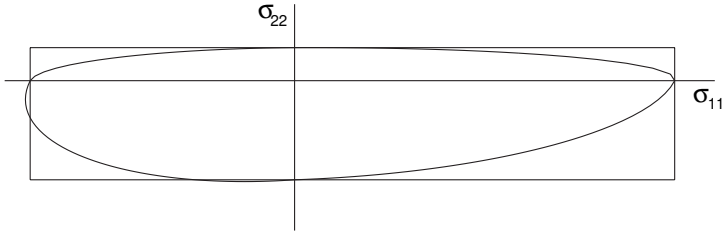


Figure 1: Tsai-Hill versus maximum stress failure envelope ($I_F = 1.0$).

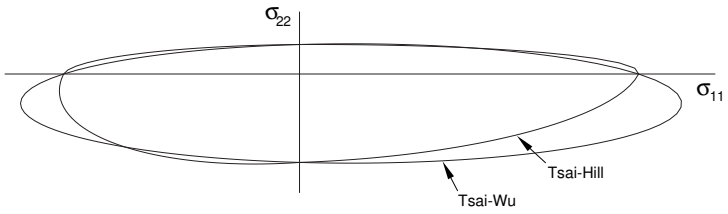


Figure 2: Tsai-Hill versus Tsai-Wu failure envelope ($I_F = 1.0$, $F_{12} = 0.0$).

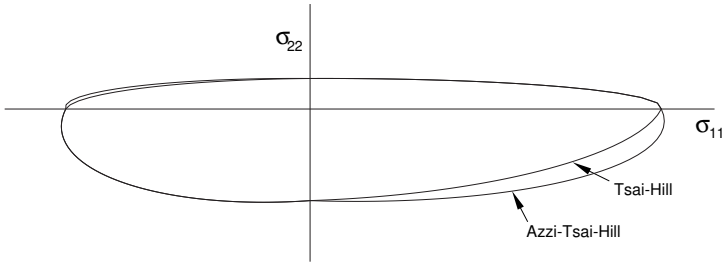


Figure 3: Tsai-Hill versus Azzi-Tsai-Hill failure envelope ($I_F = 1.0$).

Strain-Based Failure Theory

The input data for the strain-based theory are tensile and compressive strain limits, X_{ε_t} and X_{ε_c}, in the 1-direction; tensile and compressive strain limits, Y_{ε_t} and Y_{ε_c}, in the 2-direction; and shear strain limit, S_ε, in the *X*–*Y* plane.

Input File Usage: **FAIL STRAIN*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Elastic**: **Suboptions**->**Fail Strain**

Maximum Strain Theory

If $\varepsilon_{11} > 0$, $X_\varepsilon = X_{\varepsilon_t}$; otherwise, $X_\varepsilon = X_{\varepsilon_c}$. If $\varepsilon_{22} > 0$, $Y_\varepsilon = Y_{\varepsilon_t}$; otherwise, $Y_\varepsilon = Y_{\varepsilon_c}$. The maximum strain failure criterion requires that

$$I_F = \max\left(\frac{\varepsilon_{11}}{X_\varepsilon}, \frac{\varepsilon_{22}}{Y_\varepsilon}, \left|\frac{\varepsilon_{12}}{S_\varepsilon}\right|\right) < 1.0\,.$$

Elements

The plane stress orthotropic failure measures can be used with any plane stress, shell, or membrane element in Abaqus.

Output

Abaqus provides output of the failure index, *R*, if failure measures are defined with the material description. The definition of the failure index and the different output variables are described below.

Output Failure Indices

Each of the stress-based failure theories defines a failure surface surrounding the origin in the three-dimensional space $\{\sigma_{11}, \sigma_{22}, \sigma_{12}\}$. Failure occurs any time a state of stress is either on or outside this surface. The failure index, *R*, is used to measure the proximity to the failure surface. *R* is defined as the scaling factor such that, for the given stress state $\{\sigma_{11}, \sigma_{22}, \sigma_{12}\}$,

$$\left\{\frac{\sigma_{11}}{R}, \frac{\sigma_{22}}{R}, \frac{\sigma_{12}}{R}\right\} \quad \Rightarrow \quad I_F = 1.0\,;$$

that is, $1/R$ is the scaling factor with which we need to multiply all of the stress components simultaneously to lie on the failure surface. Values $R < 1.0$ indicate that the state of stress is within the failure surface, while values $R \geq 1.0$ indicate failure. For the maximum stress theory $R \equiv I_F$.

The failure index *R* is defined similarly for the maximum strain failure theory. *R* is the scaling factor such that, for the given strain state $\{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{12}\}$,

$$\left\{\frac{\varepsilon_{11}}{R}, \frac{\varepsilon_{22}}{R}, \frac{\varepsilon_{12}}{R}\right\} \quad \Rightarrow \quad I_F = 1.0\,.$$

For the maximum strain theory $R \equiv I_F$.

Output Variables

Output variable CFAILURE will provide output for all of the stress- and strain-based failure theories (see *Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), with the exception of the Yamada-Sun failure theory. In Abaqus/Standard history output can also be requested for the individual stress theories with output variables MSTRS, TSAIH, TSAIW, YAMADASUN, and AZZIT and for the strain theory with output variable MSTRN.

Output variables for the stress- and strain-based failure theories are always calculated at the material points of the element. In Abaqus/Standard element output can be requested at a location other than the material points (see *Output to the Data and Results Files*); in this case the output variables are first calculated at the material points, then interpolated to the element centroid or extrapolated to the nodes.

Plasticity Corrections

Products: Abaqus/Standard

References:

- **PLASTICITY CORRECTION*

Overview

Plasticity corrections:

- provide an estimate of the plastic solution for a model analyzed with purely elastic material response;
- can be applied to an isotropic linear elastic material model in general and static linear perturbation procedures;
- can be used with elastic-plastic materials with an isotropic von Mises yield surface (however, the correction is evaluated only in static perturbation procedures, where the material response is elastic);
- are meaningful only when plastic deformation is localized in small regions of the structure;
- are evaluated using the Neuber or Glinka rules;
- are based on either a tabular stress-strain curve or the Ramberg-Osgood definition of the plastic response of the material; and
- have no effect on the solution (additional output is provided only through postprocessing of a linear elastic solution).

Neuber and Glinka Plasticity Corrections

Many types of engineering applications require the solution of nonlinear elastic-plastic problems. The finite element analysis method provides a robust way of obtaining such results accurately. However, performing a nonlinear, elastic-plastic analysis can be computationally expensive.

The Neuber and Glinka plasticity correction rules available in Abaqus/Standard provide an effective method to approximate the elastic-plastic stress and strain solution by performing only a linear elastic analysis, which can significantly reduce the computational cost. This is particularly important in concept design optimization workflows, where the analysis must be performed multiple times. The plasticity correction capabilities are also supported in static linear perturbation steps with multiple load cases, which can further substantially decrease the computational cost of the analysis.

The plasticity corrections provide postprocessed output results based on the evaluation of the Neuber or Glinka rules applied to a linear elastic response, but they do not affect the linear solution. To trigger the evaluation of plasticity corrections, you must request one of the output variables described in *Output*.

Specifying the Plastic Response

Although the output variables associated with the Neuber and Glinka corrections are available only when the material response is purely elastic, their evaluation still requires knowledge of the plastic properties of the material. The plastic properties are used only to evaluate the additional plasticity corrections output variables; other solution results, which are based on linear elasticity, are not affected. You can define the plastic response for the evaluation of plasticity corrections by specifying:

- the coefficients of the Ramberg-Osgood model,
- the tabular definition of the hardening curve, or
- an elastic-plastic material definition with isotropic von Mises plasticity.

In the latter case, the plasticity corrections are evaluated only in static perturbation procedures, where the material response is always assumed to be purely elastic.

Ramberg-Osgood Definition

The Ramberg-Osgood model is based on the observation that the stress versus plastic strain response is linear when plotted in the logarithmic scale (that is, it has a power law relation). In this model, the total strain is decomposed into the elastic and plastic components:

$$\varepsilon = \frac{\sigma}{E} + \left(\frac{\sigma}{K}\right)^{\frac{1}{n}},$$

where ε is the equivalent strain, σ is the equivalent stress, E is the Young's modulus, and K and n are material parameters.

Input File Usage: Use the following option to define the coefficients K and n in the Ramberg-Osgood relationship:

**PLASTICITY CORRECTION*, DEFINITION=RAMBERG-OSGOOD

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity Correction**: **Definition**: **Ramberg-Osgood**

Tabular Definition

You can specify the hardening curve directly by providing the yield stress as a function of the equivalent plastic strain in tabular form.

Input File Usage: Use the following option to define the plastic response in tabular form:

**PLASTICITY CORRECTION*, DEFINITION=TABULAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity Correction**: **Definition**: **Tabular**

Elastic-Plastic Material Definition

The plasticity corrections can be evaluated using the plastic response specified in an elastic-plastic material definition with an isotropic von Mises yield surface. The definition of the hardening curve is provided as a table of yield stresses versus equivalent plastic strains, similar to the tabular definition discussed above. However, in this case, output variables for the plasticity corrections are evaluated only in static linear perturbation procedures, where the material response is always assumed to be elastic. They are not available in general procedures, which always compute a fully nonlinear elastic-plastic solution.

Neuber's Rule

Neuber's rule is one of the most widely used methods for estimating the elastic-plastic stress and strain response from purely elastic stress results. It assumes that the stress-strain product of the elastic solution is equal to the stress-strain product of the elastic-plastic solution. It can be expressed as

$$\sigma^e \varepsilon^e = \sigma^N \varepsilon^N.$$

This equation is depicted graphically in *Figure 1* and means that the areas of the two triangles shown must be equal. The dashed line is called the Neuber hyperbola, and the solution to the problem lies on the intersection

of this line with the elastic-plastic curve. To obtain Neuber's stress and strain (σ^N and ε^N), Abaqus solves the above equation together with the relationship describing the plastic response.

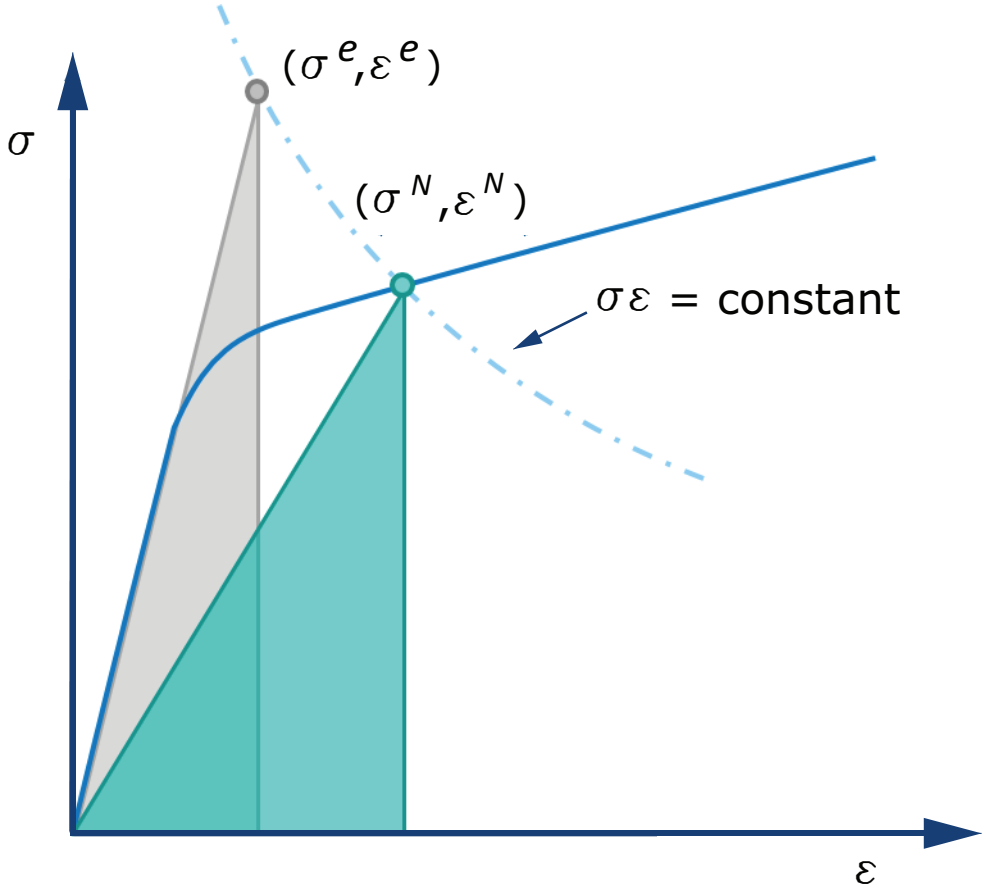


Figure 1: Graphical depiction of Neuber's rule.

Glinka's Rule

Glinka’s rule, also known as the equivalent strain energy density method (ESED), is based on the assumption that the strain energy density distribution in the localized plastic region near a notch is the same as that predicted from a linear elastic solution. This approach generally leads to smaller values of stress and strain compared to Neuber’s rule.

The corrected stress-strain response corresponds to a point on the elastic-plastic curve such that the area under the curve is equal to the area of the triangle under the purely elastic response, as shown in *Figure 2*. The figure also shows the triangle representing the Neuber response for comparison. Glinka's rule can be expressed as

$$\frac{1}{2}\sigma^e\varepsilon^e = \int_0^{\varepsilon^G} \sigma^G \qquad \varepsilon^G.$$

As in Neuber's method, to obtain Glinka's stress and strain (σ^G and ε^G) Abaqus solves the above equation together with the relationship describing the plastic response.

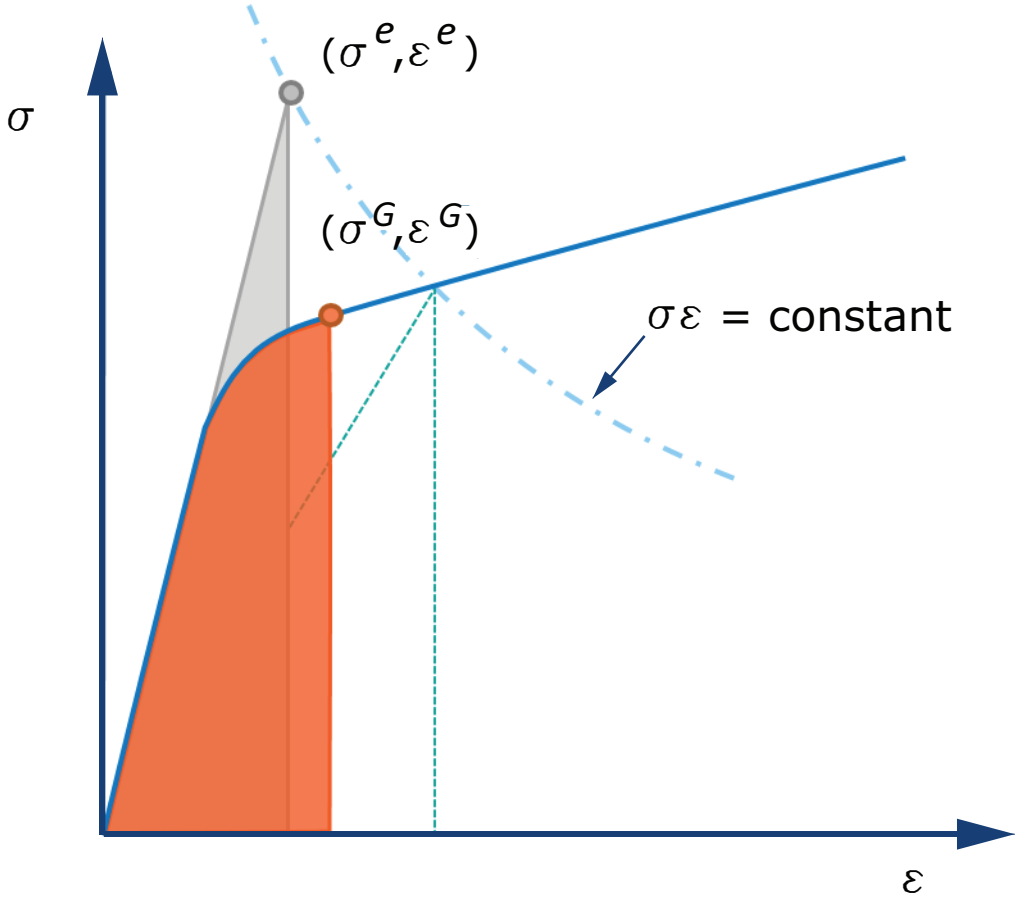


Figure 2: Graphical depiction of Glinka's rule.

Plasticity Corrections in Static Perturbation Procedures

When the static perturbation step follows a general step, Abaqus/Standard calculates the elastic stress (σ^e) that is used to evaluate the plasticity corrections as the sum of the base stress and the perturbation stress. In addition, if you request plasticity corrections for elements that have an elastic-plastic material definition, Abaqus/Standard evaluates the corrections taking into account the fully nonlinear elastic-plastic state of the material at the end of the general step. In this case, Abaqus/Standard uses the modified Neuber and Glinka rules, graphically depicted

in *Figure 3* and *Figure 4*, to compute the corrected stresses and strains. Abaqus/Standard evaluates the yield stress, taking into account the value of equivalent plastic strain in the base state and shifting the strain to account for the change of the stress-free configuration.

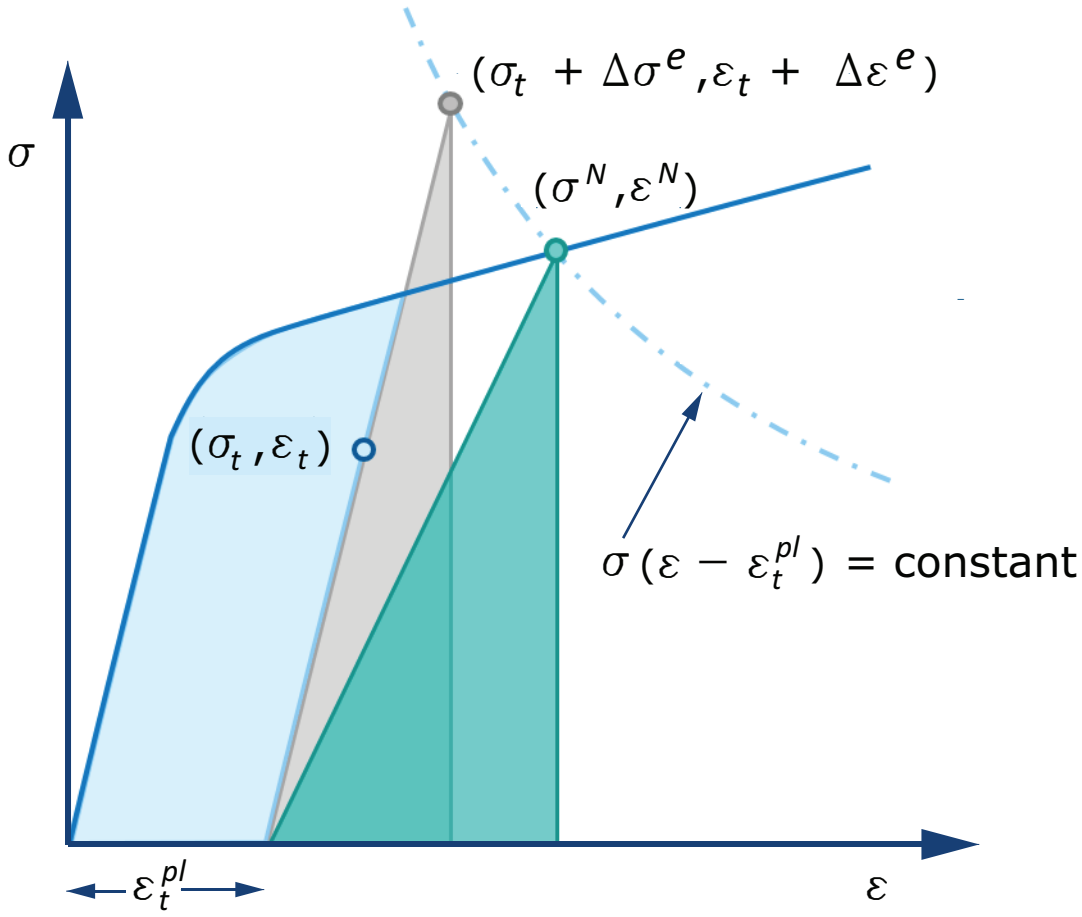


Figure 3: Graphical depiction of the modified Neuber's rule.

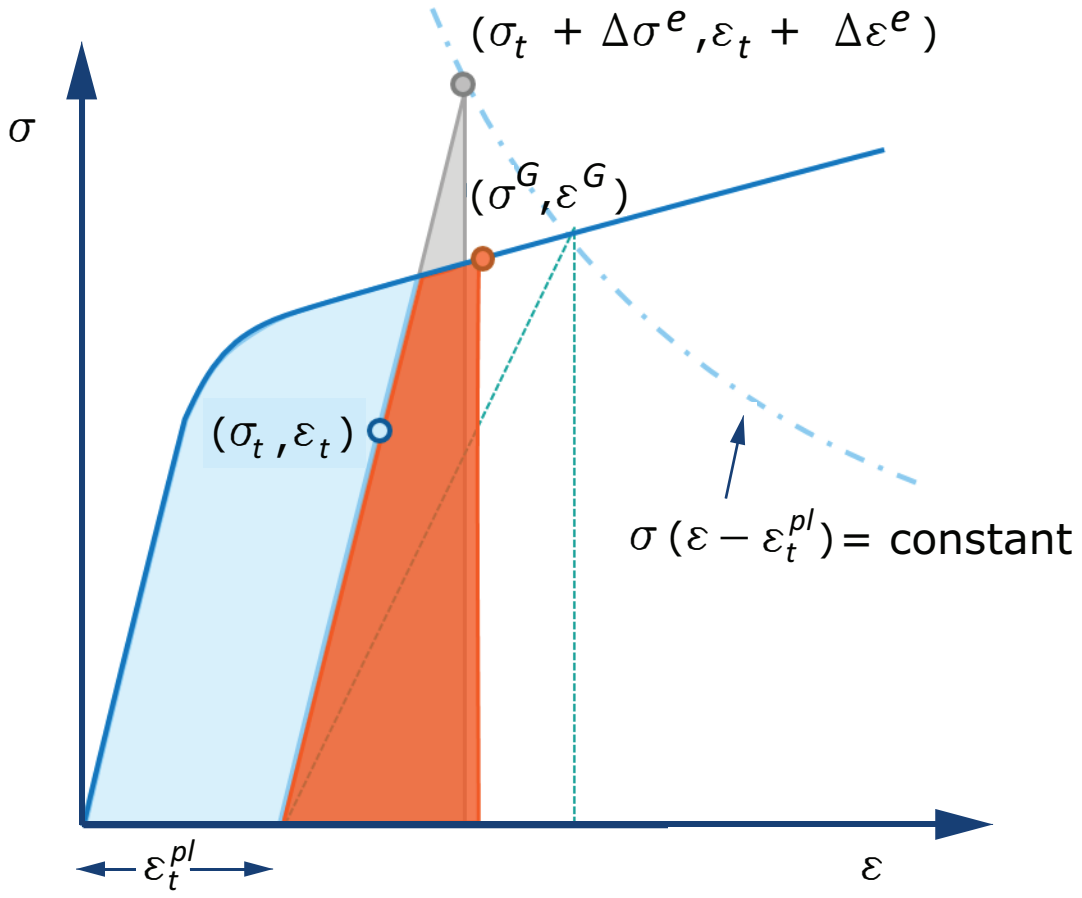


Figure 4: Graphical depiction of the modified Glinka's rule.

Material Options

The plasticity correction capabilities are available only with isotropic linearly elastic materials and elastic-plastic materials with a von Mises yield surface. In the latter case, Abaqus/Standard computes plasticity corrections only in the static linear perturbation procedure.

Elements

The plasticity correction capabilities are available with any elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

The following output variables can be requested to evaluate plasticity corrections:

GKEEQ

Glinka equivalent strain, ε^G.

GKPEEQ

Glinka equivalent plastic strain, $\varepsilon_{pl}^G = \varepsilon^G - \frac{\sigma^G}{E}$.

GKSEQ

Glinka equivalent stress, σ^G.

NBEEQ

Neuber equivalent strain, ε^N.

NBPEEQ

Neuber equivalent plastic strain, $\varepsilon_{pl}^N = \varepsilon^N - \frac{\sigma^N}{E}$.

NBSEQ

Neuber equivalent stress, σ^N.

References

- Molski, K., and G. Glinka, "A Method of Elastic-Plastic Stress and Strain Calculation at a Notch Root," *Materials Science and Engineering*, vol. 50, pp. 93–100, 1981.
- Neuber, H., "Theory of Stress Concentration for Shear-Strained Prismatical Bodies with Arbitrary Nonlinear Stress-Strain Law," *Journal of Applied Mechanics*, vol. 28, pp. 544–550, 1961.

Elastic Behavior of Porous Materials

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- **POROUS ELASTIC*
- **INITIAL CONDITIONS*
- *Creating a porous elastic material model*

Overview

The porous elastic models:

- are used to study the pressure-dependent elastic behavior of materials;
- include a logarithmic model and a power law–based model; and
- can have properties that depend on temperature and other field variables.

Logarithmic Porous Elasticity Model

The logarithmic porous elasticity model is valid for small elastic strains (normally less than 5%). It is a nonlinear, isotropic elasticity model in which the pressure stress varies as an exponential function of volumetric strain. The model allows a zero or nonzero elastic tensile stress limit.

Defining the Volumetric Behavior

Often, the elastic part of the volumetric behavior of porous materials is modeled accurately by assuming that the elastic part of the change in volume of the material is proportional to the logarithm of the pressure stress (*Figure 1*):

$$\frac{\kappa}{(1+e_0)} \ln\left(\frac{p_0+p_t^{el}}{p+p_t^{el}}\right) = J^{el} - 1,$$

where κ is the "logarithmic bulk modulus"; e_0 is the initial void ratio; p is the equivalent pressure stress, defined by

$$p = -\frac{1}{3}\,\mathrm{trace}\,\boldsymbol{\sigma} = -\frac{1}{3}\left(\sigma_{11}+\sigma_{22}+\sigma_{33}\right);$$

p_0 is the initial value of the equivalent pressure stress; J^{el} is the elastic part of the volume ratio between the current and reference configurations; and p_t^{el} is the "elastic tensile strength" of the material (in the sense that $J^{el} \to \infty$ as $p \to -p_t^{el}$).

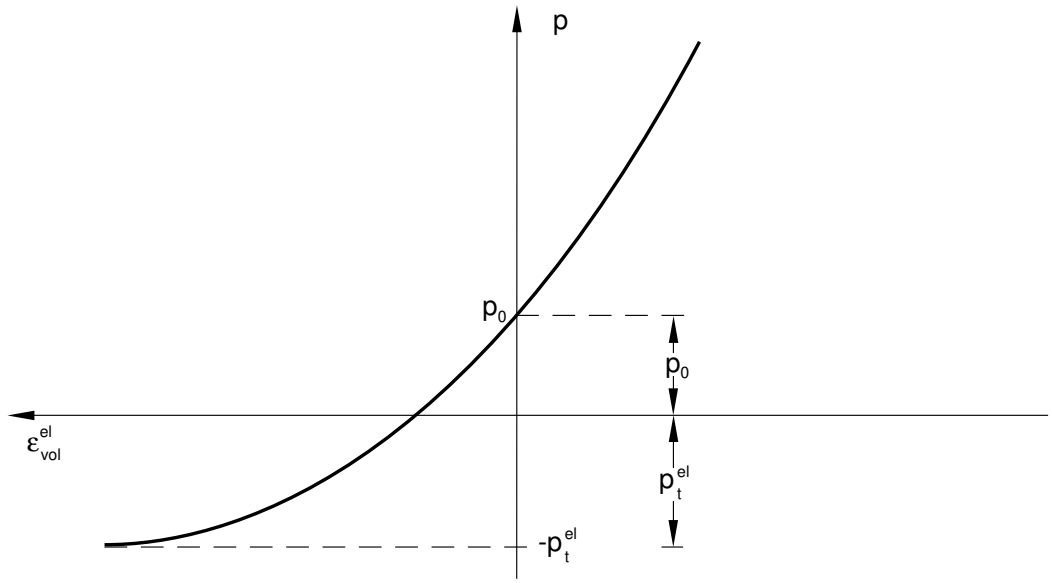


Figure 1: Porous elastic volumetric behavior.

Input File Usage: Use all three of the following options to define a porous elastic material:

**POROUS ELASTIC*, TYPE=LOGARITHMIC,SHEAR=G or POISSON to define κ and p_t^{el}
**INITIAL CONDITIONS*, TYPE=STRESS to define p_0
**INITIAL CONDITIONS*, TYPE=RATIO to define e_0

Abaqus/CAE Usage: Use all three of the following options to define a porous elastic material:

Property module: material editor: **Mechanical**->**Elasticity**->**Porous Elastic**: **Type**: **Logarithmic**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Defining the Shear Behavior

The deviatoric elastic behavior of a porous material can be defined in either of two ways.

By Defining the Shear Modulus

Give the shear modulus, G. The deviatoric stress, $\mathbf{S}$, is then related to the deviatoric part of the total elastic strain, $\mathbf{e}^{el}$, by

$$\mathbf{S} = 2G\,\mathbf{e}^{el}.$$

In this case the shear behavior is not affected by compaction of the material.

Input File Usage: **POROUS ELASTIC*, TYPE=LOGARITHMIC,SHEAR=G

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Porous Elastic**: **Type**: **Logarithmic**: **Shear**: **G**

By Defining Poisson's Ratio

Define Poisson's ratio, ν. The instantaneous shear modulus is then defined from the instantaneous bulk modulus and Poisson's ratio as

$$G = \frac{3(1-2\nu)(1+e_0)}{2(1+\nu)\kappa}\left(p + p_t^{el}\right)\exp\left(\varepsilon_{vol}^{el}\right),$$

where $\varepsilon_{vol}^{el} = \ln J^{el}$ is the logarithmic measure of the elastic volume change. In this case

$$d\mathbf{S} = 2G\, d\mathbf{e}^{el}.$$

Thus, the elastic shear stiffness increases as the material is compacted. This equation is integrated to give the total stress–total elastic strain relationship.

Input File Usage: **POROUS ELASTIC*, TYPE=LOGARITHMIC, SHEAR=POISSON

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Porous Elastic**: **Type**: **Logarithmic**: **Shear**: **Poisson**

Power Law–Based Porous Elasticity Model

In the case of the power law–based porous elasticity model, Young's modulus is related to the pressure by a power law:

$$E = E_{ref}\left[\frac{p+p_0}{p_{ref}+p_0}\right]^n \quad p > 0; E = fE_{ref} \quad p \le 0,$$

where E_{ref} is the reference Young's modulus at pressure p_{ref}, and p_0 and n are material constants. f is used to define the elastic modulus for tensile effective pressure and is calculated as follows:

$$f = \left[\frac{p_0}{p_{ref}+p_0}\right]^n.$$

Poisson's ratio is an exponential function of pressure

$$\nu = \nu_0 + (\nu_\infty - \nu_0)\left(1 - e^{-mp}\right) \quad p > 0; \nu = \nu_0 \quad p \le 0,$$

where ν_∞ is the value of the Poisson's ratio corresponding to the limit $p \to \infty$, ν_0 is the value of the Poisson's ratio at $p = 0$, and m is a material constant.

Material Options

The porous elasticity models can be used by themselves or in combination with other material options.

The logarithmic porous elasticity model can be combined with:

- the *Extended Drucker-Prager Models*;
- the *Modified Drucker-Prager/Cap Model*;
- the *Critical State (Clay) Plasticity Model*; or
- isotropic expansion to introduce thermal volume changes (*Thermal Expansion*).

The power law–based porous elasticity model can be combined with:

- the soft rock plasticity model; or
- isotropic expansion to introduce thermal volume changes (*Thermal Expansion*).

It is not possible to use porous elasticity with rate-dependent plasticity or viscoelasticity.

Porous elasticity cannot be used with the porous metal plasticity model (*Porous Metal Plasticity*).

See *Combining Material Behaviors* for more details.

Elements

Porous elasticity cannot be used with hybrid elements or plane stress elements (including shells and membranes), but it can be used with any other pure stress/displacement element in Abaqus/Standard.

If used with reduced-integration elements with total-stiffness hourglass control, Abaqus/Standard cannot calculate a default value for the hourglass stiffness of the element if the logarithmic porous elasticity model is used and the shear behavior is defined through Poisson's ratio. Hence, you must specify the hourglass stiffness. See *Section Controls* for details.

If fluid pore pressure is important (such as in undrained soils), stress/displacement elements that include pore pressure can be used.

Hypoelastic Behavior

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- **HYPOELASTIC*
- *Creating a hypoelastic material model*

Overview

The hypoelastic material model:

- is valid for small elastic strains—the stresses should not be large compared to the elastic modulus of the material;
- is used when the load path is monotonic; and
- must be defined by user subroutine *UHYPEL* if temperature dependence is to be included.

Defining Hypoelastic Material Behavior

In a hypoelastic material the rate of change of stress is defined as a tangent modulus matrix multiplying the rate of change of the elastic strain:

$$d\boldsymbol{\sigma} = \mathbf{D}^{el} : d\boldsymbol{\varepsilon}^{el},$$

where $d\boldsymbol{\sigma}$ is the rate of change of the stress (the "true," Cauchy, stress in finite-strain problems), $\mathbf{D}^{el}$ is the tangent elasticity matrix, and $d\boldsymbol{\varepsilon}^{el}$ is the rate of change of the elastic strain (the log strain in finite-strain problems).

Determining the Hypoelastic Material Parameters

The entries in $\mathbf{D}^{el}$ are provided by giving Young's modulus, E, and Poisson's ratio, ν, as functions of strain invariants. The strain invariants are defined for this purpose as

$$\begin{aligned} I_1 &= \text{trace } \boldsymbol{\varepsilon}^{el}, \\ I_2 &= \frac{1}{2}\left(\boldsymbol{\varepsilon}^{el} : \boldsymbol{\varepsilon}^{el} - I_1^2\right), \\ I_3 &= \det\left(\boldsymbol{\varepsilon}^{el}\right). \end{aligned}$$

You can define the material parameters directly or by using a user subroutine.

Direct Specification

You can define the variation of Young's modulus and Poisson's ratio directly by specifying E, ν, I_1, I_2, and I_3.

Input File Usage: **HYPOELASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hypoelastic**

User Subroutine

If specifying E and ν as functions of the strain invariants directly does not allow sufficient flexibility, you can define the hypoelastic material by user subroutine *UHYPEL*.

Input File Usage: **HYPOELASTIC*, USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hypoelastic**:
Use user subroutine UHYPEL

Plane or Uniaxial Stress

For plane stress and uniaxial stress states Abaqus/Standard does not compute the out-of-plane strain components.

For the purpose of defining the above invariants, it is assumed that $I_1 = 0$; that is, the material is assumed to be incompressible. For example, in a uniaxial stress case (such as a truss element) this assumption implies that

$$I_1 = 0,$$
$$I_2 = \frac{3}{4}\left(\varepsilon_{11}^{el}\right)^2,$$
$$I_3 = \frac{1}{4}\left(\varepsilon_{11}^{el}\right)^3.$$

Large-Displacement Analysis

For large-displacement analysis the strain measure in Abaqus is the integration of the rate of deformation. This strain measure corresponds to log strain if the principal directions do not rotate relative to the material. The strain invariant definitions should be interpreted in this way.

Material Options

The hypoelastic material model can be used only by itself in the material definition. It cannot be combined with viscoelasticity or with any inelastic response model. See *Combining Material Behaviors* for more details.

Elements

The hypoelastic material model can be used with any of the stress/displacement elements in Abaqus/Standard.

Hyperelasticity

The hyperelastic model for rubberlike materials provides a general capability for modeling the behavior of nearly incompressible elastomers under large elastic deformations.

In this section:

- *Hyperelastic Behavior of Rubberlike Materials*
- *Hyperelastic Behavior in Elastomeric Foams*
- *Anisotropic Hyperelastic Behavior*

Hyperelastic Behavior of Rubberlike Materials

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Stress Softening in Rubberlike Materials*
- *Permanent Set in Rubberlike Materials*
- **HYPERELASTIC*
- **UNIAXIAL TEST DATA*
- **BIAXIAL TEST DATA*
- **PLANAR TEST DATA*
- **VOLUMETRIC TEST DATA*
- **MULLINS EFFECT*
- *Creating an isotropic hyperelastic material model*

Overview

The hyperelastic material model:

- is isotropic and nonlinear;
- is valid for materials that exhibit instantaneous elastic response up to large strains (such as rubber, solid propellant, or other elastomeric materials); and
- requires that geometric nonlinearity be accounted for during the analysis step (*General and Perturbation Procedures*), since it is intended for finite-strain applications.

Compressibility

Most elastomers (solid, rubberlike materials) have very little compressibility compared to their shear flexibility. This behavior does not warrant special attention for plane stress, shell, membrane, beam, truss, or rebar elements, but the numerical solution can be quite sensitive to the degree of compressibility for three-dimensional solid, plane strain, and axisymmetric analysis elements. In cases where the material is highly confined (such as an O-ring used as a seal), the compressibility must be modeled correctly to obtain accurate results. In applications where the material is not highly confined, the degree of compressibility is typically not crucial; for example, it would be quite satisfactory in Abaqus/Standard to assume that the material is fully incompressible: the volume of the material cannot change except for thermal expansion.

Another class of rubberlike materials is elastomeric foam, which is elastic but very compressible. Elastomeric foams are discussed in *Hyperelastic Behavior in Elastomeric Foams*.

We can assess the relative compressibility of a material by the ratio of its initial bulk modulus, K_0, to its initial shear modulus, μ_0. This ratio can also be expressed in terms of Poisson's ratio, ν, since

$$\nu = \frac{3K_0/\mu_0 - 2}{6K_0/\mu_0 + 2}.$$

The table below provides some representative values.

K_0/μ_0	Poisson's ratio
10	0.452
20	0.475
50	0.490
100	0.495
1000	0.4995
10,000	0.49995

Compressibility in Abaqus/Standard

In Abaqus/Standard it is recommended that you use solid continuum hybrid elements for almost incompressible hyperelastic materials with an initial Poisson's ratio greater than 0.495 (that is, the ratio of K_0/μ_0 is greater than 100) to avoid potential convergence problems. Otherwise, the analysis preprocessor issues an error. Except for fully incompressible hyperelastic materials, you can use the "nonhybrid incompressible" diagnostics control to downgrade this error to a warning message.

In plane stress, shell, and membrane elements the material is free to deform in the thickness direction. Similarly, in one-dimensional elements (such as beams, trusses, and rebars) the material is free to deform in the lateral directions. In these cases, special treatment of the volumetric behavior is not necessary; the use of regular stress/displacement elements is satisfactory.

Input File Usage: Use the following option to downgrade an error message to a warning message:

**DIAGNOSTICS*, NONHYBRID INCOMPRESSIBLE=WARNING

Compressibility in Abaqus/Explicit

Except for plane stress and uniaxial cases, it is not possible to assume that the material is fully incompressible in Abaqus/Explicit because the program has no mechanism for imposing such a constraint at each material calculation point. Instead, we must provide some compressibility. The difficulty is that, in many cases, the actual material behavior provides too little compressibility for the algorithms to work efficiently. Thus, except for plane stress and uniaxial cases, you must provide enough compressibility for the code to work, knowing that this makes the bulk behavior of the model softer than that of the actual material. Some judgment is, therefore, required to decide whether or not the solution is sufficiently accurate, or whether the problem can be modeled at all with Abaqus/Explicit because of this numerical limitation.

If no value is given for the material compressibility in the hyperelastic model, by default Abaqus/Explicit assumes $K_0/\mu_0 = 20$, corresponding to Poisson's ratio of 0.475. Since typical unfilled elastomers have K_0/μ_0 ratios in the range of 1,000 to 10,000 ($\nu = 0.4995$ to $\nu = 0.49995$) and filled elastomers have K_0/μ_0 ratios in the range of 50 to 200 ($\nu = 0.490$ to $\nu = 0.497$), this default provides much more compressibility than is available in most elastomers. However, if the elastomer is relatively unconfined, this softer modeling of the material's bulk behavior usually provides quite accurate results. Unfortunately, in cases where the material is highly confined—such as when it is in contact with stiff, metal parts and has a very small amount of free surface, especially when the loading is highly compressive—it might not be feasible to obtain accurate results with Abaqus/Explicit.

If you are defining the compressibility rather than accepting the default value, an upper limit of 100 is suggested for the ratio of K_0/μ_0. Larger ratios introduce high frequency noise into the dynamic solution and require the use of excessively small time increments.

Isotropy Assumption

In Abaqus all hyperelastic models are based on the assumption of isotropic behavior throughout the deformation history. Hence, the strain energy potential can be formulated as a function of the strain invariants.

Strain Energy Potentials

Hyperelastic materials are described in terms of a "strain energy potential," $U(\varepsilon)$, which defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the strain at that point in the material. The following forms of strain energy potentials available in Abaqus to model approximately incompressible isotropic elastomers:

- Arruda-Boyce
- Hencky
- Marlow
- Mooney-Rivlin
- Neo-Hookean
- Ogden
- Polynomial
- Reduced polynomial
- Yeoh
- Valanis-Landel
- Van der Waals

You can view the reduced polynomial and Mooney-Rivlin models as particular cases of the polynomial model. In turn, you can view the Yeoh and neo-Hookean potentials as special cases of the reduced polynomial model. Therefore, we occasionally refer collectively to these models as "polynomial models."

Generally, when data from multiple experimental tests are available (typically, this requires at least uniaxial and equibiaxial test data), the Ogden and Van der Waals forms are more accurate in fitting experimental results. If limited test data are available for calibration, the Arruda-Boyce, Van der Waals, Yeoh, or reduced polynomial forms provide reasonable behavior. When only uniaxial test data is available, the Marlow or the Valanis-Landel form is recommended. If only equibiaxial or planar test data is available, the Marlow form is recommended. In this case, a strain energy potential is constructed that reproduces the test data exactly and that has reasonable behavior in other deformation modes.

The Hencky model does not support the specification of test data input for its definition. This model provides a finite strain alternative to the isotropic linear elastic definition (*Defining Isotropic Elasticity*).

Evaluating Hyperelastic Materials

Abaqus/CAE allows you to evaluate hyperelastic material behavior by automatically creating response curves using selected strain energy potentials. In addition, you can provide experimental test data for a material without specifying a particular strain energy potential and have Abaqus/CAE evaluate the material to determine the optimal strain energy potential. See *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior* for details. Alternatively, you can use single-element test cases to evaluate the strain energy potential.

Arruda-Boyce Form

The form of the Arruda-Boyce strain energy potential is

$$U = \mu\left\{\frac{1}{2}\left(\bar{I}_1 - 3\right) + \frac{1}{20\lambda_m^2}\left(\bar{I}_1^2 - 9\right) + \frac{11}{1050\lambda_m^4}\left(\bar{I}_1^3 - 27\right) + \frac{19}{7000\lambda_m^6}\left(\bar{I}_1^4 - 81\right) + \frac{519}{673750\lambda_m^8}\left(\bar{I}_1^5 - 243\right)\right\} + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right),$$

where U is the strain energy per unit of reference volume; μ, λ_m, and D are temperature-dependent material parameters; $\bar{I}_1$ is the first deviatoric strain invariant defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2,$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus, μ_0, is related to μ with the expression

$$\mu_0 = \mu\left(1 + \frac{3}{5\lambda_m^2} + \frac{99}{175\lambda_m^4} + \frac{513}{875\lambda_m^6} + \frac{42039}{67375\lambda_m^8}\right).$$

A typical value of λ_m is 7, for which $\mu_0 = 1.0125\mu$. Both the initial shear modulus, μ_0, and the parameter μ are printed in the data (`.dat`) file if you request a printout of the model data from the analysis input file processor. The initial bulk modulus is related to D with the expression

$$K_0 = \frac{2}{D}.$$

Hencky Form

The form of the Hencky strain energy potential is

$$U = \mu\left(\bar{\varepsilon}_1^2 + \bar{\varepsilon}_2^2 + \bar{\varepsilon}_3^2\right) + \frac{1}{2}K\varepsilon_v^2,$$

where U is the strain energy per unit of reference volume; $\mu = E/\left[2\left(1+\nu\right)\right]$ is the shear modulus; $K = E/\left[3\left(1-2\nu\right)\right]$ is the bulk modulus; $\bar{\varepsilon}_i$ are the deviatoric strain components; and ε_v is the volumetric strain. $\bar{\varepsilon}_i$ and ε_v are defined as

$$\bar{\varepsilon}_i = \ln\left(\bar{\lambda}_i\right)$$

and

$$\varepsilon_v = \ln\left(J^{el}\right),$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; J is the total volume ratio; J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*; and λ_i are the principal stretches.

The Hencky model provides a finite strain alternative to the isotropic linear elastic definition for geometrically nonlinear problems.

Marlow Form

The form of the Marlow strain energy potential is

$$U = U_{dev}\left(\bar{I}_1\right) + U_{vol}\left(J^{el}\right),$$

where U is the strain energy per unit of reference volume, with U_{dev} as its deviatoric part and U_{vol} as its volumetric part; $\bar{I}_1$ is the first deviatoric strain invariant defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2,$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J_{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The deviatoric part of the potential is defined by providing either uniaxial, equibiaxial, or planar test data; while the volumetric part is defined by providing the volumetric test data, defining the Poisson's ratio, or specifying the lateral strains together with the uniaxial, equibiaxial, or planar test data.

Mooney-Rivlin Form

The form of the Mooney-Rivlin strain energy potential is

$$U = C_{10}\left(\bar{I}_1 - 3\right) + C_{01}\left(\bar{I}_2 - 3\right) + \frac{1}{D_1}\left(J^{el} - 1\right)^2,$$

where U is the strain energy per unit of reference volume; C_{10}, C_{01}, and D_1 are temperature-dependent material parameters; $\bar{I}_1$ and $\bar{I}_2$ are the first and second deviatoric strain invariants defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2 \qquad \text{and} \qquad \bar{I}_2 = \bar{\lambda}_1^{(-2)} + \bar{\lambda}_2^{(-2)} + \bar{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2\left(C_{10} + C_{01}\right), \qquad K_0 = \frac{2}{D_1}.$$

Neo-Hookean Form

The form of the neo-Hookean strain energy potential is

$$U = C_{10}\left(\bar{I}_1 - 3\right) + \frac{1}{D_1}\left(J^{el} - 1\right)^2,$$

where U is the strain energy per unit of reference volume; C_{10} and D_1 are temperature-dependent material parameters; $\bar{I}_1$ is the first deviatoric strain invariant defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2,$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10}, \qquad K_0 = \frac{2}{D_1}.$$

Ogden Form

The form of the Ogden strain energy potential is

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}\left(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3\right) + \sum_{i=1}^{N} \frac{1}{D_i}\left(J^{el} - 1\right)^{2i},$$

where $\bar{\lambda}_i$ are the deviatoric principal stretches $\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$; λ_i are the principal stretches; N is a material parameter; μ_i, α_i, and D_i are temperature-dependent material parameters; and J^{el} is the elastic volume ratio as

defined below in *Thermal Expansion*. The initial shear modulus and bulk modulus for the Ogden form are given by

$$\mu_0 = \sum_{i=1}^{N} \mu_i \ , \qquad K_0 = \frac{2}{D_1}.$$

The particular material models described above—the Mooney-Rivlin and neo-Hookean forms—can also be obtained from the general Ogden strain energy potential for special choices of μ_i and α_i.

Polynomial Form

The form of the polynomial strain energy potential is

$$U = \sum_{i+j=1}^{N} C_{ij} \left(\bar{I}_1 - 3\right)^i \left(\bar{I}_2 - 3\right)^j + \sum_{i=1}^{N} \frac{1}{D_i} \left(J^{el} - 1\right)^{2i},$$

where U is the strain energy per unit of reference volume; N is a material parameter; C_{ij} and D_i are temperature-dependent material parameters; $\bar{I}_1$ and $\bar{I}_2$ are the first and second deviatoric strain invariants defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2 \qquad \text{and} \qquad \bar{I}_2 = \bar{\lambda}_1^{(-2)} + \bar{\lambda}_2^{(-2)} + \bar{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}} \lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2\left(C_{10} + C_{01}\right), \qquad K_0 = \frac{2}{D_1}.$$

For cases where the nominal strains are small or only moderately large (< 100%), the first terms in the polynomial series usually provide a sufficiently accurate model. Some particular material models—the Mooney-Rivlin, neo-Hookean, and Yeoh forms—are obtained for special choices of C_{ij}.

Reduced Polynomial Form

The form of the reduced polynomial strain energy potential is

$$U = \sum_{i=1}^{N} C_{i0} \left(\bar{I}_1 - 3\right)^i + \sum_{i=1}^{N} \frac{1}{D_i} \left(J^{el} - 1\right)^{2i},$$

where U is the strain energy per unit of reference volume; N is a material parameter; C_{i0} and D_i are temperature-dependent material parameters; $\bar{I}_1$ is the first deviatoric strain invariant defined as

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2,$$

where the deviatoric stretches $\bar{\lambda}_i = J^{-\frac{1}{3}} \lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10} \ , \qquad K_0 = \frac{2}{D_1}.$$

Valanis-Landel Form

The form of the Valanis-Landel strain energy potential is

$$U = U_{dev}\left(\overline{\lambda}_1, \overline{\lambda}_2, \overline{\lambda}_3\right) + U_{vol}\left(J_{el}\right),$$

where U is the strain energy per unit of reference volume, with U_{dev} as its deviatoric part and U_{vol} as its volumetric part. For the Valanis-Landel model it is further assumed that the deviatoric part of the strain energy potential, U_{dev}, is expressed as three separable but identical functions of principal deviatoric stretches:

$$U_{dev} = F\left(\overline{\lambda}_1\right) + F\left(\overline{\lambda}_2\right) + F\left(\overline{\lambda}_3\right),$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The deviatoric part of the potential is defined by providing either tensile and compressive uniaxial test data or tensile uniaxial and biaxial test data; while the volumetric part is defined either by providing volumetric test data, by providing the uniaxial test data with lateral strains specified, or by defining the Poisson's ratio.

Van Der Waals Form

The form of the Van der Waals strain energy potential is

$$U = \mu\left\{-\left(\lambda_m^2 - 3\right)\left[\ln(1-\eta) + \eta\right] - \frac{2}{3}a\left(\frac{\tilde{I}-3}{2}\right)^{\frac{3}{2}}\right\} + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right),$$

where

$$\tilde{I} = (1-\beta)\,\overline{I}_1 + \beta\overline{I}_2 \quad \text{and} \quad \eta = \sqrt{\frac{\tilde{I}-3}{\lambda_m^2 - 3}}.$$

Here, U is the strain energy per unit of reference volume; μ is the initial shear modulus; λ_m is the locking stretch; a is the global interaction parameter; β is an invariant mixture parameter; and D governs the compressibility. These parameters can be temperature-dependent. $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2 \qquad \text{and} \qquad \overline{I}_2 = \overline{\lambda}_1^{(-2)} + \overline{\lambda}_2^{(-2)} + \overline{\lambda}_3^{(-2)},$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = \mu\,, \qquad K_0 = \frac{2}{D}.$$

Yeoh Form

The form of the Yeoh strain energy potential is

$$\begin{aligned} U = \; & C_{10}\left(\overline{I}_1 - 3\right) + C_{20}\left(\overline{I}_1 - 3\right)^2 + C_{30}\left(\overline{I}_1 - 3\right)^3 \\ & + \frac{1}{D_1}\left(J^{el} - 1\right)^2 + \frac{1}{D_2}\left(J^{el} - 1\right)^4 + \frac{1}{D_3}\left(J^{el} - 1\right)^6, \end{aligned}$$

where U is the strain energy per unit of reference volume; C_{i0} and D_i are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant defined as

$$\overline{I}_1 = \overline{\lambda}_1^2 + \overline{\lambda}_2^2 + \overline{\lambda}_3^2,$$

where the deviatoric stretches $\overline{\lambda}_i = J^{-\frac{1}{3}} \lambda_i$, J is the total volume ratio, J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*, and λ_i are the principal stretches. The initial shear modulus and bulk modulus are given by

$$\mu_0 = 2C_{10} , \qquad K_0 = \frac{2}{D_1}.$$

Thermal Expansion

Only isotropic thermal expansion is permitted with the hyperelastic material model.

The elastic volume ratio, J^{el}, relates the total volume ratio, J, and the thermal volume ratio, J^{th}:

$$J^{el} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = \left(1 + \varepsilon^{th}\right)^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient (*Thermal Expansion*).

Defining the Hyperelastic Material Response

The mechanical response of a material is defined by choosing a strain energy potential to fit the particular material. The strain energy potential forms in Abaqus are written as separable functions of a deviatoric component and a volumetric component; that is, $U = U_{dev}\left(\overline{I}_1, \overline{I}_2\right) + U_{vol}\left(J^{el}\right)$ when defined with respect to invariants, or alternatively, $U = U_{dev}\left(\overline{\lambda}_1, \overline{\lambda}_2, \overline{\lambda}_3\right) + U_{vol}\left(J^{el}\right)$ when defined with respect to principal stretches. Alternatively, you can define the strain energy potential with user subroutines *UHYPER* and *UHYPER_STRETCH* (in Abaqus/Standard) or with *VUHYPER* and *VUHYPER_STRETCH* (in Abaqus/Explicit), in which case the strain energy potential need not be separable.

Generally for the hyperelastic material models available in Abaqus, you can either directly specify material coefficients or provide experimental test data and have Abaqus automatically determine appropriate values of the coefficients. An exception is the Marlow form: in this case, the deviatoric part of the strain energy potential must be defined with test data. The different methods for defining the strain energy potential are described in detail below.

The properties of rubberlike materials can vary significantly from one batch to another; therefore, if data are used from several experiments, all the experiments should be performed on specimens taken from the same batch of material, regardless of whether you or Abaqus compute the coefficients.

Viscoelastic and Hysteretic Materials

The elastic response of viscoelastic materials (*Time Domain Viscoelasticity* and *Parallel Rheological Framework*) and hysteretic materials (*Hysteresis in Elastomers*) can be specified by defining either the instantaneous response or the long-term response of such materials.

Instantaneous Response

To define the instantaneous response, the experiments outlined in *Experimental Tests* have to be performed within time spans much shorter than the characteristic relaxation times of these materials.

Input File Usage: **HYPERELASTIC*, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown**: **Moduli time scale (for viscoelasticity): Instantaneous**

Long-Term Response

If the long-term elastic response is used, data from experiments have to be collected after time spans much longer than the characteristic relaxation times of these materials. Long-term elastic response is the default elastic material behavior.

Input File Usage: **HYPERELASTIC*, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown**: **Moduli time scale (for viscoelasticity): Long-term**

Accounting for Compressibility

Compressibility can be defined by specifying nonzero values for D_i (except for the Marlow, Valanis-Landel and Hencky models), by setting the Poisson's ratio to a value less than 0.5, or by providing test data that characterize the compressibility. The test data method is described later in this section. If you specify the Poisson's ratio for hyperelasticity other than the Marlow or the Valanis-Landel models, Abaqus computes the initial bulk modulus from the initial shear modulus

$$D_1 = \frac{2}{K_0} = \frac{3(1-2\nu)}{\mu_0(1+\nu)}.$$

For the Marlow model and the Valanis-Landel model the specified Poisson's ratio represents a constant value, which determines the volumetric response throughout the deformation process. If D_1 is equal to zero, all the D_i must be equal to zero. In such a case the material is assumed to be fully incompressible in Abaqus/Standard, while Abaqus/Explicit assumes compressible behavior with $K_0/\mu_0 = 20$ (Poisson's ratio of 0.475).

Input File Usage: **HYPERELASTIC*, POISSON=ν

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; any **Strain energy potential** except **Unknown** or **User-defined**: **Input source: Test data**: **Poisson's ratio:** ν

Specifying Material Coefficients Directly

The parameters of the hyperelastic strain energy potentials can be given directly as functions of temperature for all forms of the strain energy potential except the Marlow and Valanis-Landel forms.

Input File Usage: Use one of the following options:

**HYPERELASTIC*, ARRUDA-BOYCE
**HYPERELASTIC*, HENCKY
**HYPERELASTIC*, MOONEY-RIVLIN
**HYPERELASTIC*, NEO HOOKE
**HYPERELASTIC*, OGDEN, N=*n* ($n \le 6$)
**HYPERELASTIC*, POLYNOMIAL, N=*n* ($n \le 6$)

*HYPERELASTIC, REDUCED POLYNOMIAL, N=n ($n \leq 6$)
*HYPERELASTIC, VAN DER WAALS
*HYPERELASTIC, YEOH

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Coefficients** and **Strain energy potential: Arruda-Boyce**, **Mooney-Rivlin**, **Neo Hooke**, **Ogden**, **Polynomial**, **Reduced Polynomial**, **Van der Waals**, or **Yeoh**

The Hencky model is not supported in Abaqus/CAE.

Using Test Data to Calibrate Material Coefficients

The material coefficients of the hyperelastic models can be calibrated by Abaqus from experimental stress-strain data, except for the Hencky model. In the case of the Marlow and Valanis-Landel models, the test data directly characterize the strain energy potential (there are no material coefficients for these models); these models are described in detail below. The value of N and experimental stress-strain data can be specified for up to four simple tests: uniaxial, equibiaxial, planar, and, if the material is compressible, a volumetric compression test. Abaqus will then compute the material parameters. The material constants are determined through a least-squares-fit procedure, which minimizes the relative error in stress. For the n nominal-stress–nominal-strain data pairs, the relative error measure E is minimized, where

$$E = \sum_{i=1}^{n} \left(1 - T_i^{\text{th}}/T_i^{\text{test}}\right)^2.$$

T_i^{test} is a stress value from the test data, and T_i^{th} comes from one of the nominal stress expressions derived below (see "Experimental tests"). Abaqus minimizes the relative error rather than an absolute error measure since this provides a better fit at lower strains. This method is available for all strain energy potentials and any order of N except for the polynomial form, where a maximum of $N = 2$ is allowed. The polynomial models are linear in terms of the constants C_{ij}; therefore, a linear least-squares procedure can be used. The Arruda-Boyce, Ogden, and Van der Waals potentials are nonlinear in some of their coefficients, thus necessitating the use of a nonlinear least-squares procedure. *Fitting of hyperelastic and hyperfoam constants* contains a detailed derivation of the related equations.

It is generally best to obtain data from several experiments involving different kinds of deformation over the range of strains of interest in the actual application and to use all of these data to determine the parameters. This is particularly true for the phenomenological models; that is, the Ogden and the polynomial models. It has been observed that to achieve good accuracy and stability, it is necessary to fit these models using test data from more than one deformation state. In some cases, especially at large strains, removing the dependence on the second invariant might alleviate this limitation. The Arruda-Boyce, neo-Hookean, and Van der Waals models with $\beta = 0$ offer a physical interpretation and provide a better prediction of general deformation modes when the parameters are based on only one test. An extensive discussion of this topic can be found in *Hyperelastic material behavior*.

This method does not allow the hyperelastic properties to be temperature dependent. However, if temperature-dependent test data are available, several curve fits can be conducted by performing a data check analysis on a simple input file. The temperature-dependent coefficients determined by Abaqus can then be entered directly in the actual analysis run.

Optionally, the parameter β in the Van der Waals model can be set to a fixed value while the other parameters are found using a least-squares curve fit.

As many data points as required can be entered from each test. It is recommended that data from all four tests (on samples taken from the same piece of material) be included and that the data points cover the range of nominal

strains expected to arise in the actual loading. For the (general) polynomial and Ogden models and for the coefficient β in the Van der Waals model, the planar test data must be accompanied by the uniaxial test data, the biaxial test data, or both of these types of test data; otherwise, the solution to the least-squares fit will not be unique.

The strain data should be given as nominal strain values (change in length per unit of original length). For the uniaxial, equibiaxial, and planar tests stress data are given as nominal stress values (force per unit of original cross-sectional area). These tests allow for entering both compression and tension data. Compressive stresses and strains are entered as negative values.

If compressibility is to be specified, the D_i or D can be computed from volumetric compression test data. Alternatively, compressibility can be defined by specifying a Poisson's ratio, in which case Abaqus computes the bulk modulus from the initial shear modulus. If no such data are given, Abaqus/Standard assumes that D or all the D_i are zero, whereas Abaqus/Explicit assumes compressibility corresponding to a Poisson's ratio of 0.475 (see "Compressibility in Abaqus/Explicit" above). For these compression tests the stress data are given as pressure values.

Input File Usage: Use one of the following options to select the strain energy potential:

**HYPERELASTIC*, TEST DATA INPUT, ARRUDA-BOYCE
**HYPERELASTIC*, TEST DATA INPUT, MOONEY-RIVLIN
**HYPERELASTIC*, TEST DATA INPUT, NEO HOOKE
**HYPERELASTIC*, TEST DATA INPUT, OGDEN, N=n ($n \leq 6$)
**HYPERELASTIC*, TEST DATA INPUT, POLYNOMIAL, N=n ($n \leq 2$)
**HYPERELASTIC*, TEST DATA INPUT, REDUCED POLYNOMIAL, N=n ($n \leq 6$)
**HYPERELASTIC*, TEST DATA INPUT, VAN DER WAALS
**HYPERELASTIC*, TEST DATA INPUT, VAN DER WAALS, BETA=β ($0 \leq \beta \leq 1$)
**HYPERELASTIC*, TEST DATA INPUT, YEOH

In addition, use at least one and up to four of the following options to give the test data (see "Experimental tests" below):

**UNIAXIAL TEST DATA*
**BIAXIAL TEST DATA*
**PLANAR TEST DATA*
**VOLUMETRIC TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Strain energy potential: Arruda-Boyce**, **Mooney-Rivlin**, **Neo Hooke**, **Ogden**, **Polynomial**, **Reduced Polynomial**, **Van der Waals** (**Beta: Fitted value** or **Specify**), or **Yeoh**

In addition, use at least one and up to four of the following options to give the test data (see "Experimental tests" below):

Test Data->Uniaxial Test Data
Test Data->Biaxial Test Data
Test Data->Planar Test Data
Test Data->Volumetric Test Data

Alternatively, you can select **Strain energy potential: Unknown** to define the material temporarily without specifying a particular strain energy potential. Then select **Material->Evaluate** to have Abaqus/CAE evaluate the material to determine the optimal strain energy potential.

Specifying the Marlow Model

The Marlow model assumes that the strain energy potential is independent of the second deviatoric invariant $\bar{I}_2$. This model is defined by providing test data that define the deviatoric behavior, and, optionally, the volumetric behavior if compressibility must be taken into account. Abaqus will construct a strain energy potential that reproduces the test data exactly, as shown in *Figure 1*.

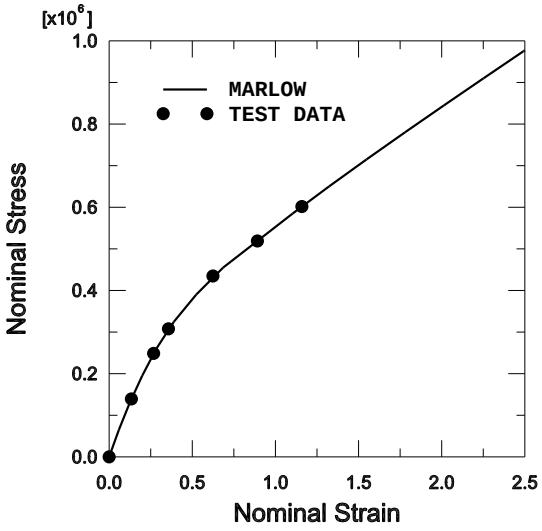


Figure 1: The results of the Marlow model with test data.

The interpolation and extrapolation of stress-strain data with the Marlow model is approximately linear for small and large strains. For intermediate strains in the range 0.1 to 1.0 a noticeable degree of nonlinearity might be observed in the interpolation/extrapolation with the Marlow model; for example, some nonlinearity is apparent between the 4th and 5th data points in *Figure 1*. To minimize undesirable nonlinearity, make sure that enough data points are specified in the intermediate strain range.

The deviatoric behavior is defined by specifying uniaxial, biaxial, or planar test data. Generally, you can specify either the data from tension tests or the data from compression tests because the tests are equivalent (see *Equivalent Experimental Tests*). However, for beams, trusses, and rebars, the data from tension and compression tests can be specified together. Volumetric behavior is defined by using one of the following three methods:

- Specify nominal lateral strains, in addition to nominal stresses and nominal strains, as part of the uniaxial, biaxial, or planar test data.
- Specify Poisson's ratio for the hyperelastic material.
- Specify volumetric test data directly. Both hydrostatic tension and hydrostatic compression data can be specified. If only hydrostatic compression data are available, as is usually the case, Abaqus will assume that the hydrostatic pressure is an antisymmetric function of the nominal volumetric strain, $\epsilon_{vol} = J_{vol} - 1$.

If you do not define volumetric behavior, Abaqus/Standard assumes fully incompressible behavior, while Abaqus/Explicit assumes compressibility corresponding to a Poisson's ratio of 0.475.

Material test data in which the stress does not vary smoothly with increasing strain might lead to convergence difficulty during the simulation. It is highly recommended that smooth test data be used to define the Marlow form. Abaqus provides a smoothing algorithm, which is described in detail later in this section.

The test data for the Marlow model can also be given as a function of temperature and field variables. You must specify the number of user-defined field variable dependencies required.

Uniaxial, biaxial, and planar test data must be given in ascending order of the nominal strains; volumetric test data must be given in descending order of the volume ratio.

Input File Usage: To define the Marlow test data as a function of temperature and/or field variables, use the following option:

**HYPERELASTIC*, MARLOW

with one of the following first three options and, optionally, the fourth option:

**UNIAXIAL TEST DATA*, DEPENDENCIES=*n*
**BIAXIAL TEST DATA*, DEPENDENCIES=*n*
**PLANAR TEST DATA*, DEPENDENCIES=*n*
**VOLUMETRIC TEST DATA*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Strain energy potential: Marlow**

In addition, select one of the following first three options and, optionally, the fourth option to give the test data (see “Experimental tests” below):

Test Data->Uniaxial Test Data
Test Data->Biaxial Test Data
Test Data->Planar Test Data
Test Data->Volumetric Test Data

In each of the **Test Data Editor** dialog boxes, you can toggle on **Use temperature-dependent data** to define the test data as a function of temperature and select the **Number of field variables** to define the test data as a function of field variables.

Alternatively, you can select **Material->Evaluate** to have Abaqus/CAE evaluate the material. If you included temperature dependencies, field variable dependencies, or lateral nominal strain in the test data—which can only be defined in the **Marlow** hyperelastic definition—**Marlow** will be the only strain energy potential available for evaluation.

Specifying the Valanis-Landel Model

In general, the deviatoric part of the strain energy potential of the Valanis-Landel model depends on both the first, $\bar{I}_1$, and the second, $\bar{I}_2$, deviatoric invariants. You define this model by providing test data and, optionally, the Poisson's ratio. Abaqus constructs a strain energy potential that reproduces the test data exactly.

You define the deviatoric behavior by specifying uniaxial test data or uniaxial and biaxial test data. If the deviatoric behavior is defined by specifying only uniaxial test data, you must provide data from both tension and compression tests together for this model. If the deviatoric behavior is defined by specifying both uniaxial and biaxial test data, you must provide data from tension tests only in both cases. In this case, Abaqus will convert the tensile biaxial data into compressive uniaxial test data since both data sets are equivalent (see *Figure 3*). The conversion is performed by assuming that the deformations are incompressible. It can be shown (using the relations provided in *Experimental Tests*) that the uniaxial nominal stress, T_U, and the nominal strain, ϵ_U, can be obtained from the following relations:

$$\epsilon_U = \frac{1}{(1+\epsilon_B)} - 1$$

and

$$T_U = -\frac{(1+\epsilon_B)}{(1+\epsilon_U)} T_B$$

where ϵ_B and T_B are the nominal biaxial strain and stress, respectively.

Volumetric behavior is defined by using one of the following methods:

- Specify nominal lateral strains, in addition to nominal stresses and nominal strains, as part of the uniaxial test data.
- Specify Poisson's ratio for the hyperelastic material.
- Specify volumetric test data directly. You can specify both hydrostatic tension and hydrostatic compression data. If only hydrostatic compression data are available, as is usually the case, Abaqus assumes that the hydrostatic pressure is an antisymmetric function of the nominal volumetric strain, $\epsilon_{vol} = J_{vol} - 1$.

If a Poisson's ratio of 0.5 is specified, the material behavior is assumed to be fully incompressible in Abaqus/Standard.

Material test data in which the stress does not vary smoothly with increasing strain can lead to convergence difficulty during the simulation. It is highly recommended that you use smooth test data to define the Valanis-Landel form. Abaqus provides a smoothing algorithm, which is described in detail later in this section.

The test data for the Valanis-Landel model can also be given as a function of temperature and field variables. You must specify the number of user-defined field variable dependencies required.

Uniaxial and biaxial test data must be given in ascending order of the nominal strains.

Input File Usage: To define the Valanis-Landel test data as a function of temperature and/or field variables, use both of the following options:

**HYPERELASTIC*, VALANIS-LANDEL
**UNIAXIAL TEST DATA*, DEPENDENCIES=*n*

and, optionally,

**BIAXIAL TEST DATA*, DEPENDENCIES=*n*
**VOLUMETRIC TEST DATA*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Strain energy potential: Valanis-Landel**

User Subroutine Specification in Abaqus/Standard

An alternative method provided in Abaqus/Standard for defining the hyperelastic material parameters allows the strain energy potential to be defined in user subroutine *UHYPER* or in user subroutine *UHYPER_STRETCH*. You can specify either compressible or incompressible behavior. Optionally, you can specify the number of property values needed as data in the user subroutine. If needed, you can specify the number of solution-dependent variables (see *About User Subroutines and Utilities*).

User subroutine *UHYPER* requires that the values of the derivatives of the strain energy density function of the hyperelastic material are defined with respect to the strain invariants.

User subroutine *UHYPER_STRETCH* assumes that the strain energy potential uses the Valanis-Landel form. It requires that the values of the derivatives of the strain energy density function of the hyperelastic material are defined with respect to the principal deviatoric stretches, $\overline{\lambda}_i$, and elastic volume ratio, J^{el}.

Input File Usage: Use one of the following options to specify the strain energy potential in user subroutine *UHYPER*:

**HYPERELASTIC*, USER, FORMULATION=INVARIANT,
TYPE=COMPRESSIBLE, PROPERTIES=*n*
**HYPERELASTIC*, USER, FORMULATION=INVARIANT,
TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

Use one of the following options to specify the strain energy potential in user subroutine *UHYPER_STRETCH*:

**HYPERELASTIC*, USER, FORMULATION=STRETCH, TYPE=COMPRESSIBLE,
PROPERTIES=*n*
**HYPERELASTIC*, USER, FORMULATION=STRETCH,
TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Coefficients** and **Strain energy potential: User-defined**: optionally, toggle on **Include compressibility** and/or specify the **Number of property values**

User Subroutine Specification in Abaqus/Explicit

Abaqus/Explicit lets you define the strain energy potential in user subroutine *VUHYPER* as a function of strain invariants. User subroutine *VUHYPER* needs to define the values of the derivatives of the strain energy density function of the hyperelastic material with respect to the strain invariants. Fully incompressible material behavior is not supported in user subroutine *VUHYPER* for plane strain, axisymmetric, and three-dimensional solid elements. However, for elements with reduced stress state (plane stress, beam, and uniaxial elements), user subroutine *VUHYPER* assumes fully incompressible behavior and the derivatives of strain energy with respect to the third invariant are ignored. Optionally, you can specify the number of properties values needed as data in the user subroutine as well as the number of solution-dependent variables, if required (see *About User Subroutines and Utilities*).

User subroutine *VUHYPER_STRETCH* assumes that the strain energy potential uses the Valanis-Landel form. It requires that the values of the derivatives of the strain energy density function of the hyperelastic material are defined with respect to the principal deviatoric stretches, $\overline{\lambda}_i$, and elastic volume ratio, J^{el}. At present, beam elements are not supported with *VUHYPER_STRETCH*.

Input File Usage: Use one of the following options to specify the strain energy potential in user subroutine *VUHYPER*:

**HYPERELASTIC*, USER, FORMULATION=INVARIANT, PROPERTIES=*n*

Use one of the following options to specify the strain energy potential in user subroutine *VUHYPER_STRETCH*:

**HYPERELASTIC*, USER, FORMULATION=STRETCH, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Coefficients** and **Strain energy potential: User-defined**: optionally, specify the **Number of property values**

Experimental Tests

For a homogeneous material, homogeneous deformation modes suffice to characterize the material constants. Abaqus accepts test data from the following deformation modes:

- Uniaxial tension and compression
- Equibiaxial tension and compression
- Planar tension and compression (also known as pure shear)
- Volumetric tension and compression

These modes are illustrated schematically in *Figure 2* and are described below. The most commonly performed experiments are uniaxial tension, uniaxial compression, and planar tension.

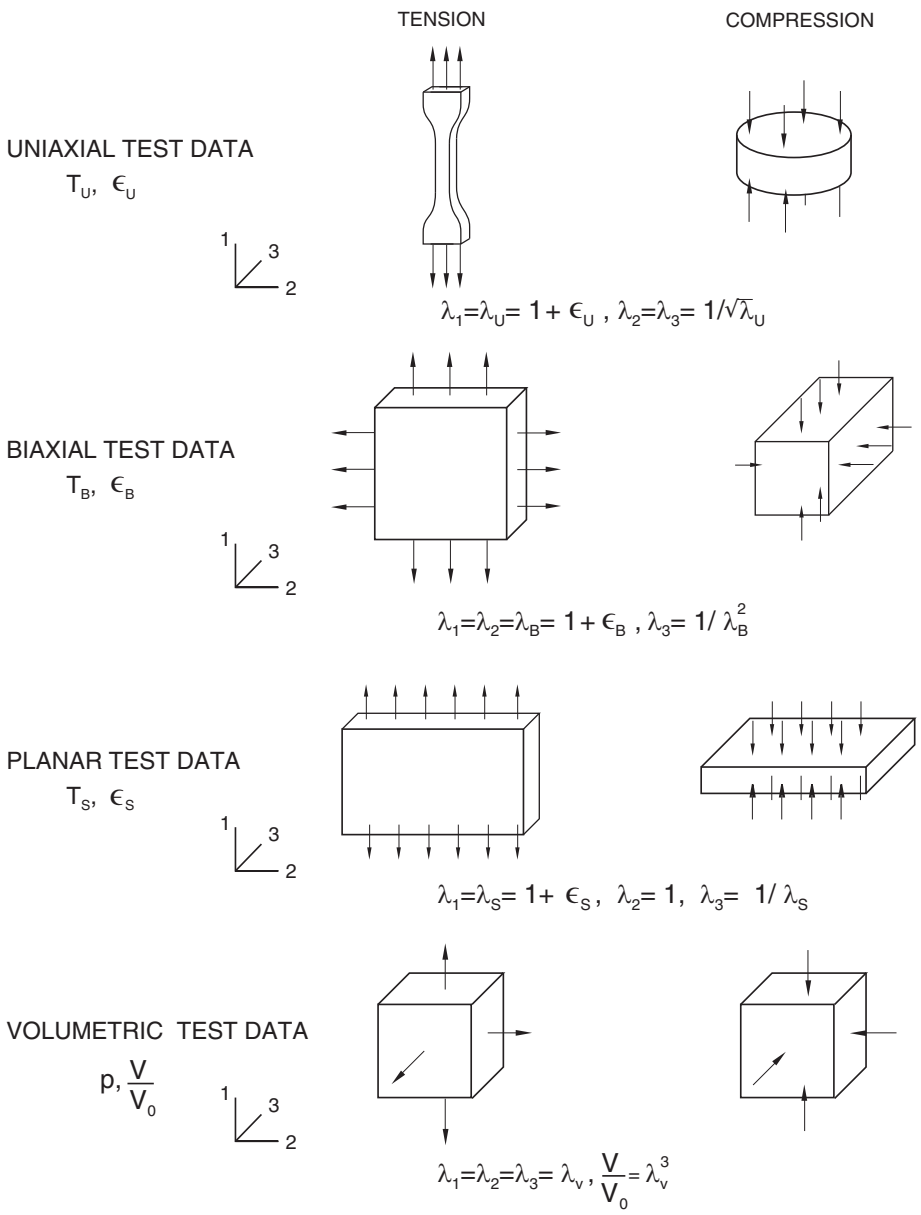


Figure 2: Schematic illustrations of deformation modes.

Combine data from these three test types to get a good characterization of the hyperelastic material behavior.

For the incompressible version of the material model, the stress-strain relationships for the different tests are developed using derivatives of the strain energy function with respect to the strain invariants. We define these relations in terms of the nominal stress (the force divided by the original, undeformed area) and the nominal, or engineering, strain defined below.

The deformation gradient, expressed in the principal directions of stretch, is

$$\mathbf{F} = \begin{bmatrix} \boldsymbol{\lambda_1} & 0 & 0 \\ 0 & \boldsymbol{\lambda_2} & 0 \\ 0 & 0 & \boldsymbol{\lambda_3} \end{bmatrix},$$

where λ_1, λ_2, and λ_3 are the principal stretches: the ratios of current length to length in the original configuration in the principal directions of a material fiber. The principal stretches, λ_i, are related to the principal nominal strains, ϵ_i, by

$$\lambda_i = 1 + \epsilon_i.$$

Because we assume incompressibility and isothermal response, $J = \det(\mathbf{F}) = 1$ and, hence, $\lambda_1\lambda_2\lambda_3 = 1$. The deviatoric strain invariants in terms of the principal stretches are then

$$\bar{I}_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2,$$

and

$$\bar{I}_2 = \lambda_1^{-2} + \lambda_2^{-2} + \lambda_3^{-2}.$$

Uniaxial Tests

The uniaxial deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_U\ , \quad \lambda_2 = \lambda_3 = 1/\sqrt{\lambda_U},$$

where λ_U is the stretch in the loading direction. The nominal strain is defined by $\varepsilon_U = \lambda_U - 1.$

To derive the uniaxial nominal stress T_U, we invoke the principle of virtual work:

$$\delta U = T_U \delta\lambda_U,$$

so that

$$T_U = \frac{\partial U}{\partial \lambda_U} = 2\left(1 - \lambda_U^{-3}\right)\left(\lambda_U \frac{\partial U}{\partial \bar{I}_1} + \frac{\partial U}{\partial \bar{I}_2}\right).$$

The uniaxial tension test is the most common of all the tests and is usually performed by pulling a “dog-bone” specimen. The uniaxial compression test is performed by loading a compression button between lubricated surfaces. The loading surfaces are lubricated to minimize any barreling effect in the button that would cause deviations from a homogeneous uniaxial compression stress-strain state.

Input File Usage: **UNIAXIAL TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data->Uniaxial Test Data**

Equibiaxial Tests

The equibiaxial deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_2 = \lambda_B\ , \quad \lambda_3 = 1/\lambda_B^2,$$

where λ_B is the stretch in the two perpendicular loading directions. The nominal strain is defined by $\varepsilon_B = \lambda_B - 1.$

To develop the expression for the equibiaxial nominal stress, T_B, we again use the principle of virtual work (assuming that the stress perpendicular to the loading direction is zero),

$$\delta U = 2T_B \delta\lambda_B,$$

so that

$$T_B = \frac{1}{2}\frac{\partial U}{\partial \lambda_B} = 2\left(\lambda_B - \lambda_B^{-5}\right)\left(\frac{\partial U}{\partial \bar{I}_1} + \lambda_B^2 \frac{\partial U}{\partial \bar{I}_2}\right).$$

In practice, the equibiaxial compression test is rarely performed because of experimental setup difficulties. In addition, this deformation mode is equivalent to a uniaxial tension test, which is straightforward to conduct.

A more common test is the equibiaxial tension test, in which a stress state with two equal tensile stresses and zero shear stress is created. This state is usually achieved by stretching a square sheet in a biaxial testing machine. It can also be obtained by inflating a circular membrane into a spheroidal shape (like blowing up a balloon). The stress field in the middle of the membrane then closely approximates equibiaxial tension, provided that the thickness of the membrane is very much smaller than the radius of curvature at this point. However, the strain distribution will not be quite uniform, and local strain measurements will be required. Once the strain and radius of curvature are known, the nominal stress can be derived from the inflation pressure.

Input File Usage: **BIAXIAL TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data->Biaxial Test Data**

Planar Tests

The planar deformation mode is characterized in terms of the principal stretches, λ_i, as

$$\lambda_1 = \lambda_S\ ,\ \lambda_2 = 1\ ,\ \lambda_3 = 1/\lambda_S,$$

where λ_S is the stretch in the loading direction. Then, the nominal strain in the loading direction is $\varepsilon_S = \lambda_S - 1.$

This test is also called a “pure shear” test since, in terms of logarithmic strains,

$$\varepsilon_1 = \ln \lambda_1 = -\ln \lambda_3 = -\varepsilon_3\ ,\quad \varepsilon_2 = \ln \lambda_2 = 0,$$

which corresponds to a state of pure shear at an angle of 45° to the loading direction.

The principle of virtual work gives

$$\delta U = T_S \delta \lambda_S,$$

where T_S is the nominal planar stress, so that

$$T_S = \frac{\partial U}{\partial \lambda_S} = 2\left(\lambda_S - \lambda_S^{-3}\right)\left(\frac{\partial U}{\partial \bar{I}_1} + \frac{\partial U}{\partial \bar{I}_2}\right).$$

For the (general) polynomial and Ogden models and for the coefficient β in the Van der Waals model this equation alone will not determine the constants uniquely. The planar test data must be augmented by uniaxial test data and biaxial test data to determine the material parameters.

Planar tests are usually done with a thin, short, and wide rectangular strip of material fixed on its wide edges to rigid loading clamps that are moved apart. If the separation direction is the *1*-direction and the thickness direction is the *3*-direction, the comparatively long size of the specimen in the *2*-direction and the rigid clamps allow us to use the approximation $\lambda_2 = 1$; that is, there is no deformation in the wide direction of the specimen. This deformation mode could also be called planar compression if the *3*-direction is considered to be the primary direction. All forms of incompressible plane strain behavior are characterized by this deformation mode. Consequently, if plane strain analysis is performed, planar test data represent the relevant form of straining of the material.

Input File Usage: **PLANAR TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data->Planar Test Data**

Volumetric Tests

The following discussion describes procedures for obtaining D_i values (or D, for the Arruda-Boyce and Van der Waals models) corresponding to the actual material behavior. With these values you can compare the material's initial bulk modulus, $K_0 = 2/D_1$, to its initial shear modulus ($\mu_0 = 2(C_{10} + C_{01})$ for the polynomial model, $\mu_0 = \sum_{i=1}^{N} \mu_i$ for Ogden's model) and then judge whether D_i values that will provide results are sufficiently realistic. For Abaqus/Explicit caution should be used; K_0/μ_0 should be less than 100. Otherwise, noisy solutions will be obtained and time increments will be excessively small (see "Compressibility in Abaqus/Explicit" above).

The D_i and D can be calculated from data obtained in pure volumetric compression of a specimen (volumetric tension tests are much more difficult to perform). In a pure volumetric test $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V$; therefore, $\bar{I}_1 = \bar{I}_2 = 3$ and $J = {\lambda_V}^3 = V/V_0$ (the volume ratio). Using the polynomial form of the strain energy potential, the total pressure stress on the specimen is obtained as

$$p = -\left(\frac{\sigma_1+\sigma_2+\sigma_3}{3}\right) = -\sum_{i=1}^{N} 2i\frac{1}{D_i}\left({\lambda_V}^3 - 1\right)^{2i-1}.$$

This equation can be used to determine the D_i. If we are using a second-order polynomial series for U, we have $N = 2$, and so two D_i are needed. Therefore, a minimum of two points on the pressure-volume ratio curve are required to give two equations for the D_i. For the Ogden and reduced polynomial potentials D_i can be determined for up to $N = 6$. A linear least-squares fit is performed when more than N data points are provided.

An approximate way of conducting a volumetric test consists of using a cylindrical rubber specimen that fits snugly inside a rigid container and whose top surface is compressed by a rigid piston. Although both volumetric and deviatoric deformation are present, the deviatoric stresses will be several orders of magnitude smaller than the hydrostatic stresses (because the bulk modulus is much higher than the shear modulus) and can be neglected. The compressive stress imposed by the rigid piston is effectively the pressure, and the volumetric strain in the rubber cylinder is computed from the piston displacement.

Nonzero values of D_i affect the uniaxial, equibiaxial, and planar stress results. However, since the material is assumed to be only slightly compressible, the techniques described for obtaining the deviatoric coefficients should give sufficiently accurate values even though they assume that the material is fully incompressible.

Input File Usage: **VOLUMETRIC TEST DATA*

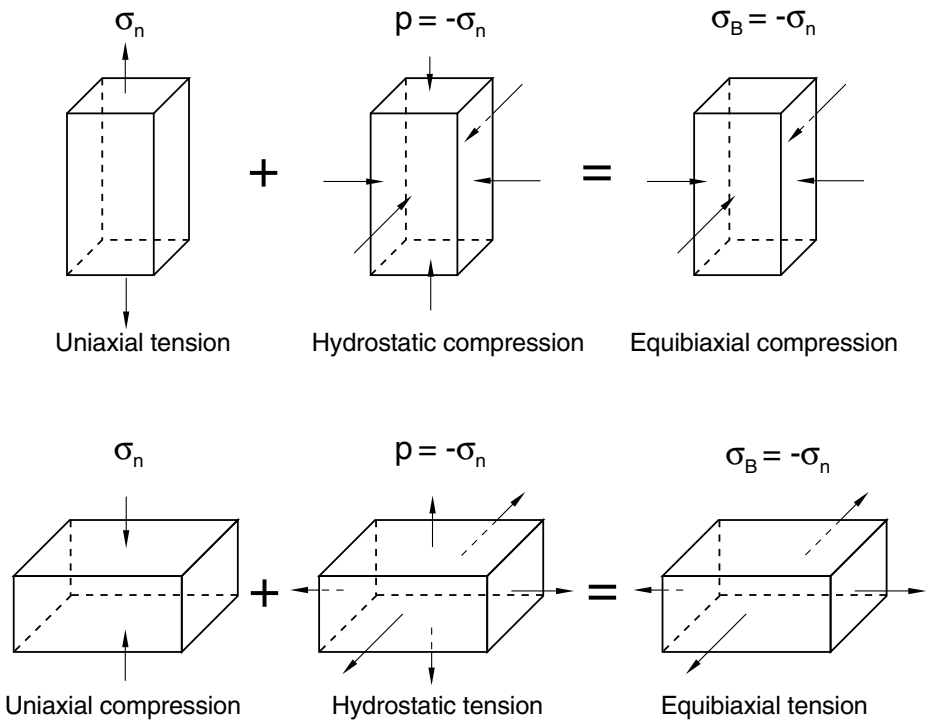
Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data->Volumetric Test Data**

Equivalent Experimental Tests

The superposition of a tensile or compressive hydrostatic stress on a loaded, fully incompressible elastic body results in different stresses but does not change the deformation. Thus, *Figure 3* shows that some apparently different loading conditions are actually equivalent in their deformations and, therefore, are equivalent tests:

- Uniaxial tension $\Longleftrightarrow$ Equibiaxial compression

- Uniaxial compression $\Longleftrightarrow$ Equibiaxial tension
- Planar tension $\Longleftrightarrow$ Planar compression



The stresses, σ_i, shown here are true (Cauchy) stresses and not nominal stresses.

Figure 3: Equivalent deformation modes through superposition of hydrostatic stress.

On the other hand, the tensile and compressive cases of the uniaxial and equibiaxial modes are independent from each other: uniaxial tension and uniaxial compression provide independent data.

Smoothing the Test Data

Experimental test data often contain noise in the sense that the test variable is both slowly varying and also corrupted by random noise. This noise can affect the quality of the strain energy potential that Abaqus derives. This noise is particularly a problem with the Marlow form, where a strain energy potential that exactly describes the test data that are used to calibrate the model is computed. It is less of a concern with the other forms, since smooth functions are fitted through the test data.

Abaqus provides a smoothing technique to remove the noise from the test data based on the Savitzky-Golay method. The idea is to replace each data point by a local average of its surrounding data points, so that the level of noise can be reduced without biasing the dominant trend of the test data. In the implementation a cubic polynomial is fitted through each data point i and n data points to the immediate left and right of that point. A least-squares method is used to fit the polynomial through these $2n + 1$ points. The value of data point i is then replaced by the value of the polynomial at the same position. Each polynomial is used to adjust one data point except near the ends of the curve, where a polynomial is used to adjust multiple points, because the first and last few points cannot be the center of the fitting set of data points. This process is applied repeatedly to all data points until two consecutive passes through the data produce nearly the same results.

By default, the test data are not smoothed. If smoothing is specified, the default value is n=3. Alternatively, you can specify the number of data points to the left and right of a data point in the moving window within which a least-squares polynomial is fit.

Input File Usage: For the Marlow form, use one of the first three options and, optionally, the fourth option; for the Valanis-Landel model, use the first option or the first two options and, optionally, the fourth option; and for the other potential forms, use one and up to four of the following options:

**UNIAXIAL TEST DATA*, SMOOTH=n ($n \geq 2$)
**BIAXIAL TEST DATA*, SMOOTH=n ($n \geq 2$)
**PLANAR TEST DATA*, SMOOTH=n ($n \geq 2$)
**VOLUMETRIC TEST DATA*, SMOOTH=n ($n \geq 2$)

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Hyperelastic**: **Material type: Isotropic**; **Input source: Test data** and **Test Data**->**Uniaxial Test Data**, **Biaxial Test Data**, **Planar Test Data**, or **Volumetric Test Data**

In each of the **Test Data Editor** dialog boxes, toggle on **Apply smoothing**, and select a value for n ($n \geq 2$).

Model Prediction of Material Behavior Versus Experimental Data

Once the strain energy potential is determined, the behavior of the hyperelastic model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the strain energy potentials determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. You can evaluate the hyperelastic behavior automatically in Abaqus/CAE. Alternatively, single-element test cases can be used to derive the nominal stress–nominal strain response of the material model.

See *Fitting of rubber test data*, which illustrates the entire process of fitting hyperelastic constants to a set of test data.

Hyperelastic Material Stability

An important consideration in judging the quality of the fit to experimental data is the concept of material or Drucker stability. Abaqus checks the Drucker stability of the material for the first three deformation modes described above.

The Drucker stability condition for an incompressible material requires that the change in the stress, $d\boldsymbol{\sigma}$, following from any infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\sigma} : d\boldsymbol{\varepsilon} > 0.$$

Using $d\boldsymbol{\sigma} = \mathbf{D} : d\boldsymbol{\varepsilon}$, where $\mathbf{D}$ is the tangent material stiffness, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0,$$

thus requiring the tangential material stiffness to be positive-definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains,

$$d\sigma_1 d\varepsilon_1 + d\sigma_2 d\varepsilon_2 + d\sigma_3 d\varepsilon_3 > 0.$$

As before, since the material is assumed to be incompressible, we can choose any value for the hydrostatic pressure without affecting the strains. A convenient choice for the stability calculation is $\sigma_3 = d\sigma_3 = 0$, which allows us to ignore the third term in the above equation.

The relation between the changes in stress and in strain can then be obtained in the form of the matrix

$$\begin{pmatrix} d\sigma_1 \\ d\sigma_2 \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} \begin{pmatrix} d\varepsilon_1 \\ d\varepsilon_2 \end{pmatrix},$$

where $D_{ij} = D_{ij}(\lambda_1, \lambda_2, \lambda_3)$. For material stability $\mathbf{D}$ must be positive-definite; thus, it is necessary that

$$D_{11} + D_{22} > 0,$$
$$D_{11}D_{22} - D_{12}D_{21} > 0.$$

This stability check is performed for the polynomial models, the Ogden potential, the Van der Waals form, the Marlow form, and the Valanis-Landel form. The Arruda-Boyce form is always stable for positive values of (μ, λ_m); hence, it suffices to check the material coefficients to ensure stability.

You should be careful when defining the C_{ij} or (μ_i, α_i) for the polynomial models or the Ogden form: especially when $N > 1$, the behavior at higher strains is strongly sensitive to the values of the C_{ij} or (μ_i, α_i), and unstable material behavior might result if these values are not defined correctly. When some of the coefficients are strongly negative, instability at higher strain levels is likely to occur.

Abaqus performs a check on the stability of the material for six different forms of loading—uniaxial tension and compression, equibiaxial tension and compression, and planar tension and compression—for $0.1 \leq \lambda_1 \leq 10.0$ (nominal strain range of $-0.9 \leq \epsilon_1 \leq 9.0$) at intervals $\Delta\lambda_1 = 0.01$. If an instability is found, Abaqus issues a warning message and prints the lowest absolute value of ϵ_1 for which the instability is observed. Ideally, no instability occurs. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you either change the material model or carefully examine and revise the material input data. If user subroutine *UHYPER*, user subroutine *UHYPER_STRETCH*, *VUHYPER* or *VUHYPER_STRETCH* is used to define the hyperelastic material, you are responsible for ensuring stability.

Improving the Accuracy and Stability of the Test Data Fit

Unfortunately, the initial fit of the models to experimental data might not come out as well as expected. This is particularly true for the most general models, such as the (general) polynomial model and the Ogden model. For some of the simpler models, stability is assured by following some simple rules.

- For positive values of the initial shear modulus, μ, and the locking stretch, λ_m, the Arruda-Boyce form is always stable.
- For positive values of the coefficient C_{10} the neo-Hookean form is always stable.
- Given positive values of the initial shear modulus, μ, and the locking stretch, λ_m, the stability of the Van der Waals model depends on the global interaction parameter, a.
- For the Yeoh model stability is assured if all $C_{i0} > 0$. Typically, however, C_{20} will be negative, since this helps capture the S-shape feature of the stress-strain curve. Thus, reducing the absolute value of C_{20} or magnifying the absolute value of C_{10} will help make the Yeoh model more stable.

In all cases the following suggestions might improve the quality of the fit:

- Both tension and compression data are allowed; compressive stresses and strains are entered as negative values. Use compression or tension data depending on the application: it is difficult to fit a single material model accurately to both tensile and compressive data.
- Always use many more experimental data points than unknown coefficients.
- If $N \geq 3$ is used, experimental data should be available to at least 100% tensile strain or 50% compressive strain.
- Perform different types of tests (for example, compression and simple shear tests). Proper material behavior for a deformation mode requires test data to characterize that mode.
- Check for warning messages about material instability or error messages about lack of convergence in fitting the test data. This check is especially important with new test data; a simple finite element model with the new test data can be run through the analysis input file processor to check the material stability.
- Use the material evaluation capability in Abaqus/CAE to compare the response curves for different strain energy potentials to the experimental data. Alternatively, you can perform one-element simulations for simple deformation modes and compare the Abaqus results against the experimental data. The *X*–*Y* plotting options in the Visualization module of Abaqus/CAE can be used for this comparison.
- Delete some data points at very low strains if large strains are anticipated. A disproportionate number of low strain points might unnecessarily bias the accuracy of the fit toward the low strain range and cause greater errors in the large strain range.
- Delete some data points at the highest strains if small to moderate strains are expected. The high strain points might force the fitting to lose accuracy and stability in the low strain range.
- Pick data points at evenly spaced strain intervals over the expected range of strains, which will result in similar accuracy throughout the entire strain range.
- The higher the order of *N*, the more oscillations are likely to occur, leading to instabilities in the stress-strain curves. If the (general) polynomial model is used, lower the order of *N* from 2 to 1 (3 to 2 for Ogden), especially if the maximum strain level is low (say, less than 100% strain).
- If multiple types of test data are used and the fit still comes out poorly, some of the test data probably contain experimental errors. New tests might be needed. One way of determining which test data are erroneous is to first calibrate the initial shear modulus μ_0^{test} of the material. Then fit each type of test data separately in Abaqus and compute the shear modulus, μ_0^{fit}, from the material constants using the relations

$$\mu_0^{fit} = 2\left(C_{10} + C_{01}\right) \quad \text{(polynomial form)} \qquad \text{or} \qquad \mu_0^{fit} = \sum_{i=1}^{N} \mu_i \quad \text{(Ogden form)}.$$

Alternatively, the initial Young's modulus, E_0^{test}, can be calibrated and compared with

$$E_0^{fit} = 6\left(C_{10} + C_{01}\right) \quad \text{(polynomial form)} \qquad \text{or} \qquad E_0^{fit} = 3\sum_{i=1}^{N} \mu_i \quad \text{(Ogden form)}.$$

The values of μ_0^{fit} or E_0^{fit} that are most different from μ_0^{test} or E_0^{test} indicate the erroneous test data.

Elements

The hyperelastic material model can be used with solid (continuum) elements, finite-strain shells (except S4 in Abaqus/Standard), continuum shells, membranes, and one-dimensional elements (trusses and rebars).

In Abaqus/Standard, the hyperelastic material model can be also used with Timoshenko beams (B21, B22, B31, B31OS, B32, B32OS, PIPE21, PIPE22, PIPE31, PIPE32, and their “hybrid” equivalents). It cannot be used with

Euler-Bernoulli beams (B23, B23H, B33, and B33H) and small-strain shells (STRI3, STRI65, S4R5, S8R, S8R5, and S9R5).

In Abaqus/Explicit the hyperelastic material model can be also used with Timoshenko beams (B21, B22, B31, and B32).

Pure Displacement Formulation Versus Hybrid Formulation in Abaqus/Standard

For continuum elements in Abaqus/Standard hyperelasticity can be used with the pure displacement formulation elements or with the "hybrid" (mixed formulation) elements. Because elastomeric materials are usually almost incompressible, fully integrated pure displacement method elements are not recommended for use with this material, except for plane stress cases. If fully or selectively reduced-integration displacement method elements are used with the almost incompressible form of this material model, a penalty method is used to impose the incompressibility constraint in anything except plane stress analysis. The penalty method can sometimes lead to numerical difficulties; therefore, the fully or selectively reduced-integrated "hybrid" formulation elements are recommended for use with hyperelastic materials.

In general, an analysis using a single hybrid element will be only slightly more computationally expensive than an analysis using a regular displacement-based element. However, when the wavefront is optimized, the Lagrange multipliers might not be ordered independently of the regular degrees of freedom associated with the element. Thus, the wavefront of a very large mesh of second-order hybrid tetrahedra might be noticeably larger than that of an equivalent mesh using regular second-order tetrahedra. This might lead to significantly higher CPU costs, disk space, and memory requirements.

Incompatible Mode Elements in Abaqus/Standard

Incompatible mode elements should be used with caution in applications involving large strains. Convergence might be slow, and in hyperelastic applications inaccuracies might accumulate. Erroneous stresses might sometimes appear in incompatible mode hyperelastic elements that are unloaded after having been subjected to a complex deformation history.

Procedures

Hyperelasticity must always be used with geometrically nonlinear analyses (*General and Perturbation Procedures*).

Hyperelastic Behavior in Elastomeric Foams

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Stress Softening in Elastomeric Foams*
- **HYPERFOAM*
- **UNIAXIAL TEST DATA*
- **BIAXIAL TEST DATA*
- **PLANAR TEST DATA*
- **VOLUMETRIC TEST DATA*
- **SIMPLE SHEAR TEST DATA*
- **MULLINS EFFECT*
- *Creating a hyperfoam material model*

Overview

The elastomeric foam material model:

- is isotropic and nonlinear;
- is valid for cellular solids whose porosity permits very large volumetric changes;
- optionally allows the specification of energy dissipation and stress softening effects (see *Stress Softening in Elastomeric Foams*);
- can deform elastically to large strains, up to 90% strain in compression; and
- requires that geometric nonlinearity be accounted for during the analysis step (see *Defining an Analysis* and *General and Perturbation Procedures*), since it is intended for finite-strain applications.

Abaqus/Explicit also provides a separate foam material model intended to capture the strain-rate sensitive behavior of low-density elastomeric foams such as used in crash and impact applications (see *Low-Density Foams*).

Mechanical Behavior of Elastomeric Foams

Cellular solids are made up of interconnected networks of solid struts or plates that form the edges and faces of cells. Foams are made up of polyhedral cells that pack in three dimensions. The foam cells can be either open (e.g., sponge) or closed (e.g., flotation foam). Common examples of elastomeric foam materials are cellular polymers such as cushions, padding, and packaging materials that utilize the excellent energy absorption properties of foams: the energy absorbed by foams is substantially greater than that absorbed by ordinary stiff elastic materials for a certain stress level.

Another class of foam materials is crushable foams, which undergo permanent (plastic) deformation. Crushable foams are discussed in *Crushable Foam Plasticity Models*.

Foams are commonly loaded in compression. *Figure 1* shows a typical compressive stress-strain curve.

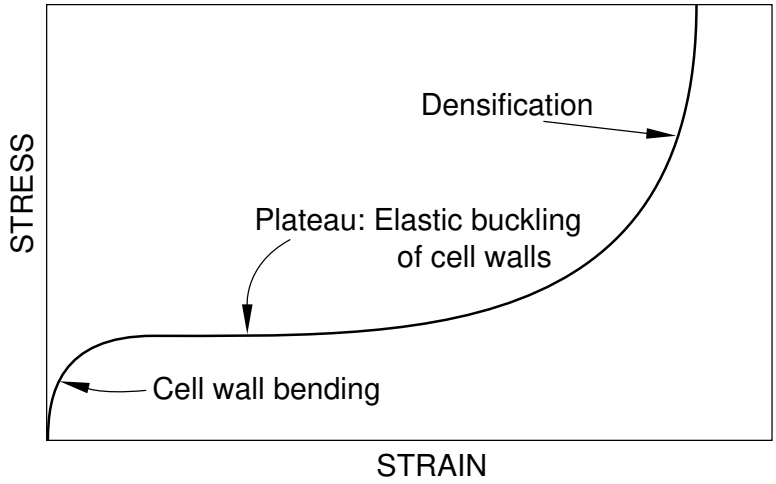


Figure 1: Typical compressive stress-strain curve.

Three stages can be distinguished during compression:

1. At small strains ($< 5\%$) the foam deforms in a linear elastic manner due to cell wall bending.
2. The next stage is a plateau of deformation at almost constant stress, caused by the elastic buckling of the columns or plates that make up the cell edges or walls. In closed cells the enclosed gas pressure and membrane stretching increase the level and slope of the plateau.
3. Finally, a region of densification occurs, where the cell walls crush together, resulting in a rapid increase of compressive stress. Ultimate compressive nominal strains of 0.7 to 0.9 are typical.

The tensile deformation mechanisms for small strains are similar to the compression mechanisms, but they differ for large strains. *Figure 2* shows a typical tensile stress-strain curve.

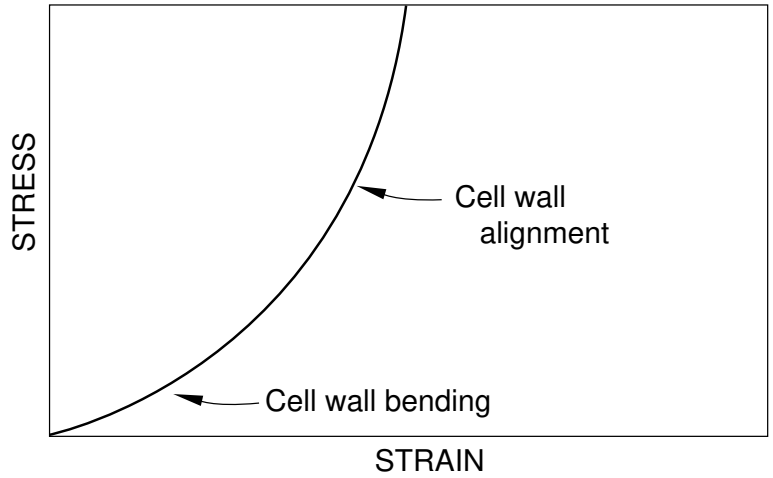


Figure 2: Typical tensile stress-strain curve.

There are two stages during tension:

1. At small strains the foam deforms in a linear, elastic manner as a result of cell wall bending, similar to that in compression.
2. The cell walls rotate and align, resulting in rising stiffness. The walls are substantially aligned at a tensile strain of about $\mathbf{1/3}$. Further stretching results in increased axial strains in the walls.

At small strains for both compression and tension, the average experimentally observed Poisson's ratio, ν, of foams is 1/3. At larger strains it is commonly observed that Poisson's ratio is effectively zero during compression: the buckling of the cell walls does not result in any significant lateral deformation. However, ν is nonzero during tension, which is a result of the alignment and stretching of the cell walls.

The manufacture of foams often results in cells with different principal dimensions. This shape anisotropy results in different loading responses in different directions. However, the hyperfoam model does not take this kind of initial anisotropy into account.

Strain Energy Potential

In the elastomeric foam material model the elastic behavior of the foams is based on the strain energy function

$$U = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left[\widehat{\lambda}_1^{\alpha_i} + \widehat{\lambda}_2^{\alpha_i} + \widehat{\lambda}_3^{\alpha_i} - 3 + \frac{1}{\beta_i} \left(\left(J^{el} \right)^{-\alpha_i \beta_i} - 1 \right) \right],$$

where N is a material parameter; μ_i, α_i, and β_i are temperature-dependent material parameters;

$$\widehat{\lambda}_i = \left(J^{th} \right)^{-\frac{1}{3}} \lambda_i \quad \rightarrow \quad \widehat{\lambda}_1 \widehat{\lambda}_2 \widehat{\lambda}_3 = J^{el};$$

and λ_i are the principal stretches. The elastic and thermal volume ratios, J^{el} and J^{th}, are defined below.

The coefficients μ_i are related to the initial shear modulus, μ_0, by

$$\mu_0 = \sum_{i=1}^{N} \mu_i,$$

while the initial bulk modulus, K_0, follows from

$$K_0 = \sum_{i=1}^{N} 2\mu_i \left(\frac{1}{3} + \beta_i \right).$$

For each term in the energy function, the coefficient β_i determines the degree of compressibility. β_i is related to the Poisson's ratio, ν_i, by the expressions

$$\beta_i = \frac{\nu_i}{1-2\nu_i}, \qquad \nu_i = \frac{\beta_i}{1+2\beta_i}.$$

Thus, if β_i is the same for all terms, we have a single effective Poisson's ratio, ν. This effective Poisson's ratio is valid for finite values of the logarithmic principal strains $\varepsilon_1, \varepsilon_2, \varepsilon_3$; in uniaxial tension $\varepsilon_2 = \varepsilon_3 = -\nu\varepsilon_1$.

Thermal Expansion

Only isotropic thermal expansion is permitted with the hyperfoam material model.

The elastic volume ratio, J^{el}, relates the total volume ratio (current volume/reference volume), J, and the thermal volume ratio, J^{th}:

$$J^{el} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = \left(1 + \varepsilon^{th} \right)^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient (*Thermal Expansion*).

Determining the Hyperfoam Material Parameters

The response of the material is defined by the parameters in the strain energy function, U; these parameters must be determined to use the hyperfoam model. Two methods are provided for defining the material parameters: you can specify the hyperfoam material parameters directly or specify test data and allow Abaqus to calculate the material parameters.

The elastic response of a viscoelastic material (*Time Domain Viscoelasticity*) can be specified by defining either the instantaneous response or the long-term response of such a material.

Instantaneous Response

To define the instantaneous response, the experiments outlined in *Experimental Tests* have to be performed within time spans much shorter than the characteristic relaxation time of the material.

Input File Usage: **HYPERFOAM*, MODULI=INSTANTANEOUS

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: **Moduli time scale (for viscoelasticity): Instantaneous**

Long-Term Response

If the long-term elastic response is used, data from experiments have to be collected after time spans much longer than the characteristic relaxation time of the viscoelastic material. Long-term elastic response is the default elastic material behavior.

Input File Usage: **HYPERFOAM*, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: **Moduli time scale (for viscoelasticity): Long-term**

Direct Specification

When the parameters N, μ_i, α_i, and ν_i are specified directly, they can be functions of temperature.

The default value of ν_i is zero, which corresponds to an effective Poisson's ratio of zero. The incompressible limit corresponds to all $\nu_i \to 0.5$. However, this material model should not be used for approximately incompressible materials: use of the hyperelastic model (*Hyperelastic Behavior of Rubberlike Materials*) is recommended if the effective Poisson's ratio $\nu_{eff} > 0.45$.

Input File Usage: **HYPERFOAM*, N=n ($n \le 6$)

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: **Strain energy potential order:** n ($n \le 6$); optionally, toggle on **Use temperature-dependent data**

Test Data Specification

The value of N and the experimental stress-strain data can be specified for up to five simple tests: uniaxial, equibiaxial, simple shear, planar, and volumetric. Abaqus contains a capability for obtaining the μ_i, α_i, and β_i for the hyperfoam model with up to six terms (N=6) directly from test data. Poisson effects can be included either by means of a constant Poisson's ratio or through specification of volumetric test data and/or lateral strains in the other test data.

It is important to recognize that the properties of foam materials can vary significantly from one batch to another. Therefore, all of the experiments should be performed on specimens taken from the same batch of material.

This method does not allow the properties to be temperature dependent.

As many data points as required can be entered from each test. Abaqus will then compute μ_i, α_i, and, if necessary, ν_i. The technique uses a least squares fit to the experimental data so that the relative error in the nominal stress is minimized.

It is recommended that data from the uniaxial, biaxial, and simple shear tests (on samples taken from the same piece of material) be included and that the data points cover the range of nominal strains expected to arise in the actual loading. The planar and volumetric tests are optional.

For all tests the strain data, including the lateral strain data, should be given as nominal strain values (change in length per unit of original length). For the uniaxial, equibiaxial, simple shear, and planar tests, stress data are given as nominal stress values (force per unit of original cross-sectional area). The tests allow for both compression and tension data; compressive stresses and strains should be entered as negative values. For the volumetric tests the stress data are given as pressure values.

Input File Usage: Use the first option to define an effective Poisson's ratio ($\nu_i = \nu$ for all i), or use the second option to define the lateral strains as part of the test data input:

**HYPERFOAM*, N=n, POISSON=ν, TEST DATA INPUT ($n \leq 6$)
**HYPERFOAM*, N=n, TEST DATA INPUT ($n \leq 6$).

In addition, use at least one and up to five of these additional options to give the experimental stress-strain data (see “Experimental tests” below):

**UNIAXIAL TEST DATA*
**BIAXIAL TEST DATA*
**PLANAR TEST DATA*
**SIMPLE SHEAR TEST DATA*
**VOLUMETRIC TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: toggle on **Use test data**; **Strain energy potential order:** n ($n \leq 6$); optionally, toggle on **Use constant Poisson's ratio:** and enter a value for the effective Poisson's ratio ($\nu_i = \nu$ for all i)

In addition, use at least one and up to five of the suboptions to give the experimental stress-strain data (see “Experimental tests” below):

Suboptions->Uniaxial Test Data

Suboptions->Biaxial Test Data

Suboptions->Planar Test Data

Suboptions->Simple Shear Test Data

Suboptions->Volumetric Test Data

Experimental Tests

For a homogeneous material, homogeneous deformation modes suffice to characterize the material parameters. Abaqus accepts test data from the following deformation modes:

- Uniaxial tension and compression
- Equibiaxial tension and compression
- Planar tension and compression (pure shear)
- Simple shear
- Volumetric tension and compression

The stress-strain relations are defined in terms of the nominal stress (the force divided by the original, undeformed area) and the nominal, or engineering, strains, ϵ_i. The principal stretches, λ_i, are related to the principal nominal strains, ϵ_i, by

$$\lambda_i = 1 + \epsilon_i.$$

Uniaxial, Equibiaxial, and Planar Tests

The deformation gradient, expressed in the principal directions of stretch, is

$$\mathbf{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},$$

where λ_1, λ_2, and λ_3 are the principal stretches: the ratios of current length to length in the original configuration in the principal directions of a material fiber. The deformation modes are characterized in terms of the principal stretches, λ_i, and the volume ratio, $J = \det(\mathbf{F})$. The elastomeric foams are not incompressible, so that $J = \lambda_1\lambda_2\lambda_3 \neq 1$. The transverse stretches, λ_2 and/or λ_3, are independently specified in the test data either as individual values from the measured lateral deformations or through the definition of an effective Poisson's ratio.

The three deformation modes use a single form of the nominal stress-stretch relation,

$$T_L = \frac{\partial U}{\partial \lambda_L} = \frac{2}{\lambda_L} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} \left(\lambda_L^{\alpha_i} - J^{-\alpha_i \beta_i} \right),$$

where T_L is the nominal stress and λ_L is the stretch in the loading direction. Because of the compressible behavior, the planar mode does not result in a state of pure shear. In fact, if the effective Poisson's ratio is zero, planar deformation is identical to uniaxial deformation.

Uniaxial Mode

In uniaxial mode $\lambda_1 = \lambda_U$, $\lambda_2 = \lambda_3$, and $J = \lambda_U \lambda_2^2$.

Input File Usage: **UNIAXIAL TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: toggle on **Use test data**, **Suboptions->Uniaxial Test Data**

Equibiaxial Mode

In equibiaxial mode $\lambda_1 = \lambda_2 = \lambda_B$ and $J = \lambda_B^2 \lambda_3$.

Input File Usage: **BIAXIAL TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: toggle on **Use test data**, **Suboptions->Biaxial Test Data**

Planar Mode

In planar mode $\lambda_1 = \lambda_P$, $\lambda_2 = 1$, and $J = \lambda_P \lambda_3$. Planar test data must be augmented by either uniaxial or biaxial test data.

Input File Usage: **PLANAR TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: toggle on **Use test data**, **Suboptions->Planar Test Data**

Simple Shear Tests

Simple shear is described by the deformation gradient

$$\mathbf{F} = \begin{bmatrix} 1 & \gamma & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where γ is the shear strain. For this deformation $J = \det(\mathbf{F}) = 1$. A schematic illustration of simple shear deformation is shown in *Figure 3*.

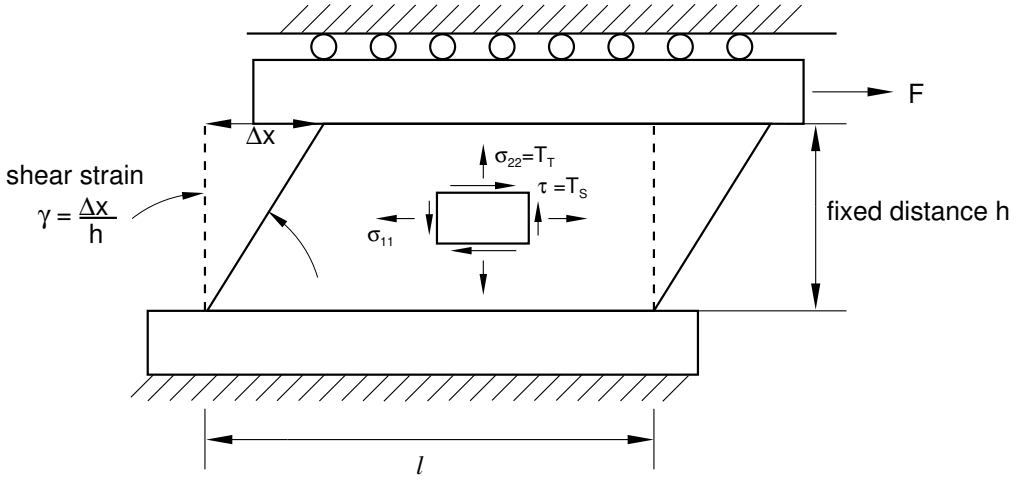


Figure 3: Simple shear test.

The nominal shear stress, T_S, is

$$T_S = \frac{\partial U}{\partial \gamma} = \sum_{j=1}^{2} \left\{ \frac{2\gamma}{2\left(\lambda_j^2 - 1\right) - \gamma^2} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} \left(\lambda_j^{\alpha_i} - 1 \right) \right\},$$

where λ_j are the principal stretches in the plane of shearing, related to the shear strain γ by

$$\lambda_{1,2} = \sqrt{1 + \frac{\gamma^2}{2} \pm \gamma \sqrt{1 + \frac{\gamma^2}{4}}}.$$

The stretch in the direction perpendicular to the shear plane is $\lambda_3 = 1$. The transverse (tensile) stress, T_T, developed during simple shear deformation due to the Poynting effect, is

$$T_T = \frac{\partial U}{\partial \varepsilon} = \sum_{j=1}^{2} \left\{ \frac{2\left(\lambda_j^2 - 1\right)}{2\lambda_j^4 - \lambda_j^2 (\gamma^2 + 2)} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} \left(\lambda_j^{\alpha_i} - 1 \right) \right\}.$$

Input File Usage: **SIMPLE SHEAR TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Hyperfoam**: toggle on **Use test data**, **Suboptions**->**Simple Shear Test Data**

Volumetric Tests

The deformation gradient, $\mathbf{F}$, is the same for volumetric tests as for uniaxial tests. The volumetric deformation mode consists of all principal stretches being equal;

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_V \ , \quad J = \lambda_V^3.$$

The pressure-volumetric ratio relation is

$$-p = \frac{\partial U}{\partial J} = \frac{2}{J} \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} \left(J^{\frac{1}{3}\alpha_i} - J^{-\alpha_i \beta_i} \right).$$

A volumetric compression test is illustrated in *Figure 4*. The pressure exerted on the foam specimen is the hydrostatic pressure of the fluid, and the decrease in the specimen volume is equal to the additional fluid entering the pressure chamber. The specimen is sealed against fluid penetration.

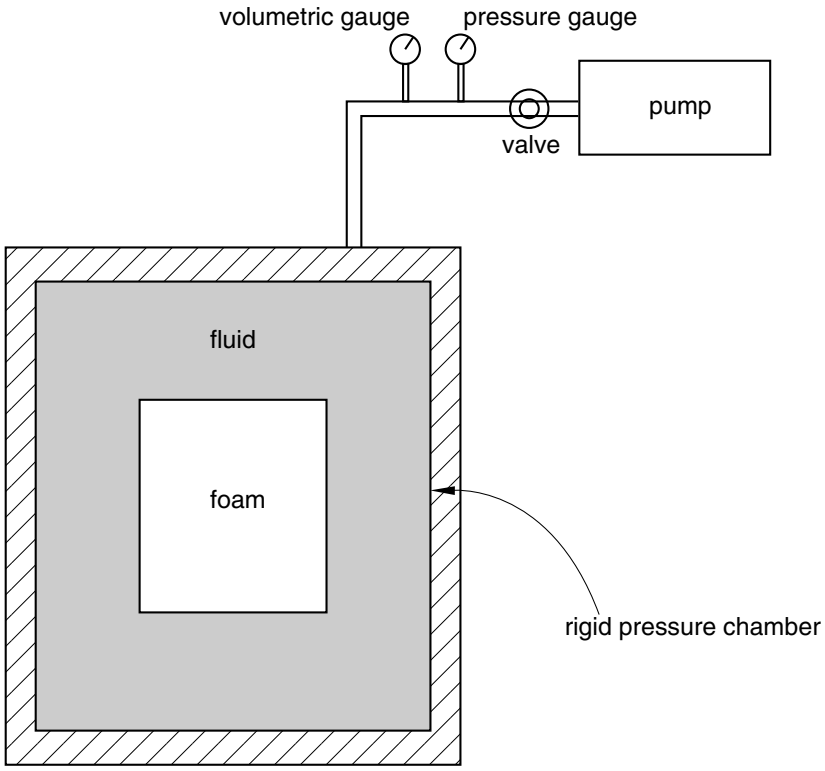


Figure 4: Volumetric compression test.

Input File Usage: **VOLUMETRIC TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperfoam**: toggle on **Use test data**, **Suboptions->Volumetric Test Data**

Difference between Compression and Tension Deformation

For small strains ($< 5\%$) foams behave similarly for both compression and tension. However, at large strains the deformation mechanisms differ for compression (buckling and crushing) and tension (alignment and stretching). Therefore, accurate hyperfoam modeling requires that the experimental data used to define the material parameters correspond to the dominant deformation modes of the problem being analyzed. If compression dominates, the pertinent tests are:

- Uniaxial compression
- Simple shear
- Planar compression (if Poisson's ratio $\boldsymbol{\nu \neq 0}$)
- Volumetric compression (if Poisson's ratio $\boldsymbol{\nu \neq 0}$)

If tension dominates, the pertinent tests are:

- Uniaxial tension
- Simple shear
- Biaxial tension (if Poisson's ratio $\boldsymbol{\nu \neq 0}$)

- Planar tension (if Poisson's ratio $\nu \neq 0$)

Lateral strain data can also be used to define the compressibility of the foam. Measurement of the lateral strains may make other tests redundant; for example, providing lateral strains for a uniaxial test eliminates the need for a volumetric test. However, if volumetric test data are provided in addition to the lateral strain data for other tests, both the volumetric test data and the lateral strain data will be used in determining the compressibility of the foam. The hyperfoam model may not accurately fit Poisson's ratio if it varies significantly between compression and tension.

Model Prediction of Material Behavior Versus Experimental Data

Once the elastomeric foam constants are determined, the behavior of the hyperfoam model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the elastomeric foam constants determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. Single-element test cases can be used to calculate the nominal stress–nominal strain response of the material model.

See *Fitting of elastomeric foam test data*, which illustrates the entire process of fitting elastomeric foam constants to a set of test data.

Elastomeric Foam Material Stability

As with incompressible hyperelasticity, Abaqus checks the Drucker stability of the material for the deformation modes described above. The Drucker stability condition for a compressible material requires that the change in the Kirchhoff stress, $d\boldsymbol{\tau}$, following from an infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\tau} : d\boldsymbol{\varepsilon} > 0,$$

where the Kirchhoff stress $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. Using $d\boldsymbol{\tau} = \mathbf{D} : d\boldsymbol{\varepsilon}$, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0.$$

This restriction requires that the tangential material stiffness $\mathbf{D}$ be positive definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains

$$d\tau_1 d\varepsilon_1 + d\tau_2 d\varepsilon_2 + d\tau_3 d\varepsilon_3 > 0.$$

Thus, the relation between changes in the stress and changes in the strain can be obtained in the form of the matrix equation

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix},$$

where $D_{ij} = D_{ij}\left(\lambda_1, \lambda_2, \lambda_3\right)$.

Since $\mathbf{D}$ must be positive definite, it is necessary that

$$D_{11} + D_{22} + D_{33} > 0,$$
$$D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0,$$
$$\det\left(\mathbf{D}\right) > 0.$$

You should be careful about defining the parameters μ_i, α_i, and ν_i: especially when $N > 1$, the behavior at higher strains is strongly sensitive to the values of these parameters, and unstable material behavior may result if these values are not defined correctly. When some of the coefficients are strongly negative, instability at higher strain levels is likely to occur. Abaqus performs a check on the stability of the material for nine different forms of loading—uniaxial tension and compression, equibiaxial tension and compression, simple shear, planar tension and compression, and volumetric tension and compression—for $0.1 \leq \lambda_1 \leq 10.0$ (nominal strain range of $-0.9 \leq \epsilon_1 \leq 9.0$), at intervals $\Delta\lambda_1 = 0.01$. If an instability is found, Abaqus issues a warning message and prints the lowest absolute value of ϵ_1 for which the instability is observed. Ideally, no instability occurs. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you carefully examine and revise the material input data.

Improving the Accuracy and Stability of the Test Data Fit

Hyperelastic Behavior of Rubberlike Materials contains suggestions for improving the accuracy and stability of elastomeric modeling. *Fitting of elastomeric foam test data* illustrates the process of fitting elastomeric foam test data.

Elements

The hyperfoam model can be used with solid (continuum) elements, finite-strain shells (except S4), and membranes. However, it cannot be used with one-dimensional solid elements (trusses and beams), small-strain shells (STRI3, STRI65, S4R5, S8R, S8R5, S9R5), or the Eulerian elements (EC3D8R and EC3D8RT).

For continuum elements elastomeric foam hyperelasticity can be used with pure displacement formulation elements or, in Abaqus/Standard, with the “hybrid” (mixed formulation) elements. Since elastomeric foams are assumed to be very compressible, the use of hybrid elements will generally not yield any advantage over the use of purely displacement-based elements.

Procedures

The hyperfoam model must always be used with geometrically nonlinear analyses (*General and Perturbation Procedures*).

Anisotropic Hyperelastic Behavior

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*
- *Stress Softening in Rubberlike Materials*
- **ANISOTROPIC HYPERELASTIC*
- **VISCOELASTIC*
- **MULLINS EFFECT*
- *Creating an anisotropic hyperelastic material model*

Overview

The anisotropic hyperelastic material model:

- provides a general capability for modeling materials that exhibit highly anisotropic and nonlinear elastic behavior (such as biomedical soft tissues and fiber-reinforced elastomers);
- can be used in combination with large-strain time-domain viscoelasticity (*Time Domain Viscoelasticity*); however, viscoelasticity is isotropic;
- optionally allows the specification of energy dissipation and stress softening effects (see *Stress Softening in Rubberlike Materials*); and
- requires that geometric nonlinearity be accounted for during the analysis step (*General and Perturbation Procedures*) since it is intended for finite-strain applications.

Anisotropic Hyperelasticity Formulations

Many materials of industrial and technological interest exhibit anisotropic elastic behavior because of the presence of preferred directions in their microstructure. Examples of such materials include common engineering materials (such as fiber-reinforced composites, reinforced rubber, and wood) as well as soft biological tissues (arterial walls, heart tissue, etc.). When these materials are subjected to small deformations (less than 2–5%), their mechanical behavior can generally be modeled adequately using conventional anisotropic linear elasticity (see *Defining Fully Anisotropic Elasticity*). Under large deformations, however, these materials exhibit highly anisotropic and nonlinear elastic behavior because of rearrangements in the microstructure, such as reorientation of the fiber directions with deformation. The simulation of these nonlinear large-strain effects calls for more advanced constitutive models formulated within the framework of anisotropic hyperelasticity. Hyperelastic materials are described in terms of a "strain energy potential," U, which defines the strain energy stored in the material per unit of reference volume (volume in the initial configuration) as a function of the deformation at that point in the material. Two distinct formulations are used for the representation of the strain energy potential of anisotropic hyperelastic materials: strain-based and invariant-based.

Strain-Based Formulation

In this case the strain energy function is expressed directly in terms of the components of a suitable strain tensor, such as the Green strain tensor (see *Strain measures*):

$$U = U\left(\boldsymbol{\varepsilon}^G\right),$$

where $\boldsymbol{\varepsilon}^G = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right)$ is Green's strain; $\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F}$ is the right Cauchy-Green strain tensor; $\mathbf{F}$ is the deformation gradient; and $\mathbf{I}$ is the identity matrix. Without loss of generality, the strain energy function can be written in the form

$$U = U\left(\overline{\boldsymbol{\varepsilon}}^G, J^{el}\right),$$

where $\overline{\boldsymbol{\varepsilon}}^G = \frac{1}{2}\left(\overline{\mathbf{C}} - \mathbf{I}\right)$ is the modified Green strain tensor; $\overline{C} = J^{-\frac{2}{3}}\mathbf{C}$ is the distortional part of the right Cauchy-Green strain; $J = \det(\mathbf{F})$ is the total volume change; and J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*.

The underlying assumption in models based on the strain-based formulation is that the preferred material directions are initially aligned with an orthogonal coordinate system in the reference (stress-free) configuration. These directions might become nonorthogonal only after deformation. Examples of this form of strain energy function include the generalized Fung-type form described below.

Invariant-Based Formulation

Using the continuum theory of fiber-reinforced composites (Spencer, 1984) the strain energy function can be expressed directly in terms of the invariants of the deformation tensor and fiber directions. For example, consider a composite material that consists of an isotropic hyperelastic matrix reinforced with N families of fibers. The directions of the fibers in the reference configuration are characterized by a set of unit vectors $\mathbf{A}_\alpha$, ($\alpha = 1, \ldots, N$). Assuming that the strain energy depends not only on deformation, but also on the fiber directions, the following form is postulated

$$U = U\left(\mathbf{C}, \mathbf{A}_\alpha\right); \qquad \alpha = 1, \ldots, N.$$

The strain energy of the material must remain unchanged if both matrix and fibers in the reference configuration undergo a rigid body rotation. Then, following Spencer (1984), the strain energy can be expressed in terms of an irreducible set of scalar invariants that form the integrity basis of the tensor $\mathbf{C}$ and the vectors $\mathbf{A}_\alpha$:

$$U = U\left(\overline{I}_1, \overline{I}_2, J^{el}, \overline{I}_{4(\alpha\beta)}, \overline{I}_{5(\alpha\beta)}; \zeta_{\alpha\beta}\right); \qquad \alpha, \beta = 1, \ldots, N,$$

where $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants; J^{el} is the elastic volume ratio (or third strain invariant); $\overline{I}_{4(\alpha\beta)}$ and $\overline{I}_{5(\alpha\beta)}$ are the *pseudo-invariants* of $\overline{\mathbf{C}}$, $\mathbf{A}_\alpha$, and $\mathbf{A}_\beta$, defined as:

$$\overline{I}_{4(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}} \cdot \mathbf{A}_\beta, \qquad \text{and} \qquad \overline{I}_{5(\alpha\beta)} = \mathbf{A}_\alpha \cdot \overline{\mathbf{C}}^2 \cdot \mathbf{A}_\beta; \qquad \alpha = 1, \ldots, N; \quad \beta = 1, \ldots, \alpha.$$

The terms $\zeta_{\alpha\beta}$ are geometric constants (independent of deformation) equal to the cosine of the angle between the directions of any two families of fibers in the reference configuration:

$$\zeta_{\alpha\beta} = \mathbf{A}_\alpha \cdot \mathbf{A}_\beta; \qquad \alpha = 1, \ldots, N; \quad \beta = 1, \ldots, \alpha.$$

Unlike for the case of the strain-based formulation, in the invariant-based formulation the fiber directions need not be orthogonal in the initial configuration. An example of an invariant-based energy function is the form proposed by Holzapfel, Gasser, and Ogden (2000) for arterial walls (see *Holzapfel-Gasser-Ogden Form* below).

Anisotropic Strain Energy Potentials

There are four forms of strain energy potentials available in Abaqus to model approximately incompressible anisotropic materials:

- the generalized Fung form (including fully anisotropic and orthotropic cases),
- the Holzapfel-Gasser-Ogden form for arterial walls,
- the Holzapfel-Ogden form for passive myocardium tissue, and
- the Kaliske-Schmidt form for reinforced polymeric materials and biomaterials.

These forms are adequate for modeling soft biological tissue. However, whereas Fung's form is purely phenomenological, the other forms are micromechanically based.

In addition, Abaqus provides a general capability to support user-defined forms of the strain energy potential by means of two sets of user subroutines: one for strain-based and one for invariant-based formulations.

Generalized Fung Form

The generalized Fung strain energy potential has the following form:

$$U = \frac{c}{2}\left(\exp(Q) - 1\right) + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right),$$

where U is the strain energy per unit of reference volume; c and D are temperature-dependent material parameters; J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*; and Q is defined as

$$Q = \bar{\boldsymbol{\varepsilon}}^G : \mathbf{b} : \bar{\boldsymbol{\varepsilon}}^G = \bar{\varepsilon}^G_{ij} b_{ijkl} \bar{\varepsilon}^G_{kl},$$

where b_{ijkl} is a dimensionless symmetric fourth-order tensor of anisotropic material constants that can be temperature dependent and $\bar{\varepsilon}^G_{ij}$ are the components of the modified Green strain tensor.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = c\mathbf{b}\,, \qquad K_0 = \frac{2}{D}.$$

Abaqus supports two forms of the generalized Fung model: fully anisotropic and orthotropic. The number of independent components b_{ijlk} that must be specified depends on the level of anisotropy of the material: 21 for the fully anisotropic case and 9 for the orthotropic case.

Input File Usage: Use one of the following options:

**ANISOTROPIC HYPERELASTIC*, DEFINITION=FUNG-ANISOTROPIC
**ANISOTROPIC HYPERELASTIC*, DEFINITION=FUNG-ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**; **Material type: Anisotropic**; **Strain energy potential: Fung-Anisotropic** or **Fung-Orthotropic**

Holzapfel-Gasser-Ogden Form

The form of the strain energy potential is based on that proposed by Holzapfel, Gasser, and Ogden (2000) and Gasser, Ogden, and Holzapfel (2006) for modeling arterial layers with distributed collagen fiber orientations:

$$U = C_{10}\left(\bar{I}_1 - 3\right) + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right) + \frac{k_1}{2k_2}\sum_{\alpha=1}^{N}\left\{\exp\left[k_2\left\langle\bar{E}_\alpha\right\rangle^2\right] - 1\right\},$$

with

$$\overline{E}_\alpha \stackrel{\text{def}}{=} \kappa\left(\overline{I}_1 - 3\right) + (1 - 3\kappa)\,\left(\overline{I}_{4(\alpha\alpha)} - 1\right),$$

where U is the strain energy per unit of reference volume; C_{10}, D, k_1, k_2, and κ are temperature-dependent material parameters; N is the number of families of fibers ($N \le 3$); $\overline{I}_1$ is the first deviatoric strain invariant; J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*; and $\overline{I}_{4(\alpha\alpha)}$ are *pseudo-invariants* of $\overline{\mathbf{C}}$ and $\mathbf{A}_\alpha$.

The model assumes that the directions of the collagen fibers within each family are dispersed (with rotational symmetry) about a mean preferred direction. The parameter κ ($0 \le \kappa \le 1/3$) describes the level of dispersion in the fiber directions. If $\rho(\Theta)$ is the orientation density function that characterizes the distribution (it represents the normalized number of fibers with orientations in the range $[\Theta, \Theta + d\Theta]$ with respect to the mean direction), the parameter κ is defined as

$$\kappa = \tfrac{1}{4}\int_0^\pi \rho(\Theta)\,\sin^3\Theta d\Theta.$$

It is also assumed that all families of fibers have the same mechanical properties and the same dispersion. When $\kappa = 0$, the fibers are perfectly aligned (no dispersion). When $\kappa = 1/3$, the fibers are randomly distributed and the material becomes isotropic; this corresponds to a spherical orientation density function.

The strain-like quantity $\overline{E}_\alpha$ characterizes the deformation of the family of fibers with mean direction $\mathbf{A}_\alpha$. For perfectly aligned fibers ($\kappa = 0$), $\overline{E}_\alpha = \overline{I}_{4(\alpha\alpha)} - 1$; and for randomly distributed fibers ($\kappa = 1/3$), $\overline{E}_\alpha = \left(\overline{I}_1 - 3\right)/3$.

The first two terms in the expression of the strain energy function represent the distortional and volumetric contributions of the noncollagenous isotropic ground material, and the third term represents the contributions from the different families of collagen fibers, taking into account the effects of dispersion. A basic assumption of the model is that collagen fibers can support tension only, because they would buckle under compressive loading. Thus, the anisotropic contribution in the strain energy function appears only when the strain of the fibers is positive or, equivalently, when $\overline{E}_\alpha > 0$. This condition is enforced by the term $\langle \overline{E}_\alpha \rangle$, where the operator $\langle \cdot \rangle$ stands for the Macauley bracket and is defined as $\langle x \rangle = \frac{1}{2}(|x| + x)$.

See *Anisotropic hyperelastic modeling of arterial layers* for an example of an application of the Holzapfel-Gasser-Ogden energy potential to model arterial layers with distributed collagen fiber orientation.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = 4C_{10}\mathcal{I} + 2(1 - 3\kappa)^2 k_1 \sum_{\alpha=1}^{N} H\left(\overline{E}_\alpha\right)\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha\mathbf{A}_\alpha\,, \qquad K_0 = \tfrac{2}{D},$$

where $\mathcal{I}$ is the fourth-order unit tensor, and $H(x)$ is the Heaviside unit step function.

Input File Usage: **ANISOTROPIC HYPERELASTIC*, DEFINITION=HOLZAPFEL-GASSER-OGDEN, LOCAL DIRECTIONS=*N*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**; **Material type: Anisotropic**; **Strain energy potential: Holzapfel**, **Number of local directions:** *N*

Holzapfel-Ogden Form

The form of the strain energy potential is based on that proposed by *Holzapfel and Ogden (2009)* for modeling passive mechanical response of myocardium tissue, which is an orthotropic material:

$$U = \frac{a}{2b}\left\{\exp\left[b\left(\overline{I}_1 - 3\right)\right] - 1\right\} + \sum_{\alpha=f,s}\frac{a_\alpha}{2b_\alpha}\left\{\exp\left[b_\alpha\left\langle\overline{I}_{4(\alpha\alpha)} - 1\right\rangle^2\right] - 1\right\}$$

$$+\frac{a_{fs}}{2b_{fs}}\left\{\exp\left[b_{fs}\left(\overline{I}_{4(fs)} - \zeta_{12}\right)^2\right] - 1\right\} + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right),$$

where U is the strain energy per unit of reference volume; the indexes f and s ($f = 1$, $s = 2$) denote the fiber and sheet directions, which are assumed to be aligned with the local directions 1 and 2, respectively; a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}, and D are temperature-dependent material parameters; $\overline{I}_1$ is the first deviatoric strain invariant; J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*; and $\overline{I}_{4(11)}$, $\overline{I}_{4(22)}$, and $\overline{I}_{4(12)}$ are *pseudo-invariants* of $\overline{\mathbf{C}}$ and fiber family directions $\mathbf{A}_1$ and $\mathbf{A}_2$. $\zeta_{12} = \mathbf{A}_1 \cdot \mathbf{A}_2$ is a geometric constant. The number of fiber family directions, N, can be either 1 or 2. If $N = 1$, Abaqus ignores the third and fourth terms in the strain energy potential.

The operator $\{\cdot\}$ stands for the Macauley bracket and is defined as $\{x\} = \frac{1}{2}(|x| + x)$.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = 4a\mathcal{I} + 4a_f\mathbf{A}_1\mathbf{A}_1\mathbf{A}_1\mathbf{A}_1 + 4a_s\mathbf{A}_2\mathbf{A}_2\mathbf{A}_2\mathbf{A}_2 + a_{fs}\left(\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_2\mathbf{A}_1\right)\left(\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_2\mathbf{A}_1\right), \qquad K_0 = \frac{2}{D},$$

where $\mathcal{I}$ is the fourth-order unit tensor.

Input File Usage: **ANISOTROPIC HYPERELASTIC*, DEFINITION=HOLZAPFEL-OGDEN, LOCAL DIRECTIONS=*N*

Abaqus/CAE Usage: The Holzapfel-Ogden form is not supported in Abaqus/CAE.

Linearization of Holzapfel-Ogden Strain Energy Potential

If the material is stretched significantly, the second and third exponential terms in the Holzapfel-Ogden strain energy potential might overflow numerically. To avoid this, the exponential terms can be linearized/approximated by polynomial functions when the stretch in the fiber direction reaches a certain value λ_{fL}.

The linearization stretch in the fiber direction, λ_{fL}, directly determines the stretch value at which the second exponential term in the strain energy potential is linearized. The third exponential term in the strain energy potential is linearized at a value of the sheet stretch, λ_{sL}, that is calculated internally by equating the stiffness in the sheet direction (at λ_{sL}) with the corresponding stiffness in the fiber direction (at λ_{fL}). These stretches are written to the data (`.dat`) file when you request printout of the model definition data. The value of the linearization stretch must be $\lambda_{fL} \geq 1$. If you do not specify the linearization stretch, the exponential terms in the strain energy potential are not linearized.

Input File Usage: **ANISOTROPIC HYPERELASTIC*, DEFINITION=HOLZAPFEL-OGDEN, LINEARIZATION STRETCH=λ_{fL}

Abaqus/CAE Usage: The Holzapfel-Ogden form is not supported in Abaqus/CAE.

Kaliske-Schmidt Form

The form of the strain energy potential is based on that proposed by *Kaliske et al. (2004)* for modeling reinforced polymeric materials and biomaterials:

$$U = \sum_{i=1}^{3} a_i \left(\overline{I}_1 - 3\right)^i + \sum_{j=1}^{3} b_j \left(\overline{I}_2 - 3\right)^j + \sum_{k=2}^{6} c_k \left(\overline{I}_{4(11)} - 1\right)^k + \sum_{l=2}^{6} d_l \left(\overline{I}_{5(11)} - 1\right)^l + \sum_{m=2}^{6} e_m \left(\overline{I}_{4(22)} - 1\right)^m + \sum_{n=2}^{6} f_n \left(\overline{I}_{5(22)} -$$

where U is the strain energy per unit of reference volume; a_i, b_j, c_k, d_l, e_m, f_n, g_p ($i, j \in \{1, 2, 3\}$ and $k, l, m, n, p \in \{2, 3, 4, 5, 6\}$), and D are temperature-dependent material parameters; $\overline{I}_1$ and $\overline{I}_2$ are the first and second deviatoric strain invariants; J^{el} is the elastic volume ratio as defined below in *Thermal Expansion*; and $\overline{I}_{4(11)}$, $\overline{I}_{4(22)}$, $\overline{I}_{4(12)}$, $\overline{I}_{5(11)}$, and $\overline{I}_{5(22)}$ are *pseudo-invariants* of $\overline{\mathbf{C}}$ and fiber family directions $\mathbf{A}_1$ and $\mathbf{A}_2$. $\zeta_{12} = \mathbf{A}_1 \cdot \mathbf{A}_2$ is a geometric constant. The number of fiber family directions, N, can be either 1 or 2. If $N = 1$, the fifth, sixth, and seventh terms in the strain energy potential are ignored.

The initial deviatoric elasticity tensor, $\overline{\mathbf{D}}_0$, and bulk modulus, K_0, are given by

$$\overline{\mathbf{D}}_0 = 4\left(a_1 + b_1\right)\mathcal{I} + 8\left(c_2 + 4d_2\right)\mathbf{A}_1\mathbf{A}_1\mathbf{A}_1\mathbf{A}_1 + 8\left(e_2 + 4f_2\right)\mathbf{A}_2\mathbf{A}_2\mathbf{A}_2\mathbf{A}_2 + 2g_2\zeta_{12}^2\left(\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_2\mathbf{A}_1\right)\left(\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_2\mathbf{A}_1\right),$$

where $\mathcal{I}$ is the fourth-order unit tensor.

Input File Usage: **ANISOTROPIC HYPERELASTIC*, DEFINITION=KALISKE-SCHMIDT, LOCAL DIRECTIONS=*N*

Abaqus/CAE Usage: The Kaliske-Schmidt form is not supported in Abaqus/CAE.

User-Defined Form: Strain-Based

Alternatively, you can define the form of a strain-based strain energy potential directly with user subroutine *UANISOHYPER_STRAIN* in Abaqus/Standard or *VUANISOHYPER_STRAIN* in Abaqus/Explicit. The derivatives of the strain energy potential with respect to the components of the modified Green strain and the elastic volume ratio, J^{el}, must be provided directly through these user subroutines.

Either compressible or incompressible behavior can be specified in Abaqus/Standard; only nearly incompressible behavior is allowed in Abaqus/Explicit.

Optionally, you can specify the number of property values needed as data in the user subroutine as well as the number of solution-dependent variables (see *About User Subroutines and Utilities*).

Input File Usage: In Abaqus/Standard use the following option to define compressible behavior:

**ANISOTROPIC HYPERELASTIC*, DEFINITION=USER, FORMULATION=STRAIN, TYPE=COMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Standard use the following option to define incompressible behavior:

**ANISOTROPIC HYPERELASTIC*, DEFINITION=USER, FORMULATION=STRAIN, TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Explicit use the following option to define nearly incompressible behavior:

*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=STRAIN, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**; **Material type: Anisotropic**, **Strain energy potential: User**, **Formulation: Strain**, **Type: Incompressible** or **Compressible**, **Number of property values:** *n*

User-Defined Form: Invariant-Based

Alternatively, you can define the form of an invariant-based strain energy potential directly with user subroutine *UANISOHYPER_INV* in Abaqus/Standard or *VUANISOHYPER_INV* in Abaqus/Explicit. Either compressible or incompressible behavior can be specified in Abaqus/Standard; only nearly incompressible behavior is allowed in Abaqus/Explicit.

Optionally, you can specify the number of property values needed as data in the user subroutine and the number of solution-dependent variables (see *About User Subroutines and Utilities*).

The derivatives of the strain energy potential with respect to the strain invariants must be provided directly through user subroutine *UANISOHYPER_INV* in Abaqus/Standard and *VUANISOHYPER_INV* in Abaqus/Explicit.

Input File Usage: In Abaqus/Standard use the following option to define compressible behavior:

*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=INVARIANT, LOCAL DIRECTIONS=*N*,
TYPE=COMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Standard use the following option to define incompressible behavior:

*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=INVARIANT, LOCAL DIRECTIONS=*N*,
TYPE=INCOMPRESSIBLE, PROPERTIES=*n*

In Abaqus/Explicit use the following option to define nearly incompressible behavior:

*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=INVARIANT, PROPERTIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**; **Material type: Anisotropic**, **Strain energy potential: User**, **Formulation: Invariant**, **Type: Incompressible** or **Compressible**, **Number of local directions:** *N*, **Number of property values:** *n*

Definition of Preferred Material Directions

You must define the preferred material directions (or fiber directions) of the anisotropic hyperelastic material.

For strain-based forms (such as the Fung form and user-defined forms using user subroutines *UANISOHYPER_STRAIN* or *VUANISOHYPER_STRAIN*), you must specify a local orientation system (*Orientations*) to define the directions of anisotropy. Components of the modified Green strain tensor are calculated with respect to this system.

For invariant-based forms of the strain energy function (such as the Holzapfel-Gasser-Ogden form, the Holzapfel-Ogden form, the Kaliske-Schmidt form, and user-defined forms using user subroutines

UANISOHYPER_INV or *VUANISOHYPER_INV*), you must specify the local direction vectors, $\mathbf{A}_\alpha$, that

characterize each family of fibers. These vectors need not be orthogonal in the initial configuration. Up to three local directions can be specified as part of the definition of a local orientation system (*Defining a Local Coordinate System Directly*); the local directions are referred to this system.

In Abaqus/CAE, the local direction vectors of the material are orthogonal and align with the axes of the assigned material orientation. The best practice is to assign the orientation using discrete orientations in Abaqus/CAE.

Material directions can be output to the output database as described in *Output* below.

Compressibility

Most soft tissues and fiber-reinforced elastomers have very little compressibility compared to their shear flexibility. This behavior does not warrant special attention for plane stress, shell, or membrane elements, but the numerical solution can be quite sensitive to the degree of compressibility for three-dimensional solid, plane strain, and axisymmetric elements. In cases where the material is highly confined (such as an O-ring used as a seal), the compressibility must be modeled correctly to obtain accurate results. In applications where the material is not highly confined, the degree of compressibility is typically not crucial; for example, it would be quite satisfactory in Abaqus/Standard to assume that the material is fully incompressible: the volume of the material cannot change except for thermal expansion.

Compressibility in Abaqus/Standard

In Abaqus/Standard the use of "hybrid" (mixed formulation) elements is required for incompressible materials. In plane stress, shell, and membrane elements the material is free to deform in the thickness direction. In this case special treatment of the volumetric behavior is not necessary; the use of regular stress/displacement elements is satisfactory.

Compressibility in Abaqus/Explicit

With the exception of the plane stress and one-dimensional cases, it is impossible to assume that the material is fully incompressible in Abaqus/Explicit because the program has no mechanism for imposing such a constraint at each material calculation point. Instead, some compressibility must be modeled. The difficulty is that, in many cases, the actual material behavior provides too little compressibility for the algorithms to work efficiently. Thus, except for the plane stress case, you must provide enough compressibility for the code to work, knowing that this makes the bulk behavior of the model softer than that of the actual material. Failing to provide enough compressibility might introduce high frequency noise into the dynamic solution and require the use of excessively small time increments. Some judgment is, therefore, required to decide whether or not the solution is sufficiently accurate or whether the problem can be modeled at all with Abaqus/Explicit because of this numerical limitation.

If no value is given for the material compressibility of the anisotropic hyperelastic model, by default

Abaqus/Explicit assumes the value $K_0/\mu_0 = 20$, where μ_0 is the largest value of the initial shear modulus (among the different material directions). The exception is for the case of user-defined forms, where some compressibility must be defined directly within user subroutine *UANISOHYPER_INV* or *VUANISOHYPER_INV*.

Thermal Expansion

Both isotropic and orthotropic thermal expansion is permitted with the anisotropic hyperelastic material model.

The elastic volume ratio, J^{el}, relates the total volume ratio, J, and the thermal volume ratio, J^{th}:

$$J^{el} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = \left(1+\varepsilon_1^{th}\right)\left(1+\varepsilon_2^{th}\right)\left(1+\varepsilon_3^{th}\right),$$

where ε_i^{th} are the principal thermal expansion strains that are obtained from the temperature and the thermal expansion coefficients (*Thermal Expansion*).

Viscoelasticity

Anisotropic hyperelastic models can be used in combination with isotropic viscoelasticity to model rate-dependent material behavior (*Time Domain Viscoelasticity*). Because of the isotropy of viscoelasticity, the relaxation function is independent of the loading direction. This assumption might not be acceptable for modeling materials that exhibit strong anisotropy in their rate-dependent behavior; therefore, this option should be used with caution.

The anisotropic hyperelastic response of rate-dependent materials (*Time Domain Viscoelasticity*) can be specified by defining either the instantaneous response or the long-term response of such materials.

Input File Usage: Use either of the following options:

**ANISOTROPIC HYPERELASTIC*, MODULI=INSTANTANEOUS
**ANISOTROPIC HYPERELASTIC*, MODULI=LONG TERM

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**; **Material type: Anisotropic**; **Moduli: Long term** or **Instantaneous**

Stress Softening

The response of typical anisotropic hyperelastic materials, such as reinforced rubbers and biological tissues, under cyclic loading and unloading usually displays stress softening effects during the first few cycles. After a few cycles, the response of the material tends to stabilize and the material is said to be *preconditioned*. Stress softening effects, often referred to in the elastomers literature as Mullins effect, can be accounted for using the anisotropic hyperelastic model in combination with the *pseudo-elasticity* model for Mullins effect in Abaqus (see *Stress Softening in Rubberlike Materials*). The stress softening effects provided by this model are isotropic.

Elements

The anisotropic hyperelastic material model can be used with solid (continuum) elements, finite-strain shells (except S4), continuum shells, and membranes. When used in combination with elements with plane stress formulations, Abaqus assumes fully incompressible behavior and ignores any amount of compressibility specified for the material.

The invariant-based anisotropic hyperelastic material model is also available with one-dimensional elements (trusses and rebars) in Abaqus/Explicit. In this case, Abaqus/Explicit assumes fully incompressible material behavior.

Pure Displacement Formulation Versus Hybrid Formulation in Abaqus/Standard

For continuum elements in Abaqus/Standard anisotropic hyperelasticity can be used with the pure displacement formulation elements or with the “hybrid” (mixed formulation) elements. Pure displacement formulation elements must be used with compressible materials, and “hybrid” (mixed formulation) elements must be used with incompressible materials.

In general, an analysis using a single hybrid element is only slightly more computationally expensive than an analysis using a regular displacement-based element. However, when the wavefront is optimized, the Lagrange multipliers might not be ordered independently of the regular degrees of freedom associated with the element. Thus, the wavefront of a very large mesh of second-order hybrid tetrahedra might be noticeably larger than that

of an equivalent mesh using regular second-order tetrahedra. This might lead to significantly higher CPU costs, disk space, and memory requirements.

Incompatible Mode Elements in Abaqus/Standard

Incompatible mode elements should be used with caution in applications involving large strains. Convergence might be slow, and in hyperelastic applications inaccuracies might accumulate. Erroneous stresses might sometimes appear in incompatible mode anisotropic hyperelastic elements that are unloaded after having been subjected to a complex deformation history.

Procedures

Anisotropic hyperelasticity must always be used with geometrically nonlinear analyses (*General and Perturbation Procedures*).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), local material directions are output whenever element field output is requested to the output database. The local directions are output as field variables (LOCALDIR1, LOCALDIR2, LOCALDIR3) representing the direction cosines; these variables can be visualized as vector plots in the Visualization module of Abaqus/CAE (Abaqus/Viewer).

Output of local material directions is suppressed if no element field output is requested or if you specify not to have element material directions written to the output database (see *Specifying the Directions for Element Output*).

References

- Gasser, T. C., R. W. Ogden, and G. A. Holzapfel, "Hyperelastic Modelling of Arterial Layers with Distributed Collagen Fibre Orientations," *Journal of the Royal Society Interface*, vol. 3, pp. 15–35, 2006.
- Holzapfel, G. A., T. C. Gasser, and R. W. Ogden, "A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models," *Journal of Elasticity*, vol. 61, pp. 1–48, 2000.
- Holzapfel, G. A., and R. W. Ogden, "A Constitutive Modelling of Passive Myocardium: a Structurally based Framework for Material Characterization," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, pp. 3445–3475, 2009.
- Kaliske, M., J. Schmidt, G. Lin, and G. Bhashyam, "Implementation of Nonlinear Anisotropic Elasticity at Finite Strains into ANSYS® Including Viscoelasticity and Damage," *23nd CAD-FEM Users' Meeting 2005 International Congress on FEM Technology with ANSYS CFX & ICEM CFD Conference*, 2005.
- Spencer, A. J. M., "Constitutive Theory for Strongly Anisotropic Solids," *A. J. M. Spencer (ed.), Continuum Theory of the Mechanics of Fibre-Reinforced Composites, CISM Courses and Lectures No. 282, International Centre for Mechanical Sciences, Springer-Verlag, Wien*, pp. 1–32, 1984.

Stress Softening in Elastomers

Abaqus provides material models for modeling stress softening of filled rubber elastomers under quasi-static cyclic loading and for including permanent energy dissipation and stress softening effects in elastomeric foams.

In this section:

- *Stress Softening in Rubberlike Materials*
- *Stress Softening in Elastomeric Foams*

Stress Softening in Rubberlike Materials

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Combining Material Behaviors*
- *Elastic Behavior*
- *Hyperelastic Behavior of Rubberlike Materials*
- *Anisotropic Hyperelastic Behavior*
- *Permanent Set in Rubberlike Materials*
- *Stress Softening in Elastomeric Foams*
- **HYPERELASTIC*
- **MULLINS EFFECT*
- **PLASTIC*
- **UNIAXIAL TEST DATA*
- **BIAXIAL TEST DATA*
- **PLANAR TEST DATA*
- *Mullins effect*

Overview

The Mullins effect functionality:

- can be used for modeling stress softening of filled rubber elastomers under quasi-static cyclic loading, a phenomenon referred to in the literature as Mullins effect;
- can be used for modeling amplitude-dependent softening under dynamic cyclic loading;
- provides an extension to the well-known isotropic hyperelastic models;
- is based on the theory of incompressible isotropic elasticity modified by the addition of a single variable, referred to as the damage variable;
- assumes that only the deviatoric part of the material response is associated with damage;
- is intended for modeling material response in situations where different parts of the model undergo different levels of damage resulting in a different material response;
- is applied to the long-term modulus when combined with viscoelasticity; and
- cannot be used with hysteresis.

Abaqus provides a similar capability that can be applied to elastomeric foams (see *Stress Softening in Elastomeric Foams*).

Material Behavior

Elastomeric materials exhibit highly nonlinear mechanical behavior, which is often described using a combination of hyperelastic and viscoelastic materials. However, filled rubber exhibits additional softening behaviors that cannot be captured accurately by these models. For example, when a rubber specimen is loaded, unloaded, and then reloaded, the load required to produce a given stretch during reloading is smaller than that required to

produce the same stretch during the primary loading phase. This behavior is known as Mullins effect. Another type of softening behavior commonly observed in rubber, particularly filled rubber, is amplitude-dependent softening under cyclic loading. Modeling the amplitude-dependent softening allows you to accurately predict the decrease of storage modulus with increased dynamic strain, a phenomenon that is known as Payne or Fletcher-Gent effect.

Abaqus provides functionality that can capture both types of softening behavior. The Ogden-Roxburgh model is particularly suitable for modeling the Mullins effect phenomenon, whereas the dynamic stiffness decrease can be predicted using the models with the exponential and slow decay forms of evolution of the damage variables. These models can also be effective at modeling material damage in a more general sense. Finally, you can also define other forms of the damage or softening behavior with dedicated user subroutines (*UMULLINS* and *VUMULLINS*). The different models are presented in detail in the following sections.

Ogden-Roxburgh Model of Mullins Effect

The real behavior of filled rubber elastomers under cyclic loading conditions is quite complex. Certain idealizations have been made for modeling purposes. In particular, the Ogden-Roxburgh model assumes two main components to the material behavior: the first component describes the response of a material point (from an undeformed state) under monotonic straining, and the second component is associated with damage and describes the unloading-reloading behavior.

Idealized Material Behavior

When an elastomeric test specimen is subjected to simple tension from its virgin state, unloaded, and then reloaded, the stress required on reloading is less than that on the initial loading for stretches up to the maximum stretch achieved during the initial loading. This stress softening phenomenon is known as the Mullins effect and reflects damage incurred during previous loading. This type of material response is depicted qualitatively in *Figure 1*.

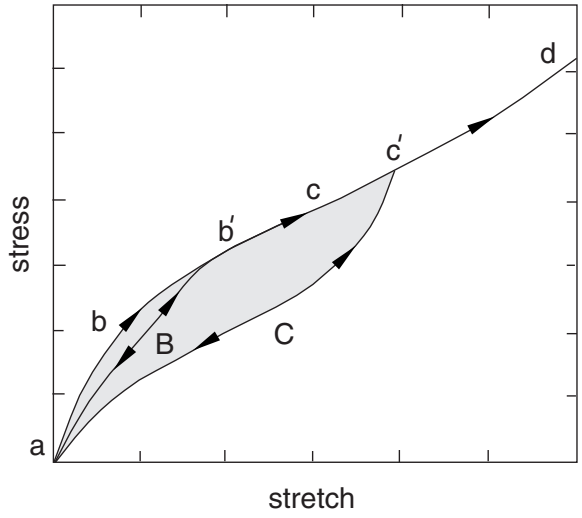


Figure 1: Idealized response of the Mullins effect model.

This figure and the accompanying description are based on work by Ogden and Roxburgh (1999), which forms the basis of the model implemented in Abaqus. Consider the primary loading path abb' of a previously unstressed material, with loading to an arbitrary point b'.

1. Upon unloading from b', the path $b'Ba$ is followed.
2. When the material is loaded again, the softened path is retraced as aBb'.

3. If further loading is subsequently applied, the path $b'c$ is followed, where $b'c$ is a continuation of the primary loading path $abb'cc'd$ (which is the path that would be followed if there was no unloading).
4. If loading is now stopped at c', the path $c'Ca$ is followed on unloading and then retraced back to c' on reloading.
5. If no further loading beyond c' is applied, the curve aCc' represents the subsequent material response, which is then elastic.
6. For loading beyond c', the primary path is again followed and the pattern described is repeated.

This is an ideal representation of Mullins effect because in practice there is some permanent set on unloading and/or viscoelastic effects such as hysteresis. Points such as b' and c' might not exist in reality in the sense that unloading from the primary curve followed by reloading to the maximum strain level attained earlier usually results in a stress that is somewhat lower than the stress corresponding to the primary curve. In addition, the cyclic response for some filled elastomers shows evidence of progressive damage during unloading from and subsequent reloading to a certain maximum strain level. Such progressive damage usually occurs during the first few cycles, and the material behavior soon stabilizes to a loading/unloading cycle that is followed beyond the first few cycles. More details regarding the actual behavior and how test data that display such behavior can be used to calibrate the Abaqus model for Mullins effect are discussed later and in *Analysis of a solid disc with Mullins effect and permanent set*.

We refer to the loading path $abb'cc'd$ as the "primary hyperelastic behavior," which is defined using a hyperelastic material model.

Stress softening is interpreted as being due to damage at the microscopic level. As the material is loaded, the damage occurs by the severing of bonds between filler particles and the rubber molecular chains. Different chain links break at different deformation levels, thereby leading to continuous damage with macroscopic deformation. An equivalent interpretation is that the energy required to cause the damage is not recoverable.

Primary Hyperelastic Behavior

Hyperelastic materials are described in terms of a "strain energy potential" function $U(\mathbf{F})$ that defines the strain energy stored in the material per unit reference volume (volume in the initial configuration). The quantity $\mathbf{F}$ is the deformation gradient tensor. To account for Mullins effect, Ogden and Roxburgh propose a material description that is based on an energy function of the form $U(\mathbf{F}, \eta)$, where the additional scalar variable, η, represents damage in the material. The damage variable controls the material properties in the sense that it enables the material response to be governed by an energy function on unloading and subsequent submaximal reloading different from that on the primary (initial) loading path from a virgin state. Because of the above interpretation of η, it is no longer appropriate to think of U as the stored elastic energy potential. Part of the energy is stored as strain energy, while the rest is dissipated due to damage. The shaded area in *Figure 1* represents the energy dissipated by damage because of deformation until the point c', while the unshaded part represents the recoverable strain energy.

The following paragraphs provide a summary of the Mullins effect model in Abaqus. For further details, see *Mullins effect*. In preparation for writing the constitutive equations for Mullins effect, it is useful to separate the deviatoric and the volumetric parts of the total strain energy density as

$$U = U_{dev} + U_{vol}.$$

In the above equation U, U_{dev}, and U_{vol} are the total, deviatoric, and volumetric parts of the strain energy density, respectively. All the hyperelasticity models in Abaqus use strain energy potential functions that are already separated into deviatoric and volumetric parts. For example, the polynomial models use a strain energy potential of the form

$$U = \sum_{i+j=1}^{N} C_{ij}\left(\bar{I}_1 - 3\right)^i \left(\bar{I}_2 - 3\right)^j + \sum_{i=1}^{N} \frac{1}{D_i}\left(J^{el} - 1\right)^{2i},$$

where the symbols have the usual interpretations. The first term on the right represents the deviatoric part of the elastic strain energy density function, and the second term represents the volumetric part.

Modified Strain Energy Density Function

The Mullins effect is accounted for by using an augmented energy function of the form

$$U\left(\bar{\lambda}_i, \eta\right) = \eta \tilde{U}_{dev}\left(\bar{\lambda}_i\right) + \phi\left(\eta\right) + \tilde{U}_{vol}\left(J^{el}\right),$$

where $\tilde{U}_{dev}\left(\bar{\lambda}_i\right)$ is the deviatoric part of the strain energy density of the primary hyperelastic behavior, defined, for example, by the first term on the right-hand side of the polynomial strain energy function given above; $\tilde{U}_{vol}\left(J^{el}\right)$ is the volumetric part of the strain energy density, defined, for example, by the second term on the right-hand-side of the polynomial strain energy function given above; $\bar{\lambda}_i (i = 1, 2)$ represent the deviatoric principal stretches; and J^{el} represents the elastic volume ratio. The function $\phi\left(\eta\right)$ is a continuous function of the damage variable η and is referred to as the "damage function." When the deformation state of the material is on a point on the curve that represents the primary hyperelastic behavior, $\eta = 1$, $\phi\left(\eta\right) = 0$, $U\left(\bar{\lambda}_i, 1\right) = \tilde{U}_{dev}\left(\bar{\lambda}_i\right) + \tilde{U}_{vol}\left(J^{el}\right)$, and the augmented energy function reduces to the strain energy density function of the primary hyperelastic behavior. The damage variable varies continuously during the course of the deformation and always satisfies $0 < \eta \leq 1$. The above form of the energy function is an extension of the form proposed by Ogden and Roxburgh to account for material compressibility.

Stress Computation

With the above modification to the energy function, the stresses are given by

$$\boldsymbol{\sigma}\left(\eta, \bar{\lambda}_i, J^{el}\right) = \eta \tilde{\mathbf{S}}\left(\bar{\lambda}_i\right) - \tilde{p}\left(J^{el}\right)\mathbf{I},$$

where $\tilde{\mathbf{S}}$ is the deviatoric stress corresponding to the primary hyperelastic behavior at the current deviatoric deformation level $\bar{\lambda}_i$ and $\tilde{p}$ is the hydrostatic pressure of the primary hyperelastic behavior at the current volumetric deformation level J^{el}. Therefore, the deviatoric stress as a result of Mullins effect is obtained by scaling the deviatoric stress of the primary hyperelastic behavior with the damage variable η. The pressure stress is the same as that of the primary behavior. The model predicts loading/unloading along a single curve (that is different, in general, from the primary hyperelastic behavior) from any given strain level that passes through the origin of the stress-strain plot. It cannot capture permanent strains on removal of the load. The model also predicts that a purely volumetric deformation will not have any damage or Mullins effect associated with it.

Damage Variable

The damage variable, η, varies with the deformation according to

$$\eta = 1 - \frac{1}{r}\,\mathrm{erf}\left(\frac{U_{dev}^m - \tilde{U}_{dev}}{m + \beta U_{dev}^m}\right),$$

where U_{dev}^m is the maximum value of $\tilde{U}_{dev}$ at a material point during its deformation history; r, β, and m are material parameters; and $\mathrm{erf}\left(x\right)$ is the error function defined as

$$\text{erf}\,(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-w^2\right) dw.$$

When $\widetilde{U}_{dev} = U^m_{dev}$, corresponding to a point on the primary curve, $\eta = 1.0$. However, η attains its minimum value, η_m, given by

$$\eta_m = 1 - \frac{1}{r}\,\text{erf}\left(\frac{U^m_{dev}}{m + \beta U^m_{dev}}\right),$$

on removal of deformation, when $\widetilde{U}_{dev} = 0$. For all intermediate values of $\widetilde{U}_{dev}$, η varies monotonically between 1.0 and η_m. While the parameters r and β are dimensionless, the parameter m has units of energy density (that is, energy per unit volume), which is equivalent to stress units. The equation for η reduces to that proposed by Ogden and Roxburgh when $\beta = 0$. The material parameters can be specified directly or can be computed by Abaqus based on curve-fitting of unloading-reloading test data. These parameters are subject to the restrictions $r > 1$, $\beta \geq 0$, and $m \geq 0$ (the parameters β and m cannot both be zero). Alternatively, the damage variable η can be defined through user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit.

If the parameter $\beta = 0$ and the parameter m has a value that is small compared to U^m_{dev}, the slope of the stress-strain curve at the initiation of unloading from relatively large strain levels might become very high. As a result, the response might become discontinuous, as illustrated in *Figure 2*.

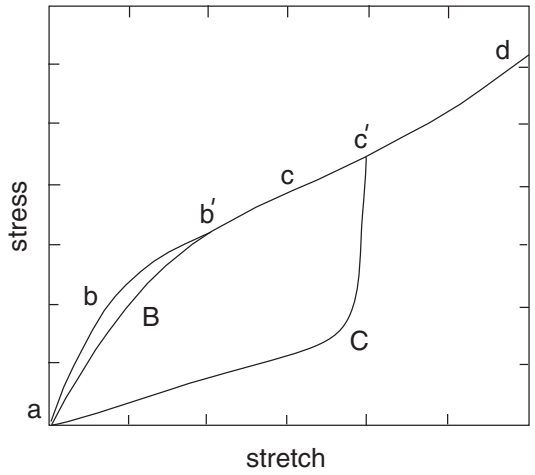


Figure 2: Overly stiff response at the initiation of unloading.

This kind of behavior might lead to convergence problems in Abaqus/Standard. In Abaqus/Explicit the high stiffness leads to very small stable time increments, thereby leading to a degradation in performance. You can avoid this problem by choosing a small value for β. The choice $\beta = 0$ can be used to define the original Ogden-Roxburgh model. The default value of β is 0 in Abaqus/Standard and 0.1 in Abaqus/Explicit.

The parameters r, β, and m do not have direct physical interpretations in general. The parameter m controls whether damage occurs at low strain levels. If $m = 0$, there is a significant amount of damage at low strain levels. On the other hand, a nonzero m leads to little or no damage at low strain levels. For further discussion regarding the implications of this model to the energy dissipation, see *Mullins effect*. The qualitative effects of varying the parameters r and β individually, while holding the other parameters fixed, are shown in *Figure 3*.

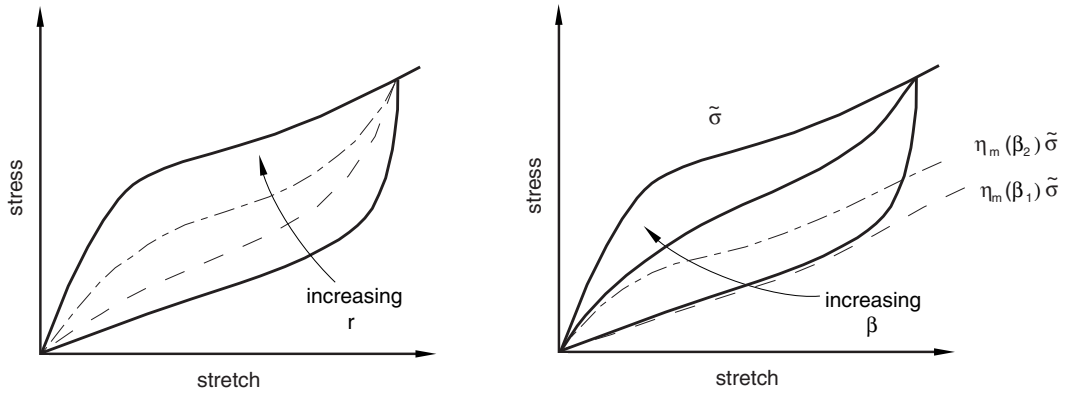


Figure 3: Qualitative dependence of damage on material properties.

The figure on the left shows the unloading-reloading curve from a certain maximum strain level for increasing values of r. It suggests that the parameter r controls the amount of damage, with decreasing damage for increasing r. This behavior follows from the fact that the larger the value of r, the less the damage variable η can deviate from unity. The figure on the right shows the unloading-reloading curve from a certain maximum strain level for increasing values of β. It suggests that increasing β also leads to lower amounts of damage. The figure also shows that the unloading-reloading response approaches the asymptotic response given by $\eta_m\tilde{\sigma}$, where η_m is the minimum value of η, faster for lower values of β. The dashed curves represent the asymptotic response at two different values of β (β_1 and β_2). For fixed values of r and m, η_m is a function of β. In particular, if $m = 0$,

$$\eta_m = 1 - \frac{1}{r}\,\mathrm{erf}\left(\frac{1}{\beta}\right).$$

The above relation is approximately true if U_{dev}^{m} is much greater than m.

Damage Energy

The damage energy dissipation, represented by the shaded area in *Figure 1* for deformation until $\boldsymbol{c}'$, is computed as follows:

- When the damaged material is in a fully unloaded state, the augmented energy function has the residual value $U(\mathbf{I}, \eta_m) = \phi(\eta_m)$.
- The residual value of the energy function upon complete unloading represents the energy dissipated because of damage in the material.
- The recoverable part of the energy is obtained by subtracting the dissipated energy from the augmented energy as $\eta\tilde{U}_{dev}(\bar{\lambda}_i) + \phi(\eta) + \tilde{U}_{vol} - \phi(\eta_m)$.

The damage energy accumulates with progressive deformation along the primary loading curve and remains constant during unloading. During unloading, the recoverable part of the strain energy is released, and it becomes zero when the material point is completely unloaded. Upon further reloading from a completely unloaded state, the recoverable part of the strain energy increases from zero. When the maximum strain that was attained earlier is exceeded upon reloading, further accumulation of damage energy occurs.

Specifying the Mullins Effect Material Model in Abaqus

The primary hyperelastic behavior is defined using the hyperelastic material model (see *Hyperelastic Behavior of Rubberlike Materials*). The Mullins effect model can be defined by specifying the Mullins effect parameters directly or by using test data to calibrate the parameters.

Specifying the Parameters Directly

The parameters r, m, and β of the Mullins effect can be given directly as functions of temperature and/or field variables.

Input File Usage: **MULLINS EFFECT*, DEFINITION=OGDEN-ROXBURGH

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **Constants**

Using Test Data to Calibrate the Parameters

Experimental unloading-reloading data from different strain levels can be specified for up to three simple tests: uniaxial, biaxial, and planar. Abaqus then computes the material parameters using a nonlinear least-squares curve fitting algorithm. It is generally best to obtain data from several experiments involving different kinds of deformation over the range of strains of interest in the actual application and to use all these data to determine the parameters. It is also important to obtain a good curve-fit for the primary hyperelastic behavior if the primary behavior is defined using test data.

By default, Abaqus attempts to fit all three parameters to the given data. This is possible in general, except in the situation when the test data correspond to unloading-reloading from only a single value of U_{dev}^{m}. In this case the parameters m and β cannot be determined independently; one of them must be specified. If you specify neither m nor β, Abaqus needs to assume a default value for one of these parameters. In light of the potential problems discussed earlier with $\beta = 0$, Abaqus assumes that $m = 0$ in the above situation. The curve-fitting can also be carried out by specifying any one or two of the material parameters to be fixed, predetermined values.

As many data points as required can be entered from each test. It is recommended that data from all three tests (on samples taken from the same piece of material) be included and that the data points cover unloading/reloading from/to the range of nominal strain expected to arise in the actual loading.

The strain data should be given as nominal strain values (change in length per unit of original length). The stress data should be given as nominal stress values (force per unit of original cross-sectional area). These tests allow for entering both compression and tension data. Compressive stresses and strains are entered as negative values.

For each set of test input, the data point with the maximum nominal strain identifies the point of unloading. This point is used by the curve-fitting algorithm to compute U_{dev}^{m} for that curve.

Figure 4 shows some typical unloading-reloading data from three different strain levels.

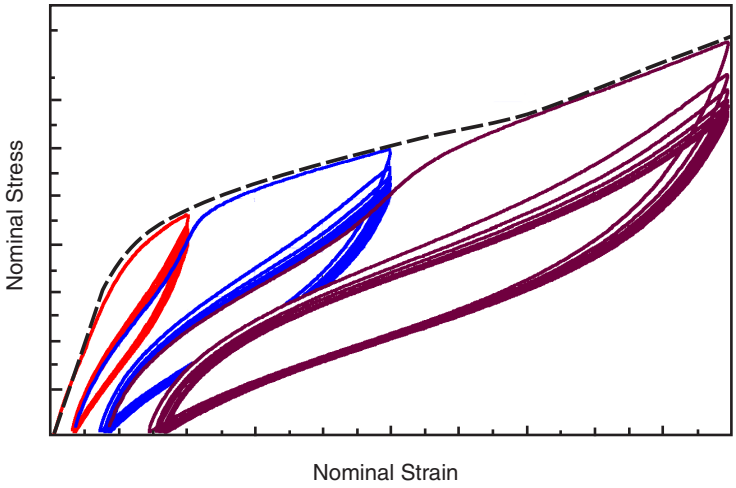


Figure 4: Typical available test data for Mullins effect.

The data include multiple loading and unloading cycles from each strain level. As *Figure 4* indicates, the loading/unloading cycles from any given strain level do not occur along a single curve, and there is some amount of hysteresis. There is also some amount of permanent set on removal of the applied load. The data also show evidence of progressive damage with repeated cycling at any given maximum strain level. The response appears to stabilize after a number of cycles. When such data are used to calibrate the Mullins effect model, the resulting response will capture the overall stiffness characteristics, while ignoring effects such as hysteresis, permanent set, or progressive damage. The above data can be provided to Abaqus in the following manner:

- The primary curve can be made up of the data points indicated by the dashed curve in *Figure 4*. Essentially, this consists of an envelope of the first loading curves to the different strain levels.
- You can specify the unloading-reloading curves from the three different strain levels by providing the data points as is; that is, as the repeated unloading-reloading cycles shown in *Figure 4*. As discussed earlier, you need to provide the data from the different strain levels as different tables so that they can be distinguished . For example, assuming that the test data correspond to the uniaxial tension state, three tables of uniaxial test data are required for the three different strain levels shown in *Figure 4*. In this case Abaqus will provide a best fit using all the data points (from all strain levels). The resulting fit would result in a response that is an average of all the test data at any given strain level. While permanent set can be modeled (see *Permanent Set in Rubberlike Materials*), hysteresis will be lost in the process.
- Alternatively, you can provide any one unloading-reloading cycle from each different strain level. If the component is expected to undergo repeated cyclic loading, the latter might be, for example, the stabilized cycle at each strain level. On the other hand, if the component is expected to undergo predominantly monotonic loading with perhaps small amounts of unloading, the very first unloading curve at each strain level might be the appropriate input data for calibrating the Mullins coefficients.

Once the Mullins effect constants are determined, the behavior of the Mullins effect model in Abaqus is established. However, the quality of this behavior must be assessed: the prediction of material behavior under different deformation modes must be compared against the experimental data. You must judge whether the Mullins effect constants determined by Abaqus are acceptable, based on the correlation between the Abaqus predictions and the experimental data. Single-element test cases can be used to derive the nominal stress–nominal strain response of the material model.

The steps that can be taken for improving the quality of the fit for the Mullins effect parameters are similar in essence to the guidelines provided for curve fitting the primary hyperelastic behavior (see *Hyperelastic Behavior of Rubberlike Materials* for details). In addition, the quality of the fit for the Mullins effect parameters depends on a good fit for the primary hyperelastic behavior, if the primary behavior is defined using test data.

You can evaluate the quality of the fit by carrying out a numerical experiment with a single element that is loaded in the same mode for which test data has been provided. Alternatively, you can obtain the numerical response for both the primary and the softening behavior by requesting model definition data output (see *About Output*) and carrying out a data check analysis. The response computed by Abaqus is printed in the data (`.dat`) file along with the experimental data. This tabular data can be plotted in Abaqus/CAE for comparison and evaluation purposes. The primary hyperelastic behavior can also be evaluated with the automated material evaluation tools in Abaqus/CAE.

Input File Usage: **MULLINS EFFECT*, DEFINITION=OGDEN-ROXBURGH, TEST DATA INPUT, BETA *and/or* M *and/or* R

In addition, use at least one and up to three of the following options to give the unloading-reloading test data (see *Experimental Tests*):

**UNIAXIAL TEST DATA*
**BIAXIAL TEST DATA*
**PLANAR TEST DATA*

You can enter multiple unloading-reloading curves from different strain levels for any given test type by repeating the specification of the appropriate test data option.

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **Test Data Input**: enter the values for up to two of the values **r**, **m**, and **beta**. In addition, select and enter data for at least one of the following: **Add Test->Biaxial Test**, **Planar Test**, or **Uniaxial Test**

Exponential and Slow Decay Softening

The exponential and slow decay softening 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.

Figure 5 graphically depicts the material response and the evolution of the damage variable. The top dashed line shows the hyperelastic response of the undamaged material, the bottom dashed line shows the response of the material that reached the maximum degradation, and the solid line shows the actual response of the material, which includes softening. As shown:

- When the material is loaded (path OAB) the damage variable evolves and the stress response deviates from that of the undamaged material.
- If the specimen is subsequently unloaded and reloaded along the path OCB, the damage variable will remain constant and the material response is hyperelastic until the previous maximum value of stretch is reached (point B).
- From this point, if the stretches increase, the damage variable will further evolve and softening will take place (path BD).
- Damage continues until the maximum specified value ($D_{\max} = 1 - \eta_F$) is reached.

Therefore, the deviatoric stress-strain response is always bounded by two curves: the undamaged curve, $\tilde{S}\left(\bar{\lambda}_i\right)$, and the maximum damage curve, $\eta_F \tilde{S}\left(\bar{\lambda}_i\right)$.

This behavior is different from the approach used by the Ogden-Roxburgh model, where the material always follows the specified hyperelastic response under monotonic loading, and the damage variable changes in value during unloading and reloading.

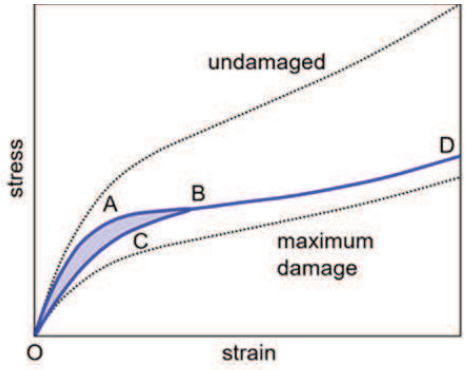


Figure 5: Response of material with slow decay form of damage variable evolution

Strain Energy Density Function

The strain energy is split into the deviatoric and volumetric parts in a way that is similar to what is done for the Ogden-Roxburgh model, and softening affects only the deviatoric response. The strain energy function has the following form

$$U\left(\overline{\lambda}_i,\eta\right)=\eta\widetilde{U}_{dev}\left(\overline{\lambda}_i\right)+\widetilde{U}_{vol}\left(J^{el}\right),$$

where $\widetilde{U}_{dev}\left(\overline{\lambda}_i\right)$ is the deviatoric part of the strain energy density of the undamaged hyperelastic behavior, $\widetilde{U}_{vol}\left(J^{el}\right)$ is the volumetric part of the strain energy density, $\overline{\lambda}_i(i=1,2)$ represent the deviatoric principal stretches; and J^{el} represents the elastic volume ratio.

The stresses are obtained from the following formula

$$\boldsymbol{\sigma}\left(\eta,\overline{\lambda}_i,J^{el}\right)=\eta\widetilde{\mathbf{S}}\left(\overline{\lambda}_i\right)-\tilde{p}\left(J^{el}\right)\mathbf{I},$$

where $\widetilde{\mathbf{S}}$ is the deviatoric stress corresponding to the undamaged hyperelastic behavior at the current deviatoric deformation level $\overline{\lambda}_i$, and $\tilde{p}$ is the hydrostatic pressure of the undamaged hyperelastic behavior at the current volumetric deformation level J^{el}.

Damage Variable

To complete the description of the model, the damage variable must be specified. Two forms are available to describe the evolution of the damage variable: the exponential form and the slow decay form. Both describe the evolution of damage as a monotonic function of maximum strain energy, $\eta\left(U_{dev}^{m}\right)$.

The exponential form is given by

$$\eta\left(U_{dev}^{m}\right)=\eta_F+\left(1-\eta_F\right)e^{-\left(\frac{U_{dev}^{m}}{U_h}\right)^{\alpha}},$$

where U_{dev}^{m} is the maximum value of $\widetilde{U}_{dev}$ at a material point during its deformation history; η_F, U_h and α are material parameters, and η_F represents the minimum value of η, corresponding to the maximum value of damage ($D_{\max}=1-\eta_F$).

The slow decay form is given by

$$\eta\left(U_{dev}^{m}\right)=\eta_{F}+\left(1-\eta_{F}\right)\left(1-e^{-\left(\frac{U_{dev}^{m}}{U_{h}}\right)^{\alpha}}\right)/\left(\frac{U_{dev}^{m}}{U_{h}}\right)^{\alpha}$$

where U_{dev}^{m} is the maximum value of $\tilde{U}_{dev}$ at a material point during its deformation history; η_F, U_h and α are material parameters similar to those in the exponential form.

Damage Energy

The damage energy dissipation, represented by the shaded area in *Figure 5* for deformation until B, is computed by incrementally integrating $\phi'(\eta)=-\tilde{U}_{dev}\left(\bar{\lambda}_i\right)$ with $\phi(1)=0$.

The recoverable part of the energy is given by $\eta\tilde{U}_{dev}\left(\bar{\lambda}_i\right)+\tilde{U}_{vol}$ and is represented by the unshaded area below the curve OCB in *Figure 5* for deformation until B.

The damage energy accumulates with progressive deformation along the primary loading curve and remains constant during unloading. During unloading, the recoverable part of the strain energy is released, and it becomes zero when the material point is completely unloaded. Upon further reloading from a completely unloaded state, the recoverable part of the strain energy increases from zero. When the maximum strain that was attained earlier is exceeded upon reloading, further accumulation of damage energy occurs.

Specifying the Mullins Effect Material Model with Exponential and Slow Decay Forms of Damage Variable Evolution in Abaqus

You define the undamaged material behavior using the hyperelastic material model (see *Hyperelastic Behavior of Rubberlike Materials*). You define the exponential and slow decay forms of the damage variable evolution by specifying the η_F, U_h and α parameters directly (optionally as functions of temperature and/or field variables).

Input File Usage: Use the following option to define parameters for the exponential form of the evolution of the damage variable.

**MULLINS EFFECT*, DEFINITION=EXPONENTIAL

Input File Usage: Use the following option to define parameters for the slow decay form of the evolution of the damage variable.

**MULLINS EFFECT*, DEFINITION=SLOW DECAY

Convergence Difficulties in Abaqus/Standard

Models with the exponential and slow decay forms of the damage variable might exhibit strain softening behavior as presented in *Figure 6* for certain values of material parameters (for example such as to model material degradation and failure). This leads to instabilities and strain localization effects that often cause severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. Therefore, Abaqus/Standard should not be used in such cases, and it is recommended to use Abaqus/Explicit instead.

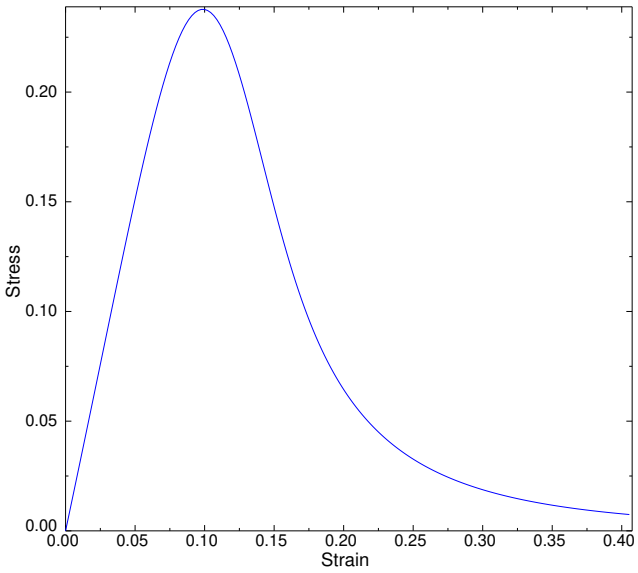


Figure 6: Uniaxial tension of Neo-Hookean material with slow decay form of damage variable evolution

User Subroutine Specification

An alternative method for defining the Mullins effect involves defining the damage variable in user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit. Optionally, you can specify the number of property values needed as data in the user subroutine. You must provide the damage variable, η, and its derivative, $\frac{d\eta}{d\tilde{U}_{dev}}$. The latter contributes to the Jacobian of the overall system of equations and is required to ensure good convergence characteristics in Abaqus/Standard. If needed, you can specify the number of solution-dependent variables (*About User Subroutines and Utilities*). These solution-dependent variables can be updated in the user subroutine. The damage dissipation energy and the recoverable part of the energy can also be defined for output purposes.

User subroutines *UMULLINS* and *VUMULLINS* can be used in combination with all hyperelastic potentials in Abaqus, including user-defined potentials (by means of user subroutines *UHYPER*, *UANISOHYPER_INV*, and *UANISOHYPER_STRAIN* in Abaqus/Standard, and *VUANISOHYPER_INV* and *VUANISOHYPER_STRAIN* in Abaqus/Explicit).

Input File Usage: **MULLINS EFFECT*, DEFINITION=USER, PROPERTIES=*constants*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **User Defined**

Viscoelasticity

When viscoelasticity is used in combination with Mullins effect, stress softening is applied to the long-term behavior.

In this case you should specify the parameters mU_h carefully. If the underlying hyperelastic behavior is defined with an instantaneous modulus, m and U_h are interpreted to be instantaneous. Otherwise, $m$and U_h are considered

to be long term. The parameters m and U_h have units of energy density (that is, energy per unit volume), which is equivalent to stress units.

Elements

The Mullins effect material model can be used with all element types that support the use of the hyperelastic material model, except for beam and one-dimensional elements for polynomial, Arruda-Boyce, and Van der Waals hyperelastic models in Abaqus/Explicit.

Procedures

The Mullins effect material model can be used in all procedure types that support the use of the hyperelastic material model. In linear perturbation steps in Abaqus/Standard, the current material tangent stiffness is used to determine the response. Specifically, when a linear perturbation is carried out about a base state that is on the primary curve, Abaqus/Standard uses the unloading tangent stiffness.

In Abaqus/Explicit, the unloading tangent stiffness is always used to compute the stable time increment. Therefore, the inclusion of Mullins effect leads to more increments in the analysis, even when no unloading takes place.

The Mullins effect material model can also be used in a steady-state transport analysis in Abaqus/Standard to obtain steady-state rolling solutions. For more information, see *Steady-State Transport Analysis* and *Analysis of a solid disc with Mullins effect and permanent set*.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the Mullins effect material model:

DMENER	Energy dissipated per unit volume by damage.
ELDMD	Total energy dissipated in element by damage.
ALLDMD	Energy dissipated in whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
SENER	The recoverable part of the energy per unit volume.
ELSE	The recoverable part of the energy in the element.
ALLSE	The recoverable part of the energy in the whole (partial) model.
ESEDEN	The recoverable part of the energy per unit volume in the element.
MSDMG	The damage variable, η.
UMULMAX	The maximum deviatoric strain energy density, U^m_{dev}.

References

- Ogden, R. W., and D. G. Roxburgh, "A Pseudo-Elastic Model for the Mullins Effect in Filled Rubber," *Proceedings of the Royal Society of London, Series A*, vol. 455 2861–2877, 1999.

Stress Softening in Elastomeric Foams

Products: Abaqus/Standard Abaqus/Explicit

References:

- *About the Material Library*
- *Combining Material Behaviors*
- *Elastic Behavior*
- *Hyperelastic Behavior in Elastomeric Foams*
- *Stress Softening in Rubberlike Materials*
- *HYPERFOAM
- *MULLINS EFFECT
- *UNIAXIAL TEST DATA
- *BIAXIAL TEST DATA
- *PLANAR TEST DATA

Overview

Energy dissipation in elastomeric foams in Abaqus:

- uses an approach based on the generalized Mullins effect functionality for elastomeric rubbers (*Stress Softening in Rubberlike Materials*);
- provides an extension to the isotropic elastomeric foam model (*Hyperelastic Behavior in Elastomeric Foams*);
- is intended for modeling energy absorption in foam components subjected to dynamic loading under deformation rates that are high compared to the characteristic relaxation time of the foam; and
- cannot be used with viscoelasticity in Abaqus/Standard.

Energy Dissipation in Elastomeric Foams

Abaqus provides a mechanism to include permanent energy dissipation and stress softening effects in elastomeric foams. The approach is similar to that used to model the Mullins effect in elastomeric rubbers, described in *Stress Softening in Rubberlike Materials*. The functionality is primarily intended for modeling energy absorption in foam components subjected to dynamic loading under deformation rates that are high compared to the characteristic relaxation time of the foam; in such cases it is acceptable to assume that the foam material is damaged permanently.

The material response is depicted qualitatively in *Figure 1*.

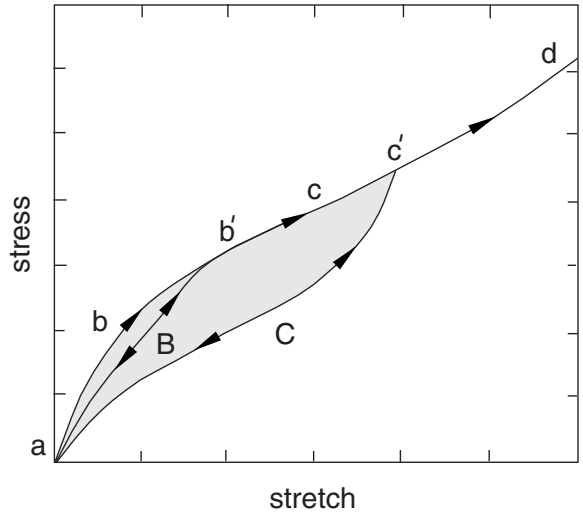


Figure 1: Typical stress-stretch response of an elastomeric foam material with energy dissipation.

Consider the primary loading path abb' of a previously unstressed foam, with loading to an arbitrary point b'. On unloading from b', the path $b'Ba$ is followed. When the material is loaded again, the softened path is retraced as aBb'. If further loading is then applied, the path $b'c$ is followed, where $b'c$ is a continuation of the primary loading path $abb'cc'd$ (which is the path that would be followed if there were no unloading). If loading is now stopped at c', the path $c'Ca$ is followed on unloading and then retraced back to c' on reloading. If no further loading beyond c' is applied, the curve aCc' represents the subsequent material response, which is then elastic. For loading beyond c', the primary path is again followed and the pattern described is repeated. The shaded area in *Figure 1* represents the energy dissipated by damage in the material for deformation until c'.

Modified Strain Energy Density Function

Energy dissipation effects are accounted for by introducing an augmented strain energy density function of the form

$$U\left(\widehat{\lambda}_i,\eta\right)=\eta\widetilde{U}\left(\widehat{\lambda}_i\right)+\phi\left(\eta\right),$$

where $\widehat{\lambda}_i(i=1,2,3)$ represent the principal mechanical stretches and $\widetilde{U}\left(\widehat{\lambda}_i\right)$ is the strain energy potential for the primary foam behavior described in *Hyperelastic Behavior in Elastomeric Foams*, defined by the polynomial strain energy function

$$\widetilde{U}\left(\widehat{\lambda}_i\right)=\sum_{i=1}^{N}\frac{2\mu_i}{\alpha_i^2}\left[\widehat{\lambda}_1^{\alpha_i}+\widehat{\lambda}_2^{\alpha_i}+\widehat{\lambda}_3^{\alpha_i}-3+\frac{1}{\beta_i}\left(\left(J^{el}\right)^{-\alpha_i\beta_i}-1\right)\right].$$

The function $\phi\left(\eta\right)$ is a continuous function of the damage variable, η, and is referred to as the "damage function." The damage variable varies continuously during the course of the deformation and always satisfies $0<\eta\leq 1$, with $\eta=1$ on the points of the primary curve. The damage function $\phi\left(\eta\right)$ satisfies the condition $\phi\left(1\right)=0$; thus, when the deformation state of the material is on a point on the curve that represents the primary foam behavior,

$U\left(\widehat{\lambda}_i, 1\right) = \widetilde{U}\left(\widehat{\lambda}_i\right)$ and the augmented energy function reduces to the strain energy potential for the primary foam behavior.

The above expression of the augmented strain energy density function is similar to the form proposed by Ogden and Roxburgh to model the Mullins effect in filled rubber elastomers (see *Stress Softening in Rubberlike Materials*), with the difference that in the case of elastomeric foams an augmentation of the total strain energy (including the volumetric part) is considered. This modification is required for the model to predict energy absorption under pure hydrostatic loading of the foam.

Stress Computation

With the above modification to the energy function, the stresses are given by

$$\boldsymbol{\sigma}\left(\eta, \widehat{\lambda}_i\right) = \eta\widetilde{\boldsymbol{\sigma}}\left(\widehat{\lambda}_i\right),$$

where $\widetilde{\boldsymbol{\sigma}}$ is the stress corresponding to the primary foam behavior at the current deformation level $\widehat{\lambda}_i$. Thus, the stress is obtained by simply scaling the stress of the primary foam behavior by the damage variable, η. From any given strain level the model predicts unloading/reloading along a single curve (that is different, in general, from the primary foam behavior) that passes through the origin of the stress-strain plot. The model also predicts energy dissipation under purely volumetric deformation.

Damage Variable

The damage variable, η, varies with the deformation according to

$$\eta = 1 - \frac{1}{r}\,\mathrm{erf}\left(\frac{U^m - \widetilde{U}}{m + \beta U^m}\right),$$

where U^m is the maximum value of $\widetilde{U}$ at a material point during its deformation history; r, β, and m are material parameters; and $\mathrm{erf}\,(x)$ is the error function. When $\widetilde{U} = U^m$, corresponding to a point on the primary curve, $\eta = 1.0$. On the other hand, upon removal of deformation, when $\widetilde{U} = 0$, the damage variable, η, attains its minimum value, η_m, given by

$$\eta_m = 1 - \frac{1}{r}\,\mathrm{erf}\left(\frac{U^m}{m + \beta U^m}\right).$$

For all intermediate values of $\widetilde{U}$, η varies monotonically between 1.0 and η_m. While the parameters r and β are dimensionless, the parameter m has the dimensions of energy. The material parameters can be specified directly or can be computed by Abaqus based on curve fitting of unloading-reloading test data. These parameters are subject to the restrictions $r > 1$, $\beta \geq 0$, and $m \geq 0$ (the parameters β and m cannot both be zero). Alternatively, the damage variable η can be defined through user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit.

If the parameter $\beta = 0$ and the parameter m has a value that is small compared to U^m, the slope of the stress-strain curve at the initiation of unloading from relatively large strain levels might become very high. As a result, the response might become discontinuous. This kind of behavior might lead to convergence problems in Abaqus/Standard. In Abaqus/Explicit the high stiffness will lead to very small stable time increments, thereby leading to a degradation in performance. This problem can be avoided by choosing a small value for β. In Abaqus/Standard the default value of β is 0. In Abaqus/Explicit, however, the default value of β is 0.1. Thus, if you do not specify a value for β, it is assumed to be 0 in Abaqus/Standard and 0.1 in Abaqus/Explicit.

The parameters r, β, and m do not have direct physical interpretations in general. The parameter m controls whether damage occurs at low strain levels. If $m = 0$, there is a significant amount of damage at low strain levels. On the other hand, a nonzero m leads to little or no damage at low strain levels. For further discussion regarding the implications of this model on the energy dissipation, see *Mullins effect*.

Specifying Properties for Energy Dissipation in Elastomeric Foams

The primary elastomeric foam behavior is defined by using the hyperfoam material model. Energy dissipation can be defined by specifying the parameters in the expression of the damage variable directly or by using test data to calibrate the parameters. Alternatively, you can define the Mullins effect model with user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit.

Specifying the Parameters Directly

The parameters r, m, and β in the expression of the damage variable can be given directly as functions of temperature and/or field variables.

Input File Usage: **MULLINS EFFECT*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **Constants**

Using Test Data to Calibrate the Parameters

Experimental unloading-reloading data from different strain levels can be specified for up to three simple tests: uniaxial, biaxial, and planar. Abaqus will then compute the material parameters using a nonlinear least-squares curve fitting algorithm. See *Stress Softening in Rubberlike Materials* for a detailed discussion of this approach.

Input File Usage: **MULLINS EFFECT*, TEST DATA INPUT, BETA *and/or* M *and/or* R

In addition, use at least one and up to three of the following options to give the unloading-reloading test data:

**UNIAXIAL TEST DATA*
**BIAXIAL TEST DATA*
**PLANAR TEST DATA*

Multiple unloading-reloading curves from different strain levels for any given test type can be entered by repeated specification of the appropriate test data option.

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **Test Data Input**: enter the values for up to two of the values **r**, **m**, and **beta**. In addition, enter data for at least one of the following **Suboptions->Biaxial Test**, **Planar Test**, or **Uniaxial Test**

User Subroutine Specification

An alternative method for specifying energy dissipation involves defining the damage variable in user subroutine *UMULLINS* in Abaqus/Standard and *VUMULLINS* in Abaqus/Explicit. Optionally, you can specify the number of property values needed as data in the user subroutine. You must provide the damage variable, η, and its

derivative, $\frac{d\eta}{d\tilde{U}}$. The latter contributes to the Jacobian of the overall system of equations and is necessary to ensure good convergence characteristics in Abaqus/Standard. If needed, you can specify the number of solution-dependent variables (*About User Subroutines and Utilities*). These solution-dependent variables can be updated in the user subroutine. The damage dissipation energy and the recoverable part of the energy can also be defined for output purposes.

Input File Usage: **MULLINS EFFECT*, USER, PROPERTIES=constants

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Damage for Elastomers->Mullins Effect: **Definition**: **User Defined**

Elements

The model can be used with all element types that support the use of the elastomeric foam material model.

Procedures

The model can be used in all procedure types that support the use of the elastomeric foam material model. In linear perturbation steps in Abaqus/Standard the current material tangent stiffness is used to determine the response. Specifically, when a linear perturbation is carried out about a base state that is on the primary curve, the unloading tangent stiffness will be used.

In Abaqus/Explicit the unloading tangent stiffness is always used to compute the stable time increment. As a result, the inclusion of stress-softening effects might lead to more increments in the analysis, even when no unloading actually takes place.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the Mullins effect material model:

DMENER	Energy dissipated per unit volume by damage.
ELDMD	Total energy dissipated in element by damage.
ALLDMD	Energy dissipated in whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
SENER	The recoverable part of the energy per unit volume.
ELSE	The recoverable part of the energy in the element.
ALLSE	The recoverable part of the energy in the whole (partial) model.
ESEDEN	The recoverable part of the energy per unit volume in the element.

MSDMG The damage variable, η.

UMULMAX The maximum strain energy density, U^m.

The damage energy dissipation, represented by the shaded area in *Figure 1* for deformation until $\bar{\mathbf{c}}'$, is computed as follows. When the damaged material is in a fully unloaded state, the augmented energy function has the residual value $U(\mathbf{I}, \eta_m) = \phi(\eta_m)$. The residual value of the energy function upon complete unloading represents the energy dissipated due to damage in the material. The recoverable part of the energy is obtained by subtracting the dissipated energy from the augmented energy as $\eta\tilde{U}\left(\hat{\lambda}_i\right) + \phi(\eta) - \phi(\eta_m)$.

The damage energy accumulates with progressive deformation along the primary curve and remains constant during unloading. During unloading, the recoverable part of the strain energy is released. The latter becomes zero when the material point is unloaded completely. Upon further reloading from a completely unloaded state, the recoverable part of the strain energy increases from zero. When the maximum strain that was attained earlier is exceeded upon reloading, further accumulation of damage energy occurs.

Linear Viscoelasticity

The viscoelastic model is used to specify time-dependent material behavior. In Abaqus/Standard it is also used to specify frequency-dependent material behavior.

In this section:

- *Time Domain Viscoelasticity*
- *Frequency Domain Viscoelasticity*
- *Structural Relaxation in Glass*

Time Domain Viscoelasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Frequency Domain Viscoelasticity*
- **VISCOELASTIC*
- **SHEAR TEST DATA*
- **VOLUMETRIC TEST DATA*
- **COMBINED TEST DATA*
- **TRS*
- *Defining time domain viscoelasticity*

Overview

The time domain viscoelastic material model:

- describes rate-dependent material behavior for materials in which dissipative losses primarily caused by "viscous" (internal damping) effects must be modeled in the time domain;
- can be used to describe isotropic, transversely isotropic, and orthotropic dissipative losses in the time domain;
- can be used only with *Linear Elastic Behavior*, *Hyperelastic Behavior of Rubberlike Materials*, or *Hyperelastic Behavior in Elastomeric Foams* to define the continuum elastic material properties;
- can be used in Abaqus/Explicit with *Linear Elastic Traction-Separation Behavior*;
- is active only during a transient static analysis (*Quasi-Static Analysis*), a transient implicit dynamic analysis (*Implicit Dynamic Analysis Using Direct Integration*), an explicit dynamic analysis (*Explicit Dynamic Analysis*), a steady-state transport analysis (*Steady-State Transport Analysis*), a fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*), a fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*), or a transient (consolidation) coupled pore fluid diffusion and stress analysis (*Coupled Pore Fluid Diffusion and Stress Analysis*);
- can be used in large-strain problems;
- can be calibrated using time-dependent creep test data, time-dependent relaxation test data, or frequency-dependent cyclic test data; and
- can be used to couple viscous dissipation with the temperature field in a fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*) or a fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*).

Defining Isotropic Viscoelasticity

Isotropic time domain viscoelasticity is available in Abaqus for small-strain applications where the rate-independent elastic response can be defined with a linear elastic material model and for large-strain applications where the rate-independent elastic response must be defined with a hyperelastic or hyperfoam material model.

For isotropic time domain viscoelasticity, Abaqus assumes that the shear (deviatoric) and volumetric behaviors are independent in multiaxial stress states (except when used for an elastomeric foam).

Defining the Shear Behavior

You can define the shear behavior at small strain and at large strain.

Small Strain

Consider a shear test at small strain in which a time varying shear strain, $\gamma(t)$, is applied to the material. The response is the shear stress $\tau(t)$. The viscoelastic material model defines $\tau(t)$ as

$$\tau(t) = \int_0^t G_R(t-s)\,\dot{\gamma}(s)\,ds,$$

where $G_R(t)$ is the time-dependent "shear relaxation modulus" that characterizes the material's response. This constitutive behavior can be illustrated by considering a relaxation test in which a strain γ is suddenly applied to a specimen and then held constant for a long time. The beginning of the experiment, when the strain is suddenly applied, is taken as zero time, so that

$$\tau(t) = \int_0^t G_R(t-s)\,\dot{\gamma}(s)\,ds = G_R(t)\,\gamma \quad (\text{since } \dot{\gamma} = 0 \text{ for } t > 0),$$

where γ is the fixed strain. The viscoelastic material model is "long-term elastic" in the sense that, after having been subjected to a constant strain for a very long time, the response settles down to a constant stress; that is, $G_R(t) \to G_\infty$ as $t \to \infty$.

The shear relaxation modulus can be written in dimensionless form:

$$g_R(t) = G_R(t)/G_0,$$

where $G_0 = G_R(0)$ is the instantaneous shear modulus, so that the expression for the stress takes the form

$$\tau(t) = G_0 \int_0^t g_R(t-s)\,\dot{\gamma}(s)\,ds.$$

The dimensionless relaxation function has the limiting values $g_R(0) = 1$ and $g_R(\infty) = G_\infty/G_0$.

Anisotropic Elasticity in Abaqus/Explicit

The equation for the shear stress can be transformed using integration by parts:

$$\tau(t) = G_0\left(\gamma + \int_0^t \dot{g}_R(s)\,\gamma(t-s)\,ds\right).$$

It is convenient to write this equation in the form

$$\tau(t) = \tau_0(t) + \int_0^t \dot{g}_R(s)\,\tau_0(t-s)\,ds,$$

where $\tau_0(t)$ is the instantaneous shear stress at time *t*. This can be generalized to multidimensions as

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_0(t) + \int_0^t \dot{g}_R(s)\,\boldsymbol{\tau}_0(t-s)\,ds,$$

where $\boldsymbol{\tau}(t)$ is the deviatoric part of the stress tensor and $\boldsymbol{\tau}_0(t)$ is the deviatoric part of the instantaneous stress tensor. Here the viscoelasticity is assumed to be isotropic; that is, the relaxation function is independent of the loading direction.

This form allows a straightforward generalization to anisotropic elastic deformations, where the deviatoric part of the instantaneous stress tensor is computed as $\boldsymbol{\tau}_0(t) = \overline{\mathbf{D}}_0 : \mathbf{e}$. Here $\overline{\mathbf{D}}_0$ is the instantaneous deviatoric elasticity tensor, and $\mathbf{e}$ is the deviatoric part of the strain tensor.

Large Strain

The above form also allows a straightforward generalization to nonlinear elastic deformations, where the deviatoric part of the instantaneous stress $\boldsymbol{\tau}_0(t)$ is computed using a hyperelastic strain energy potential. This generalization yields a linear viscoelasticity model, in the sense that the dimensionless stress relaxation function is independent of the magnitude of the deformation.

In the above equation the instantaneous stress, $\boldsymbol{\tau}_0$, applied at time $t - s$ influences the stress, $\boldsymbol{\tau}$, at time t. Therefore, to create a proper finite-strain formulation, map the stress that existed in the configuration at time $t - s$ into the configuration at time t. In Abaqus, you can do this with the "standard-push-forward" transformation with the relative deformation gradient $\mathbf{F}_{t-s}(t)$:

$$\mathbf{F}_{t-s}(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t-s)},$$

that results in the following hereditary integral:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{dev}\left[\int_0^t \dot{g}_R(s)\,\overline{\mathbf{F}}_t^{-1}(t-s)\cdot\boldsymbol{\tau}_0(t-s)\cdot\overline{\mathbf{F}}_t^{-T}(t-s)\,ds\right],$$

where $\boldsymbol{\tau}$ is the deviatoric part of the Kirchhoff stress.

The finite-strain theory is described in more detail in *Finite-strain viscoelasticity*.

Defining the Volumetric Behavior

The volumetric behavior can be written in a form that is similar to the shear behavior:

$$p(t) = -\mathbf{K}_0 \int_0^t k_R(t-s)\,\dot{\varepsilon}^{vol}(s)\,ds,$$

where

p
: is the hydrostatic pressure;

$\mathbf{K}_0$
: is the instantaneous elastic bulk modulus;

$k_R(t)$
: s the dimensionless bulk relaxation modulus; and

ε^{vol}
: is the volume strain.

The expansion above applies to small strain and finite strain because the volume strains are generally small and you do not need to map the pressure from time $t - s$ to time t.

Defining Nonisotropic Viscoelasticity

The stress response of materials with nonisotropic viscoelastic behavior can be written in a general tensorial form:

$$\sigma_{ij}(t) = \int_0^t D_{R,ijkl}(t-s)\,\dot{\epsilon}_{kl}(s)\,ds,$$

where $D_{R,ijkl}(t)$ is the $ijkl$ component of the time-dependent "relaxation elastic modulus" $\mathbf{D}_R(t)$.

The $ijkl$ component of the relaxation elastic modulus can be written in dimensionless form (no summation over repeated indices):

$$r_{R,(ijkl)}(t) = D_{R,ijkl}(t)/D_{0,ijkl}$$

where $D_{0,ijkl}$ is the $ijkl$ component of the instantaneous elastic modulus $\mathbf{D}_0 = \mathbf{D}_R(0)$, so that the expression for the stress takes the form (no summation over indices inside the parentheses)

$$\sigma_{ij}(t) = D_{0,ijkl}\int_0^t r_{R,(ijkl)}(t-s)\,\dot{\epsilon}_{kl}(s)\,ds,$$

Nonisotropic viscoelasticity is available in Abaqus/Standard for small-strain deformation only.

Defining Viscoelastic Behavior for Traction-Separation Elasticity in Abaqus/Explicit

Time domain viscoelasticity can be used in Abaqus/Explicit to model rate-dependent behavior of cohesive elements with traction-separation elasticity (*Defining Elasticity in Terms of Tractions and Separations for Cohesive Elements*). In this case the evolution equation for the normal and two shear nominal tractions take the form:

$$t_n(t) = t_n^0(t) + \int_0^t \dot{k}_R(s)\,t_n^0(t-s)\,ds,$$
$$t_s(t) = t_s^0(t) + \int_0^t \dot{g}_R(s)\,t_s^0(t-s)\,ds,$$
$$t_t(t) = t_t^0(t) + \int_0^t \dot{g}_R(s)\,t_t^0(t-s)\,ds,$$

where $t_n^0(t)$, $t_s^0(t)$, and $t_t^0(t)$ are the instantaneous nominal tractions at time t in the normal and the two local shear directions, respectively. The functions $g_R(t)$ and $k_R(t)$ now represent the dimensionless shear and normal relaxation moduli, respectively. There is close similarity between the viscoelastic formulation for the continuum elastic response discussed in the previous sections and the formulation for cohesive behavior with traction-separation elasticity after reinterpreting shear and bulk relaxation as shear and normal relaxation.

For the case of uncoupled traction elasticity, the viscoelastic normal and shear behaviors are assumed to be independent. The normal relaxation modulus is defined as

$$k_R(t) = K_{nn}(t)/K_{nn}^0,$$

where K_{nn}^0 is the instantaneous normal moduli. The shear relaxation modulus is assumed to be isotropic and, therefore, independent of the local shear directions:

$$g_R(t) = K_{ss}(t)/K_{ss}^0 = K_{tt}(t)/K_{tt}^0,$$

where K_{ss}^0 and K_{tt}^0 are the instantaneous shear moduli.

For the case of coupled traction-separation elasticity, the normal and shear relaxation moduli must be the same, $g_R(t) = k_R(t)$, and you must use the same relaxation data for both behaviors.

Temperature Effects

The effect of temperature, θ, on the material behavior is introduced through the dependence of instantaneous stress, τ_0, on temperature and through a reduced time concept. For isotropic viscoelasticity, the expression for the linear-elastic shear stress is rewritten as

$$\tau(t) = G_0(\theta) \int_0^t g_R(\xi(t) - \xi(s)) \dot{\gamma}(s)\, ds,$$

where the instantaneous shear modulus G_0 is temperature dependent and $\xi(t)$ is the reduced time, defined by

$$\xi(t) = \int_0^t \frac{ds}{A(\theta(s))},$$

where $A(\theta(t))$ is a shift function at time t. This reduced time concept for temperature dependence is usually referred to as thermorheologically simple (TRS) temperature dependence. Often the shift function is approximated by the Williams-Landel-Ferry (WLF) form. See *Thermorheologically Simple Temperature Effects* below, for a description of the WLF and other forms of the shift function available in Abaqus.

The reduced time concept is also used for the volumetric behavior of isotropic viscoelasticity, the isotropic viscoelasticity with large-strain formulation, the nonisotropic viscoelasticity with small-strain formulation, and the traction-separation formulation.

Numerical Implementation

Abaqus assumes that the viscoelastic material is defined by a Prony series expansion of the dimensionless relaxation modulus:

$$g_R(t) = 1 - \sum_{i=1}^{N} \overline{g}_i^P \left(1 - e^{-t/\tau_i^G}\right),$$

where N, $\overline{g}_i^P$, and τ_i^G, $i = 1, 2, \ldots, N$, are material constants.

Isotropic Viscoelasticity with Elasticity

For Isotropic viscoelasticity combined with linear isotropic elasticity, substitution in the small-strain expression for the shear stress yields

$$\tau(t) = G_0 \left(\gamma - \sum_{i=1}^{N} \gamma_i\right),$$

where

$$\gamma_i = \frac{\overline{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \gamma(t-s)\, ds.$$

The γ_i are interpreted as state variables that control the stress relaxation, and

$$\gamma^{cr} = \sum_{i=1}^{N} \gamma_i$$

is the "creep" strain: the difference between the total mechanical strain and the instantaneous elastic strain (the stress divided by the instantaneous elastic modulus). In Abaqus/Standard γ^{cr} is available as the creep strain output variable CE (*Abaqus/Standard Output Variable Identifiers*).

A similar Prony series expansion is used for the volumetric response, which is valid for both small- and finite-strain applications:

$$p = -K_0 \left(\varepsilon^{vol} - \sum_{i=1}^{N} \varepsilon_i^{vol} \right),$$

where

$$\varepsilon_i^{vol} = \frac{\bar{k}_i^P}{\tau_i^K} \int_0^t e^{-s/\tau_i^K} \varepsilon^{vol} \left(t - s \right) \, ds.$$

Abaqus assumes that $\tau_i^G = \tau_i^K = \tau_i$.

For isotropic viscoelasticity combined with linear anisotropic elasticity, when the deviatoric and volumetric behavior are decoupled or share the same relaxation data, the Prony series expansion, in combination with the generalized small-strain expression for the deviatoric stress, yields

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 - \sum_{i=1}^{N} \boldsymbol{\tau}_i,$$

where

$$\boldsymbol{\tau}_i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \boldsymbol{\tau}_0 \left(t - s \right) \, ds.$$

The $\boldsymbol{\tau}_i$ are interpreted as state variables that control the stress relaxation.

For finite strains, the Prony series expansion, in combination with the finite-strain expression for the shear stress, produces the following expression for the deviatoric stress:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 - \sum_{i=1}^{N} \mathbf{dev} \left(\boldsymbol{\tau}_i \right),$$

where

$$\boldsymbol{\tau}_i = \frac{\bar{g}_i^P}{\tau_i^G} \int_0^t e^{-s/\tau_i^G} \overline{\mathbf{F}}_t^{-1} \left(t - s \right) \cdot \boldsymbol{\tau}_0 \left(t - s \right) \cdot \overline{\mathbf{F}}_t^{-T} \left(t - s \right) \, ds.$$

The $\boldsymbol{\tau}_i$ are interpreted as state variables that control the stress relaxation.

When isotropic viscoelasticity is defined, if the instantaneous material behavior is defined by linear elasticity or hyperelasticity, the bulk and shear behavior can be defined independently. However, if the instantaneous behavior is defined by the hyperfoam model, the deviatoric and volumetric constitutive behavior are coupled and it is mandatory to use the same relaxation data for both behaviors. For linear anisotropic elasticity, the same relaxation data should be used for both behaviors when the elasticity definition is such that the deviatoric and volumetric response is coupled; otherwise, relaxation data should be defined for each component of the relaxation modulus.

In all the above expressions, you can introduce temperature dependence by replacing e^{-s/τ_i^G} by $e^{-\xi(s)/\tau_i^G}$ and e^{-s/τ_i^K} by $e^{-\xi(s)/\tau_i^K}$.

Nonisotropic Viscoelasticity with Elasticity

For nonisotropic viscoelasticity, the bulk and shear behavior can be coupled, and it is mandatory to specify relaxation behavior for all independent constants of the elastic moduli. The Prony series expansion, in combination with the generalized small-strain elasticity expression for the stress, yields

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_0 - \sum_{i=1}^{N} \boldsymbol{\sigma}_i,$$

where

$$\sigma_{i,mn} = \frac{\bar{\tau}^P_{i,(mnkl)}}{\tau_i} D_{0,mnkl} \int_0^t e^{-s/\tau_i} \epsilon_{kl}\,(t-s)\;\; ds.$$

$\bar{\tau}^P_{i,(mnkl)}$ are the parameters of the Prony series for the $mnkl$ component of the dimensionless relaxation modulus. Abaqus assumes that all components of the relaxation modulus share the same τ_i. There are no summation over indices inside the parentheses in the equation above.

In the expression above, you can introduce temperature dependence by replacing e^{-s/τ_i} with $e^{-\xi(s)/\tau_i}$.

Traction-Separation Elasticity

For traction-separation elasticity, the Prony series expansion yields

$$\mathbf{t} = \begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{Bmatrix} t^0_n \\ t^0_s \\ t^0_t \end{Bmatrix} - \sum_{i=1}^{N} \begin{Bmatrix} t^i_n \\ t^i_s \\ t^i_t \end{Bmatrix} = \mathbf{t}^0 - \sum_{i=1}^{N} \mathbf{t}^i,$$

where

$$t^i_n = \frac{\bar{k}^P_i}{\tau^K_i} \int_0^t e^{-s/\tau^K_i} t^0_n\,(t-s)\; ds,$$

$$t^i_s = \frac{\bar{g}^P_i}{\tau^G_i} \int_0^t e^{-s/\tau^G_i} t^0_s\,(t-s)\; ds,$$

$$t^i_t = \frac{\bar{g}^P_i}{\tau^G_i} \int_0^t e^{-s/\tau^G_i} t^0_t\,(t-s)\; ds.$$

The $\mathbf{t}^i$ are interpreted as state variables that control the relaxation of the traction stresses.

For coupled traction-separation elasticity you must use the same relaxation data for the normal and shear behaviors.

In all the expressions above, you can introduce temperature dependence by replacing e^{-s/τ^G_i} by $e^{-\xi(s)/\tau^G_i}$ and e^{-s/τ^K_i} by $e^{-\xi(s)/\tau^K_i}$.

Determination of Isotropic Viscoelastic Material Parameters

The above equations are used to model the time-dependent behavior of a viscoelastic material. For isotropic viscoelasticity, the relaxation parameters for the shear and volumetric behaviors can be defined in one of the following ways:

- direct specification of the Prony series parameters,
- inclusion of creep test data,
- inclusion of relaxation test data, or
- inclusion of frequency-dependent data obtained from sinusoidal oscillation experiments.

Temperature effects are included in the same manner regardless of the method used to define the viscoelastic material.

Abaqus/CAE allows you to evaluate the behavior of viscoelastic materials by automatically creating response curves based on experimental test data or coefficients. A viscoelastic material can be evaluated only if it is defined in the time domain and includes hyperelastic and/or elastic material data. See *Evaluating hyperelastic, hyperfoam and viscoelastic material behavior*.

Direct Specification

The Prony series parameters $\overline{g}_i^P$, $\overline{k}_i^P$, and τ_i can be defined directly for each term in the Prony series. There is no restriction on the number of terms that can be used. If a relaxation time is associated with only one of the two moduli, leave the other one blank or enter a zero. The number of lines of data given defines the number of terms in the Prony series, *N*. If this model is used in conjunction with the hyperfoam material model, the two modulus ratios must be the same ($\overline{g}_i^P = \overline{k}_i^P$).

Input File Usage: **VISCOELASTIC*, TIME=PRONY,TYPE=ISOTROPIC

The data line is repeated as often as needed to define the first, second, third, etc. terms in the Prony series.

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time** and **Time: Prony**

Enter as many rows of data in the table as needed to define the first, second, third, etc. terms in the Prony series.

Creep Test Data

If creep test data are specified, Abaqus calculates the terms in the Prony series automatically. The normalized shear and bulk compliances are defined as

$$j_S(t) = G_0 J_S(t) \qquad \text{and} \qquad j_K(t) = K_0 J_K(t),$$

where $J_S(t) = \gamma(t)/\tau_0$ is the shear compliance, $\gamma(t)$ is the total shear strain, and τ_0 is the constant shear stress in a shear creep test; $J_K(t) = \varepsilon^{vol}(t)/p_0$ is the volumetric compliance; $\varepsilon^{vol}(t)$ is the total volumetric strain; and p_0 is the constant pressure in a volumetric creep test. At time $t = 0$, $j_S(0) = j_K(0) = 1$.

The creep data are converted to relaxation data through the convolution integrals

$$\int_0^t g_R(s)\, j_S(t-s)\, ds = t \qquad \text{and} \qquad \int_0^t k_R(s)\, j_K(t-s)\, ds = t.$$

Abaqus then uses the normalized shear modulus $g_R(t)$ and normalized bulk modulus $k_R(t)$ in a nonlinear least-squares fit to determine the Prony series parameters.

Using the Shear and Volumetric Test Data Consecutively

The shear test data and volumetric test data can be used consecutively to define the normalized shear and bulk compliances as functions of time. A separate least-squares fit is performed on each data set; and the two derived sets of Prony series parameters, $\left(\overline{g}_i^P, \tau_i^G\right)$ and $\left(\overline{k}_i^P, \tau_i^K\right)$, are merged into one set of parameters, $\left(\overline{g}_i^P, \overline{k}_i^P, \tau_i\right)$.

Input File Usage: Use the following three options (the first option is required; only one of the second and third options is required):

**VISCOELASTIC*, TIME=CREEP TEST DATA
**SHEAR TEST DATA*
**VOLUMETRIC TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time** and **Time: Creep test data**

In addition, select one or both of the following:

Test Data->Shear Test Data
Test Data->Volumetric Test Data

Using the Combined Test Data

Alternatively, the combined test data can be used to specify the normalized shear and bulk compliances simultaneously as functions of time. A single least-squares fit is performed on the combined set of test data to determine one set of Prony series parameters, $\left(\overline{g}_i^P, \overline{k}_i^P, \tau_i\right)$.

Input File Usage: Use both of the following options:

**VISCOELASTIC*, TIME=CREEP TEST DATA
**COMBINED TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time**, **Time: Creep test data**, and **Test Data->Combined Test Data**

Relaxation Test Data

As with creep test data, Abaqus will calculate the Prony series parameters automatically from relaxation test data.

Using the Shear and Volumetric Test Data Consecutively

Again, the shear test data and volumetric test data can be used consecutively to define the relaxation moduli as functions of time. A separate nonlinear least-squares fit is performed on each data set; and the two derived sets of Prony series parameters, $\left(\overline{g}_i^P, \tau_i^G\right)$ and $\left(\overline{k}_i^P, \tau_i^K\right)$, are merged into one set of parameters, $\left(\overline{g}_i^P, \overline{k}_i^P, \tau_i\right)$.

Input File Usage: Use the following three options (the first option is required; only one of the second and third options is required):

**VISCOELASTIC*, TIME=RELAXATION TEST DATA
**SHEAR TEST DATA*
**VOLUMETRIC TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time** and **Time: Relaxation test data**

In addition, select one or both of the following:

Test Data->Shear Test Data
Test Data->Volumetric Test Data

Using the Combined Test Data

Alternatively, the combined test data can be used to specify the relaxation moduli simultaneously as functions of time. A single least-squares fit is performed on the combined set of test data to determine one set of Prony series parameters, $\left(\overline{g}_i^P, \overline{k}_i^P, \tau_i\right)$.

Input File Usage: Use both of the following options:

**VISCOELASTIC*, TIME=RELAXATION TEST DATA
**COMBINED TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time**, **Time: Relaxation test data**, and **Test Data->Combined Test Data**

Frequency-Dependent Test Data

The Prony series terms can also be calibrated using frequency-dependent test data. In this case Abaqus uses analytical expressions that relate the Prony series relaxation functions to the storage and loss moduli. The expressions for the shear moduli, obtained by converting the Prony series terms from the time domain to the frequency domain by making use of Fourier transforms, can be written as follows:

$$G_s(\omega) = G_0\left[1 - \sum_{i=1}^{N} \overline{g}_i^P\right] + G_0 \sum_{i=1}^{N} \frac{\overline{g}_i^P \tau_i^2 \omega^2}{1+\tau_i^2\omega^2},$$

$$G_\ell(\omega) = G_0 \sum_{i=1}^{N} \frac{\overline{g}_i^P \tau_i \omega}{1+\tau_i^2\omega^2},$$

where $G_s(\omega)$ is the storage modulus, $G_\ell(\omega)$ is the loss modulus, ω is the angular frequency, and N is the number of terms in the Prony series. These expressions are used in a nonlinear least-squares fit to determine the Prony series parameters from the storage and loss moduli cyclic test data obtained at M frequencies by minimizing the error function χ^2:

$$\chi^2 = \sum_{i=1}^{M} \frac{1}{G_\infty^2}\left[\left(G_s - \overline{G}_s\right)_i^2 + \left(G_\ell - \overline{G}_\ell\right)_i^2\right],$$

where $\overline{G}_s$ and $\overline{G}_\ell$ are the test data and G_0 and G_∞, respectively, are the instantaneous and long-term shear moduli. The expressions for the bulk moduli, $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$, are written analogously.

The frequency domain data are defined in tabular form by giving the real and imaginary parts of ωg^* and ωk^*—where ω is the circular frequency—as functions of frequency in cycles per time. $g^*(\omega)$ is the Fourier transform of the nondimensional shear relaxation function $g(t) = \frac{G_R(t)}{G_\infty} - 1$. Given the frequency-dependent storage and loss moduli $G_s(\omega)$, $G_\ell(\omega)$, $\mathrm{K}_s(\omega)$, and $\mathrm{K}_\ell(\omega)$, the real and imaginary parts of ωg^* and ωk^* are then given as

$$\omega\Re(g^*) = G_\ell/G_\infty, \qquad \omega\Im(g^*) = 1 - G_s/G_\infty, \quad \omega\Re(k^*) = \mathrm{K}_\ell/\mathrm{K}_\infty, \qquad \omega\Im(k^*) = 1 - \mathrm{K}_s/\mathrm{K}_\infty,$$

where G_∞ and K_∞ are the long-term shear and bulk moduli determined from the elastic or hyperelastic properties.

Input File Usage: **VISCOELASTIC*, TIME=FREQUENCY DATA

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time** and **Time: Frequency data**

Calibrating the Prony Series Parameters

You can specify two optional parameters related to the calibration of Prony series parameters for viscoelastic materials: the error tolerance and N_{max}. The error tolerance is the allowable average root-mean-square error of data points in the least-squares fit, and its default value is 0.01. N_{max} is the maximum number of terms N in the Prony series, and its default (and maximum) value is 13. Abaqus will perform the least-squares fit from $N = 1$ to $N = N_{max}$ until convergence is achieved for the lowest N with respect to the error tolerance.

You can use the following guidelines to determine the number of terms in the Prony series from test data. Based on these guidelines, you can choose N_{max}.

- There should be enough data pairs for determining all the parameters in the Prony series terms. Therefore, assuming that N is the number of Prony series terms, there should be a total of at least $2N$ data points in shear test data, $2N$ data points in volumetric test data, $3N$ data points in combined test data, and $2N$ data points in the frequency domain.
- The number of terms in the Prony series should be typically not more than the number of logarithmic "decades" spanned by the test data. The number of logarithmic "decades" is defined as $\log_{10}(t_{max}/t_{min})$, where t_{max} and t_{min} are the maximum and minimum time in the test data, respectively.

Input File Usage: **VISCOELASTIC*, ERRTOL=*error_tolerance*, NMAX=N_{max}

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Time**; **Time: Creep test data**, **Relaxation test data**, or **Frequency data**; **Maximum number of terms in the Prony series:** N_{max}; and **Allowable average root-mean-square error:** *error_tolerance*

Thermorheologically Simple Temperature Effects

Regardless of the method you use to define the viscoelastic behavior, you can include thermorheologically simple temperature effects by specifying the method to use to define the shift function. Abaqus supports the following forms of the shift function: the Williams-Landel-Ferry (WLF) form, the Arrhenius form, the tabular form, and user-defined forms.

Thermorheologically simple temperature effects can also be included in the definition of equation of state models with viscous shear behavior (see *Viscous Shear Behavior*).

Williams-Landel-Ferry (WLF) Form

The shift function can be defined by the Williams-Landel-Ferry (WLF) approximation, which takes the form:

$$\log_{10}(A) = -\frac{C_1(\theta-\theta_0)}{C_2+(\theta-\theta_0)},$$

where θ_0 is the reference temperature at which the relaxation data are given; θ is the temperature of interest; and C_1, C_2 are calibration constants obtained at this temperature. If $\theta \le \theta_0 - C_2$, deformation changes will be elastic, based on the instantaneous moduli.

For more information on the WLF equation, see *Viscoelasticity*.

Input File Usage: **TRS*, DEFINITION=WLF

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Viscoelastic**: **Domain: Time**, **Time:** ***any method***, and **Suboptions**->**Trs**: **Shift function: WLF**

Arrhenius Form

The Arrhenius shift function is commonly used for semicrystalline polymers. It takes the form

$$\ln(A) = \frac{E_0}{R}\left(\frac{1}{\theta-\theta^Z} - \frac{1}{\theta_0-\theta^Z}\right),$$

where E_0 is the activation energy, R is the universal gas constant, θ^Z is the absolute zero in the temperature scale being used, θ_0 is the reference temperature at which the relaxation data are given, and θ is the temperature of interest.

Input File Usage: Use the following option to define the Arrhenius shift function:

**TRS*, DEFINITION=ARRHENIUS

In addition, use the **PHYSICAL CONSTANTS* option to specify the universal gas constant and absolute zero.

Abaqus/CAE Usage: The Arrhenius shift function is not supported in Abaqus/CAE.

Tabular Form

You can specify the shift function in tabular form. In this case you provide the logarithm of base 10 of the shift function, $\log_{10}(A)$, as a function of temperature and field variables.

Input File Usage: **TRS*, DEFINITION=TABULAR, DEPENDENCIES=*n*

Abaqus/CAE Usage: Specifying the shift function in tabular form is not supported in Abaqus/CAE.

User-Defined Form

The shift function can be specified alternatively in user subroutines *UTRS* in Abaqus/Standard and *VUTRS* in Abaqus/Explicit.

Input File Usage: **TRS*, DEFINITION=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Elasticity**->**Viscoelastic**: **Domain: Time**, **Time:** ***any method***, and **Suboptions**->**Trs**: **Shift function: User subroutine UTRS**

Determination of Nonisotropic Viscoelastic Material Parameters

For nonisotropic viscoelasticity, the relaxation parameters can only be specified directly. The Prony series parameters $\bar{r}^P_{i,mnkl}$, and τ_i must be specified directly for each term in the Prony series. There is no restriction on the number of terms that can be used. If a relaxation time is associated with only some of the components, leave the others blank or enter zeros. The number of lines of data given defines the number of terms in the Prony series, *N*.

Only transversely isotropic and orthotropic viscoelastic behaviors are supported.

Input File Usage: Use the following to specify Prony series parameters for time domain viscoelasticity that is transversely isotropic:

**VISCOELASTIC*, TIME=PRONY,TYPE=TRANSVERSELY ISOTROPIC

Repeat the set of data lines to define each term in the Prony series.

Use the following to specify Prony series parameters for time domain viscoelasticity that is orthotropic:

**VISCOELASTIC*, TIME=PRONY,TYPE=ORTHOTROPIC

Repeat the set of data lines to define each term in the Prony series.

Abaqus/CAE Usage: Direct specification of Prony series parameters for generalized small-strain elasticity is not supported in Abaqus/CAE.

Defining the Rate-Independent Part of the Material Response

In all cases elastic moduli must be specified to define the rate-independent part of the material behavior. Small-strain linear elastic behavior is defined by an elastic material model (*Linear Elastic Behavior*), and large-deformation behavior is defined by a hyperelastic (*Hyperelastic Behavior of Rubberlike Materials*) or hyperfoam (*Hyperelastic Behavior in Elastomeric Foams*) material model. The rate-independent elasticity for any of these models can be defined in terms of either instantaneous elastic moduli or long-term elastic moduli. The choice of defining the elasticity in terms of instantaneous or long-term moduli is a matter of convenience only; it does not have an effect on the solution.

The effective relaxation moduli are obtained by multiplying the instantaneous elastic moduli with the dimensionless relaxation functions as described below.

Linear Elastic Isotropic Materials

For linear elastic isotropic material behavior

$$G_R(t) = G_0\left(1 - \sum_{k=1}^{N} \bar{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right)$$

and

$$\mathbf{K}_R(t) = \mathbf{K}_0\left(1 - \sum_{k=1}^{N} \bar{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

where G_0 and $\mathbf{K_0}$ are the instantaneous shear and bulk moduli determined from the values of the user-defined instantaneous elastic moduli E_0 and ν_0: $G_0 = E_0/2\left(1+\nu_0\right)$ and $\mathbf{K}_0 = E_0/3\left(1-2\nu_0\right)$.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$G_\infty = G_0\left(1-\sum_{k=1}^{N}\overline{g}_k^P\right) \qquad \text{and} \qquad K_\infty = K_0\left(1-\sum_{k=1}^{N}\overline{k}_k^P\right).$$

Linear Elastic Anisotropic Materials with Isotropic Viscoelastic Behaviors

For linear elastic anisotropic material with isotropic viscoelastic behavior, the relaxation coefficients are applied to the elastic moduli as

$$\overline{\mathbf{D}}_R\left(t\right) = \overline{\mathbf{D}}_0\left(1-\sum_{k=1}^{N}\overline{g}_k^P\left(1-e^{-t/\tau_k}\right)\right)$$

and

$$\mathbf{K}_R\left(t\right) = \mathbf{K}_0\left(1-\sum_{k=1}^{N}\overline{k}_k^P\left(1-e^{-t/\tau_k}\right)\right),$$

where $\overline{\mathbf{D}}_0$ and $\mathbf{K_0}$ are the instantaneous deviatoric elasticity tensor and bulk moduli determined from the values of the user-defined instantaneous elastic moduli $\mathbf{D_0}$. If both shear and bulk relaxation coefficients are specified and they are unequal, Abaqus issues an error message if the elastic moduli $\mathbf{D_0}$ is such that the deviatoric and volumetric response is coupled.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\overline{\mathbf{D}}_\infty = \overline{\mathbf{D}}_0\left(1-\sum_{k=1}^{N}\overline{g}_k^P\right) \qquad \text{and} \qquad K_\infty = K_0\left(1-\sum_{k=1}^{N}\overline{k}_k^P\right).$$

Linear Elastic Materials with Nonisotropic Viscoelastic Behaviors

For linear elastic material with nonisotropic viscoelastic behaviors, the relaxation coefficients are applied to the elastic moduli as

$$D_{R,ijmn}\left(t\right) = D_{0,ijmn}\left(1-\sum_{k=1}^{N}\overline{r}_{k,ijmn}^P\left(1-e^{-t/\tau_k}\right)\right)$$

where $\mathbf{D_0}$ is the instantaneous elastic moduli. There is no summation over repeated indices $ijmn$.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$D_{\infty,ijmn} = D_{0,ijmn}\left(1-\sum_{k=1}^{N}\overline{r}_{k,ijmn}^P\right).$$

Hyperelastic Materials

For hyperelastic material behavior the relaxation coefficients are applied either to the constants that define the energy function or directly to the energy function. For the polynomial function and its particular cases (reduced polynomial, Mooney-Rivlin, neo-Hookean, and Yeoh)

$$C_{ij}^R\left(t\right) = C_{ij}^0\left(1-\sum_{k=1}^{N}\overline{g}_k^P\left(1-e^{-t/\tau_k}\right)\right),$$

for the Ogden function

$$\mu_i^R(t) = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

for the Arruda-Boyce and Van der Waals functions

$$\mu^R(t) = \mu^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

and for the Marlow function

$$U_{dev}^R(t) = U_{dev}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right).$$

For the coefficients governing the compressible behavior of the polynomial models and the Ogden model

$$D_i^R(t) = D_i^0 / \left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

for the Arruda-Boyce and Van der Waals functions

$$D^R(t) = D^0 / \left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

and for the Marlow function

$$U_{vol}^R(t) = U_{vol}^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right).$$

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$C_{ij}^{\infty} = C_{ij}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right), \quad \text{or} \quad \mu_i^{\infty} = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right), \quad \text{or} \quad \mu^{\infty} = \mu^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right),$$

while the instantaneous bulk compliance moduli are obtained from

$$D_i^{\infty} = D_i^0 / \left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right), \quad \text{or} \quad D^{\infty} = D^0 / \left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right);$$

for the Marlow functions we have

$$U_{dev}^{\infty} = U_{dev}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right) \quad \text{and} \quad U_{vol}^{\infty} = U_{vol}^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right).$$

Mullins Effect

If long-term moduli are defined for the underlying hyperelastic behavior, the instantaneous value of the parameter m in Mullins effect is determined from

$$m^{\infty} = m^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right).$$

Elastomeric Foams

For elastomeric foam material behavior the instantaneous shear and bulk relaxation coefficients are assumed to be equal and are applied to the material constants μ_i in the energy function:

$$\mu_i^R(t) = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right) = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right).$$

If only the shear relaxation coefficients are specified, the bulk relaxation coefficients are set equal to the shear relaxation coefficients and vice versa. If both shear and bulk relaxation coefficients are specified and they are unequal, Abaqus issues an error message.

If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\mu_i^\infty = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right) = \mu_i^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right).$$

Traction-Separation Elasticity

For cohesive elements with uncoupled traction-separation elastic behavior:

$$K_{nn}(t) = K_{nn}^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

$$K_{ss}(t) = K_{ss}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

and

$$K_{tt}(t) = K_{tt}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

where K_{nn}^0 is the instantaneous normal modulus and K_{ss}^0 and K_{tt}^0 are the instantaneous shear moduli. If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$K_{nn}^\infty/K_{nn}^0 = \left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right), \qquad \text{and} \qquad K_{ss}^\infty/K_{ss}^0 = K_{tt}^\infty/K_{tt}^0 = \left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right).$$

For cohesive elements with coupled traction-separation elastic behavior the shear and bulk relaxation coefficients must be equal:

$$\mathbf{K}(t) = \mathbf{K}^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\left(1 - e^{-t/\tau_k}\right)\right) = \mathbf{K}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\left(1 - e^{-t/\tau_k}\right)\right),$$

where $\mathbf{K}^0$ is the user-defined instantaneous elasticity matrix. If long-term elastic moduli are defined, the instantaneous moduli are determined from

$$\mathbf{K}^\infty = \mathbf{K}^0\left(1 - \sum_{k=1}^{N} \overline{k}_k^P\right) = \mathbf{K}^0\left(1 - \sum_{k=1}^{N} \overline{g}_k^P\right).$$

Material Response in Different Analysis Procedures

The time-domain viscoelastic material model is active during the following procedures:

- transient static analysis (*Quasi-Static Analysis*),
- transient implicit dynamic analysis (*Implicit Dynamic Analysis Using Direct Integration*),
- explicit dynamic analysis (*Explicit Dynamic Analysis*),
- steady-state transport analysis (*Steady-State Transport Analysis*),
- fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*),

- fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*), and
- transient (consolidation) coupled pore fluid diffusion and stress analysis (*Coupled Pore Fluid Diffusion and Stress Analysis*).

Viscoelastic material response is always ignored in a static analysis. It can also be ignored in a coupled temperature-displacement analysis, a coupled thermal-electrical-structural analysis, or a soils consolidation analysis by specifying that no creep or viscoelastic response is occurring during the step even if creep or viscoelastic material properties are defined (see *Fully Coupled Thermal-Stress Analysis* or *Coupled Pore Fluid Diffusion and Stress Analysis*). In these cases, it is assumed that the loading is applied instantaneously, so that the resulting response corresponds to an elastic solution based on instantaneous elastic moduli.

Abaqus/Standard provides the option to use the fully relaxed long-term elastic solution directly in a static or steady-state transport analysis without having to perform a transient analysis (see *Rate-Dependent Yield and Friction*). The viscous damping stresses (the internal stresses associated with each of the Prony-series terms) are increased gradually from their values at the beginning of the step to their long-term values at the end of the step if the long-term value is specified.

If a time-domain viscoelastic response is specified for a frequency domain procedure, the viscoelastic response is ignored and the material response is based on the instantaneous elastic moduli.

In Abaqus/Explicit, you can use the fully relaxed long-term elastic solution directly in an explicit dynamic or in an explicit dynamic coupled thermal stress analysis (see *Controlling Material Rate Dependence* and *Controlling Material and Friction Rate Dependence*).

Material Options

The viscoelastic material model must be combined with an elastic material model. It is used with the isotropic, transversely isotropic, orthotropic, or anisotropic linear elasticity model (*Linear Elastic Behavior*) to define classical, linear, small-strain, viscoelastic behavior or with the hyperelastic (*Hyperelastic Behavior of Rubberlike Materials*) or hyperfoam (*Hyperelastic Behavior in Elastomeric Foams*) models to define large-deformation, nonlinear, viscoelastic behavior. It can also be used with anisotropic linear elasticity and with traction-separation elastic behavior in Abaqus/Explicit. The elastic properties defined for these models can be temperature dependent.

Viscoelasticity cannot be combined with any of the plasticity models. See *Combining Material Behaviors* for more details.

Elements

The time domain viscoelastic material model can be used with any stress/displacement, coupled temperature-displacement, or thermal-electrical-structural element in Abaqus, except when you specify transversely isotropic, orthotropic, or anisotropic linear behaviors in Abaqus/Standard. In this case, Abaqus/Standard supports only three-dimensional, plane strain, and axisymmetric continuum stress/displacement elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning in Abaqus/Standard if viscoelasticity is defined:

EE Elastic strain corresponding to the stress state at time t and the instantaneous elastic material properties.

CE Equivalent creep strain defined as the difference between the total strain and the elastic strain.

Considerations for Steady-State Transport Analysis

When a steady-state transport analysis (*Steady-State Transport Analysis*) is combined with large-strain viscoelasticity, the viscous dissipation, CENER, is computed as the energy dissipated per revolution as a material point is transported around its streamline; that is,

$$W_{cr} = \oint \boldsymbol{\sigma} : d\boldsymbol{\varepsilon}.$$

Consequently, all the material points in a given streamline report the same value for CENER, and other derived quantities such as ELCD and ALLCD also have the meaning of dissipation per revolution. The recoverable elastic strain energy density, SENER, is approximated as

$$W_{el} = W_{el}^{t} + W_{cr}^{t} + \Delta W - W_{cr},$$

where ΔW is the incremental energy input and t is the time at the beginning of the current increment. Since two different units are used in the quantities appearing in the above equation, a proper meaning cannot be assigned to quantities such as SENER, ELSE, ALLSE, and ALLIE.

Considerations for Large-Strain Viscoelasticity

In Abaqus/Standard the viscous energy dissipated is computed only approximately for large-strain viscoelasticity.

Abaqus/Explicit does not compute the viscous dissipation for performance reasons for the case of large-strain viscoelasticity. Instead, the contribution of viscous dissipation is included in the strain energy output, SENER; and CENER is output as zero. Consequently, special care must be exercised when interpreting strain energy results of large-strain viscoelastic materials in Abaqus/Explicit since they include viscous dissipation effects.

Frequency Domain Viscoelasticity

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- *Time Domain Viscoelasticity*
- **VISCOELASTIC*
- *Defining frequency domain viscoelasticity*

Overview

The frequency domain viscoelastic material model:

- describes frequency-dependent material behavior in small steady-state harmonic oscillations for those materials in which dissipative losses caused by "viscous" (internal damping) effects must be modeled in the frequency domain;
- can be used to describe isotropic, transversely isotropic, and orthotropic dissipative losses in the frequency domain;
- can be used in large-strain problems;
- can be used only with *Linear Elastic Behavior*, *Hyperelastic Behavior of Rubberlike Materials*, and *Hyperelastic Behavior in Elastomeric Foams* to define the long-term elastic material properties;
- can be used with the elastic-damage gasket behavior (*Defining a Nonlinear Elastic Model with Damage*) to define the effective thickness-direction storage and loss moduli for gasket elements; and
- is active only during the direct-solution steady-state dynamic (*Direct-Solution Steady-State Dynamic Analysis*), the subspace-based steady-state dynamic (*Subspace-Based Steady-State Dynamic Analysis*), the natural frequency extraction (*Natural Frequency Extraction*), and the complex eigenvalue extraction (*Complex Eigenvalue Extraction*) procedures.

Defining Isotropic Viscoelasticity

For isotropic frequency domain viscoelasticity, Abaqus assumes that the shear (deviatoric) and volumetric behaviors are independent in multiaxial stress states.

Defining the Shear Behavior

Consider a shear test at small strain, in which a harmonically varying shear strain γ is applied:

$$\gamma(t) = \gamma_0 \exp(i\omega t),$$

where γ_0 is the amplitude, $i = \sqrt{-1}$, ω is the circular frequency, and t is time. We assume that the specimen has been oscillating for a very long time so that a steady-state solution is obtained. The solution for the shear stress then has the form

$$\tau(t) = \left(G_s(\omega) + iG_\ell(\omega)\right)\gamma_0 \exp(i\omega t),$$

where G_s and G_ℓ are the shear storage and loss moduli. These moduli can be expressed in terms of the (complex) Fourier transform $g^*(\omega)$ of the nondimensional shear relaxation function $g(t) = \frac{G_R(t)}{G_\infty} - 1$:

$$G_s(\omega) = G_\infty(1 - \omega\mathcal{I}(g^*)), \qquad G_\ell(\omega) = G_\infty(\omega\mathcal{R}(g^*)),$$

where $G_R(t)$ is the time-dependent shear relaxation modulus, $\mathcal{R}(g^*)$ and $\mathcal{I}(g^*)$ are the real and imaginary parts of $g^*(\omega)$, and G_∞ is the long-term shear modulus. See *Frequency domain viscoelasticity* for details.

The above equation states that the material responds to steady-state harmonic strain with a stress of magnitude $G_s\gamma_0$ that is in phase with the strain and a stress of magnitude $G_\ell\gamma_0$ that lags the excitation by $90°$. Therefore, we can regard the factor

$$G^*(\omega) = G_s(\omega) + iG_\ell(\omega)$$

as the complex, frequency-dependent shear modulus of the steadily vibrating material. The absolute magnitude of the stress response is

$$\left|\tau\right| = \sqrt{G_s^2(\omega) + G_\ell^2(\omega)}\ \left|\gamma_0\right|,$$

and the phase lag of the stress response is

$$\phi = \arctan\left(\frac{G_\ell(\omega)}{G_s(\omega)}\right).$$

Measurements of $|\tau|$ and ϕ as functions of frequency in an experiment can, therefore, be used to define G_s and G_ℓ and, thus, $\mathcal{R}(g^*)$ and $\mathcal{I}(g^*)$ as functions of frequency.

Unless stated otherwise explicitly, all modulus measurements are assumed to be “true” quantities.

Defining the Volumetric Behavior

For isotropic viscoelasticity, the frequency dependence of the shear (deviatoric) and volumetric behaviors are independent. The volumetric behavior is defined by the bulk storage and loss moduli $\mathbf{K}_s(\omega)$ and $\mathbf{K}_\ell(\omega)$. Similar to the shear moduli, these moduli can also be expressed in terms of the (complex) Fourier transform $k^*(\omega)$ of the nondimensional bulk relaxation function $k(t)$:

$$K_s(w) = K_\infty(1 - \omega\mathcal{I}(k^*)), \qquad K_\ell(w) = K_\infty(\omega\mathcal{R}(k^*)),$$

where $\mathbf{K}_\infty$ is the long-term elastic bulk modulus.

Large-Strain Viscoelasticity

The linearized vibrations can also be associated with an elastomeric material whose long-term (elastic) response is nonlinear and involves finite strains (a hyperelastic material). We can retain the simplicity of the steady-state small-amplitude vibration response analysis in this case by assuming that the linear expression for the shear stress still governs the system, except that now the long-term shear modulus G_∞ can vary with the amount of static prestrain $\overline{\gamma}$:

$$G_\infty = G_\infty(\overline{\gamma}).$$

The essential simplification implied by this assumption is that the frequency-dependent part of the material's response, defined by the Fourier transform $g^*(\omega)$ of the relaxation function, is not affected by the magnitude of

the prestrain. Therefore, strain and frequency effects are separated, which is a reasonable approximation for many materials.

Another implication of the above assumption is that the anisotropy of the viscoelastic moduli has the same strain dependence as the anisotropy of the long-term elastic moduli. Therefore, the viscoelastic behavior in all deformed states can be characterized by measuring the (isotropic) viscoelastic moduli in the undeformed state.

In situations where the above assumptions are not reasonable, the data can be specified based on measurements at the prestrain level about which the steady-state dynamic response is desired. In this case you must measure G_s, G_ℓ, and G_∞ (likewise K_s, K_ℓ, and K_∞) at the prestrain level of interest. Alternatively, the viscoelastic data can be given directly in terms of uniaxial and volumetric storage and loss moduli that might be specified as functions of frequency and prestrain (see *Direct Specification of Storage and Loss Moduli for Large-Strain Isotropic Viscoelasticity* below.)

The generalization of these concepts to arbitrary three-dimensional deformations is provided in Abaqus/Standard by assuming that the frequency-dependent material behavior has two independent components: one associated with shear (deviatoric) straining and the other associated with volumetric straining. In the general case of a compressible material, the model is, therefore, defined for kinematically small perturbations about a predeformed state as

$$\frac{1}{J}\Delta^{\nabla}(J\mathbf{S}) = (1+i\omega g^*)\,\mathbf{C}^S\big|_0 : \Delta\mathbf{e} + \mathbf{Q}\big|_0 \Delta\varepsilon^{\text{vol}},$$

and

$$\Delta p = -\mathbf{Q}\big|_0 : \Delta\mathbf{e} - (1+i\omega k^*)\,K\big|_0 \Delta\varepsilon^{\text{vol}},$$

where

$\mathbf{S}$

is the deviatoric stress, $\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$;

p

is the equivalent pressure stress, $p = -\frac{1}{3}\,\text{trace}\,(\boldsymbol{\sigma})$;

$\Delta^{\nabla}(J\mathbf{S})$

is the part of the stress increment caused by incremental straining (as distinct from the part of the stress increment caused by incremental rotation of the preexisting stress with respect to the coordinate system);

J

is the ratio of current to original volume;

$\Delta\mathbf{e}$

is the (small) incremental deviatoric strain, $\Delta\mathbf{e} = \Delta\boldsymbol{\varepsilon} - \frac{1}{3}\Delta\varepsilon^{\text{vol}}\,\mathbf{I}$;

$\dot{\mathbf{e}}$

is the deviatoric strain rate, $\dot{\mathbf{e}} = \dot{\boldsymbol{\varepsilon}} - \frac{1}{3}\dot{\varepsilon}^{\text{vol}}\,\mathbf{I}$;

$\Delta\varepsilon^{\text{vol}}$

is the (small) incremental volumetric strain, $\Delta\varepsilon^{\text{vol}} = \text{trace}\,(\Delta\boldsymbol{\varepsilon})$;

$\dot{\varepsilon}^{\text{vol}}$

is the rate of volumetric strain, $\dot{\varepsilon}^{\text{vol}} = \text{trace}\,(\dot{\boldsymbol{\varepsilon}})$;

$\mathbf{C}^{S}|_{0}$

is the deviatoric tangent elasticity matrix of the material in its predeformed state (for example, C_{1212} is the tangent shear modulus of the prestrained material);

$\mathbf{Q}|_{0}$

is the volumetric strain-rate/deviatoric stress-rate tangent elasticity matrix of the material in its predeformed state; and

$K|_{0}$

is the tangent bulk modulus of the predeformed material.

For a fully incompressible material, only the deviatoric terms in the first constitutive equation above remain and the viscoelastic behavior is completely defined by $g^{*}(\omega)$.

Defining Nonisotropic Viscoelasticity

For nonisotropic frequency domain viscoelasticity, the viscoelastic behavior is defined by the storage and loss moduli $\mathbf{D}_{s}(\omega)$ and $\mathbf{D}_{\ell}(\omega)$. Similar to the shear and bulk moduli defined for isotropic viscoelasticity, the components of the elastic moduli can also be expressed in terms of the (complex) Fourier transform $D_{ijkl}^{*}(\omega)$ of the nondimensional relaxation function $D_{ijkl}(t)$:

$$D_{s,ijkl}(w) = D_{\infty,ijkl}\left(1 - \omega\,\Im(D_{ijkl}^{*})\right), \qquad D_{\ell,ijkl}(w) = D_{\infty,ijkl}\left(\omega\,\Re(D_{ijkl}^{*})\right),$$

where $D_{\infty,ijkl}$ is the $ijkl$ component of the long-term elastic modulus $\mathbf{D}_{\infty}$.

You can use nonisotropic viscoelasticity only with the small-strain formulation.

Determination of Isotropic Viscoelastic Material Parameters

When the dissipative part of the material behavior is isotropic, it can be defined by giving the real and imaginary parts of g^{*} and k^{*} (for compressible materials) as functions of frequency. The moduli can be defined as functions of the frequency in one of three ways: by a power law, by tabular input, or by a Prony series expression for the shear and bulk relaxation moduli.

Power Law Frequency Dependence

The frequency dependence can be defined by the power law formulas

$$g^*(\omega) = g_1^* f^{-a} \qquad \text{and} \qquad k^*(\omega) = k_1^* f^{-b},$$

where a and b are real constants, g_1^* and k_1^* are complex constants, and $f = \omega/2\pi$ is the frequency in cycles per time.

Input File Usage: **VISCOELASTIC*, FREQUENCY=FORMULA

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Frequency** and **Frequency: Formula**

Tabular Frequency Dependence

The frequency domain response can alternatively be defined in tabular form by giving the real and imaginary parts of ωg^* and ωk^*—where ω is the circular frequency—as functions of frequency in cycles per time. Given the frequency-dependent storage and loss moduli $G_s(\omega)$, $G_\ell(\omega)$, $\mathbf{K}_s(\omega)$, and $\mathbf{K}_\ell(\omega)$, the real and imaginary parts of ωg^* and ωk^* are then given as

$$\omega\Re(g^*) = G_\ell/G_\infty, \qquad \omega\Im(g^*) = 1 - G_s/G_\infty, \quad \omega\Re(k^*) = \mathbf{K}_\ell/\mathbf{K}_\infty, \qquad \omega\Im(k^*) = 1 - \mathbf{K}_s/\mathbf{K}_\infty,$$

where G_∞ and $\mathbf{K}_\infty$ are the long-term shear and bulk moduli determined from the elastic or hyperelastic properties.

Abaqus provides an alternative approach for specifying the viscoelastic properties of hyperelastic and hyperfoam materials. This approach involves the direct (tabular) specification of storage and loss moduli from uniaxial and volumetric tests, as functions of excitation frequency and a measure of the level of prestrain. The level of prestrain refers to the level of elastic deformation at the base state about which the steady-state harmonic response is desired. This approach is discussed in *Direct Specification of Storage and Loss Moduli for Large-Strain Isotropic Viscoelasticity* below.

Input File Usage: **VISCOELASTIC*, FREQUENCY=TABULAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Frequency** and **Frequency: Tabular**

Prony Series Parameters

The frequency dependence can also be obtained from a time domain Prony series description of the dimensionless shear and bulk relaxation moduli:

$$g_R(t) = 1 - \sum_{i=1}^{N} \bar{g}_i^P \left(1 - e^{-t/\tau_i}\right),$$

$$k_R(t) = 1 - \sum_{i=1}^{N} \bar{k}_i^P \left(1 - e^{-t/\tau_i}\right),$$

where N, $\bar{g}_i^P$, $\bar{k}_i^P$, and τ_i, $i = 1, 2, \ldots, N$, are material constants. Using Fourier transforms, the expression for the time-dependent shear modulus can be written in the frequency domain as follows:

$$G_s(\omega) = G_0 \left[1 - \sum_{i=1}^{N} \bar{g}_i^P\right] + G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i^2 \omega^2}{1+\tau_i^2\omega^2},$$

$$G_\ell(\omega) = G_0 \sum_{i=1}^{N} \frac{\bar{g}_i^P \tau_i \omega}{1+\tau_i^2\omega^2},$$

where $G_s(\omega)$ is the storage modulus, $G_\ell(\omega)$ is the loss modulus, ω is the angular frequency, and N is the number of terms in the Prony series. The expressions for the bulk moduli, $K_s(\omega)$ and $K_\ell(\omega)$, are written analogously. Abaqus/Standard automatically performs the conversion from the time domain to the frequency domain. The

Prony series parameters $\overline{g}_i^P, \overline{k}_i^P, \tau_i$ can be defined in one of three ways: direct specification of the Prony series parameters, inclusion of creep test data, or inclusion of relaxation test data. If you specify creep test data or relaxation test data, Abaqus/Standard determines the Prony series parameters in a nonlinear least-squares fit. A detailed description of the calibration of Prony series terms is provided in *Time Domain Viscoelasticity*.

For the test data, you can specify the normalized shear and bulk data separately as functions of time or specify the normalized shear and bulk data simultaneously. A nonlinear least-squares fit is performed to determine the

Prony series parameters, $\left(\overline{g}_i^P, \overline{k}_i^P, \tau_i\right)$.

Input File Usage: Use one of the following options to specify Prony data, creep test data, or relaxation test data:

**VISCOELASTIC*, FREQUENCY=PRONY
**VISCOELASTIC*, FREQUENCY=CREEP TEST DATA
**VISCOELASTIC*, FREQUENCY=RELAXATION TEST DATA

Use one or both of the following options to specify the normalized shear and bulk data separately as functions of time:

**SHEAR TEST DATA*
**VOLUMETRIC TEST DATA*

Use the following option to specify the normalized shear and bulk data simultaneously:

**COMBINED TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Frequency** and **Frequency: Prony**, **Creep test data**, or **Relaxation test data**

Use one or both of the following options to specify the normalized shear and bulk data separately as functions of time:

Test Data->Shear Test Data
Test Data->Volumetric Test Data

Use the following option to specify the normalized shear and bulk data simultaneously:

Test Data->Combined Test Data

Thermorheologically Simple Temperature Effects in Frequency Domain Viscoelasticity

You can include thermorheologically simple temperature effects in frequency domain viscoelasticity. In this case the reduced angular frequency, ω_r, is used to obtain the frequency-dependent material moduli. The reduced angular frequency is computed as

$$\omega_r = A(\theta)\,\omega,$$

where $A(\theta)$ and θ denote the shift function and temperature, respectively. Abaqus/Standard supports the following forms of the shift function: the Williams-Landel-Ferry (WLF) form, the Arrhenius form, the tabular form, and user-defined forms (see *Thermorheologically Simple Temperature Effects*).

Determination of Nonisotropic Viscoelastic Material Parameters

When the dissipative part of the material behavior is nonisotropic, it can be defined by giving the real and imaginary parts of $\mathbf{D}^*$ as a function of frequency. The moduli can be defined as function of the frequency by a Prony series expression for the relaxation moduli.

The frequency dependence can be obtained from a time domain Prony series description of the $mnkl$ component of the dimensionless relaxation moduli:

$$r_{R,(mnkl)}(t) = 1 - \sum_{i=1}^{N} \bar{r}^{P}_{i,(mnkl)} \left(1 - e^{-t/\tau_i}\right),$$

where N, $\bar{r}^{P}_{i,(mnkl)}$; and τ_i, $i = 1, 2, \ldots, N$, are material constants. Abaqus assumes that same τ_i are used for all components. Using Fourier transforms, the expression for the time-dependent modulus can be written in the frequency domain as follows:

$$D_{s,mnkl}(\omega) = D_{0,mnkl}\left[1 - \sum_{i=1}^{N} \bar{r}^{P}_{i,(mnkl)}\right] + D_{0,mnkl} \sum_{i=1}^{N} \frac{\bar{r}^{P}_{i,(mnkl)} \tau_i^2 \omega^2}{1+\tau_i^2\omega^2},$$

$$D_{l,mnkl}(\omega) = D_{0,mnkl} \sum_{i=1}^{N} \frac{\bar{r}^{P}_{i,(mnkl)} \tau_i \omega}{1+\tau_i^2\omega^2},$$

where

$D_{s,mnkl}(\omega)$
: is the $mnkl$ component of the storage modulus;

$D_{l,mnkl}(\omega)$
: is the $mnkl$ component of the loss modulus;

ω
: is the angular frequency; and

N
: is the number of terms in the Prony series. Abaqus/Standard automatically performs the conversion from the time domain to the frequency domain.

You can specify the Prony series parameters $\bar{r}^{P}_{i,(mnkl)}$, τ_i only by defining them directly.

You can include thermorheologically simple temperature effects in frequency domain nonisotropic viscoelasticity. Abaqus applies thermorheologically simple temperature effects the same way for nonisotropic viscoelasticity as for isotropic viscoelasticity.

Input File Usage: Use the following option to specify Prony data for frequency domain viscoelasticity that is transversely isotropic:

**VISCOELASTIC*, FREQUENCY=PRONY, TYPE=TRANSVERSELY ISOTROPIC

Use the following option to specify Prony data for frequency domain viscoelasticity that is orthotropic:

*VISCOELASTIC, FREQUENCY=PRONY, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Defining Prony data for components of the dimensionless elastic moduli is not supported in Abaqus/CAE.

Conversion of Frequency-Dependent Elastic Moduli for Isotropic Viscoelasticity

For some cases of small straining of isotropic viscoelastic materials, the material data are provided as frequency-dependent uniaxial storage and loss moduli, $E_s(\omega)$ and $E_\ell(\omega)$, and bulk moduli, $\mathrm{K}_s(\omega)$ and $\mathrm{K}_\ell(\omega)$. In that case the data must be converted to obtain the frequency-dependent shear storage and loss moduli $G_s(\omega)$ and $G_\ell(\omega)$.

The complex shear modulus is obtained as a function of the complex uniaxial and bulk moduli with the expression

$$G^* = \frac{3\mathrm{K}^* E^*}{9\mathrm{K}^* - E^*}.$$

Replacing the complex moduli by the appropriate storage and loss moduli, this expression transforms into

$$G_s + iG_\ell = \frac{3(\mathrm{K}_s + i\mathrm{K}_\ell)(E_s + iE_\ell)}{9(\mathrm{K}_s + i\mathrm{K}_\ell) - (E_s + iE_\ell)}.$$

After some algebra one obtains

$$G_s = 3\frac{9E_s(\mathrm{K}_s^2 + \mathrm{K}_\ell^2) - \mathrm{K}_s(E_s^2 + E_\ell^2)}{(9\mathrm{K}_s - E_s)^2 + (9\mathrm{K}_\ell - E_\ell)^2}, \quad G_\ell = 3\frac{9E_\ell(\mathrm{K}_s^2 + \mathrm{K}_\ell^2) - \mathrm{K}_\ell(E_s^2 + E_\ell^2)}{(9\mathrm{K}_s - E_s)^2 + (9\mathrm{K}_\ell - E_\ell)^2}.$$

Shear Strain Only

In many cases the viscous behavior is associated only with deviatoric straining, so that the bulk modulus is real and constant: $\mathrm{K}_s = \mathrm{K}_\infty$ and $\mathrm{K}_\ell = 0$. For this case, the expressions for the shear moduli simplify to

$$G_s = 3\mathrm{K}_\infty \frac{9E_s\mathrm{K}_\infty - E_s^2 - E_\ell^2}{(9\mathrm{K}_\infty - E_s)^2 + E_\ell^2}, \qquad G_\ell = 3\mathrm{K}_\infty \frac{9E_\ell\mathrm{K}_\infty}{(9\mathrm{K}_\infty - E_s)^2 + E_\ell^2}.$$

Incompressible Materials

If the bulk modulus is very large compared to the shear modulus, the material can be considered to be incompressible, and the expressions simplify further to

$$G_s = E_s/3, \qquad G_\ell = E_\ell/3.$$

Direct Specification of Storage and Loss Moduli for Large-Strain Isotropic Viscoelasticity

For large-strain viscoelasticity Abaqus allows direct specification of storage and loss moduli from uniaxial and volumetric tests. This approach can be used when the assumption of the independence of viscoelastic properties on the prestrain level is too restrictive.

You specify the storage and loss moduli directly as tabular functions of frequency, and you specify the level of prestrain at the base state about which the steady-state dynamic response is desired. For uniaxial test data the measure of prestrain is the uniaxial nominal strain; for volumetric test data the measure of prestrain is the volume ratio. Abaqus internally converts the data that you specify to ratios of shear/bulk storage and loss moduli to the corresponding long-term elastic moduli. Subsequently, the basic formulation described in *Large-Strain Viscoelasticity* above is used.

For a general three-dimensional stress state it is assumed that the deviatoric part of the viscoelastic response depends on the level of prestrain through the first invariant of the deviatoric left Cauchy-Green strain tensor (see *Hyperelastic material behavior* for a definition of this quantity), while the volumetric part depends on the prestrain through the volume ratio. A consequence of these assumptions is that for the uniaxial case, data can be specified from a uniaxial-tension preload state or from a uniaxial-compression preload state but not both.

The storage and loss moduli that you specify are assumed to be nominal quantities.

Input File Usage: Use the following option to specify only the uniaxial storage and loss moduli:

**VISCOELASTIC*, PRELOAD=UNIAXIAL

You can also use the following option to specify the volumetric (bulk) storage and loss moduli:

**VISCOELASTIC*, PRELOAD=VOLUMETRIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Viscoelastic**: **Domain: Frequency** and **Frequency: Tabular**

Use the following option to specify only the uniaxial storage and loss moduli:

Type: Isotropic or **Traction**: **Preload: Uniaxial**

Use the following option to specify only the volumetric storage and loss moduli:

Type: Isotropic: **Preload: Volumetric**

Use the following option to specify both uniaxial and volumetric moduli:

Type: Isotropic: **Preload: Uniaxial and Volumetric**

Defining the Rate-Independent Part of the Material Behavior

In all cases elastic moduli must be specified to define the rate-independent part of the material behavior. The elastic behavior is defined by an elastic, hyperelastic, or hyperfoam material model. Since the frequency domain viscoelastic material model is developed around the long-term elastic moduli, the rate-independent elasticity must be defined in terms of long-term elastic moduli. This implies that the response in any analysis procedure other than a direct-solution steady-state dynamic analysis (such as a static preloading analysis) corresponds to the fully relaxed long-term elastic solution.

Material Options

The viscoelastic material model must be combined with the isotropic, transversely isotropic, orthotropic (including engineering constants), or anisotropic linear elasticity model to define classical, linear, small-strain, viscoelastic behavior. It is combined with the hyperelastic or hyperfoam model to define large-deformation, nonlinear, viscoelastic behavior. The long-term elastic properties defined for these models can be temperature dependent.

Viscoelasticity cannot be combined with any of the plasticity models. See *Combining Material Behaviors* for more details.

Defining the Model with Nonisotropic Linear Elastic Behavior

Abaqus supports the combination of frequency domain viscoelasticity with nonisotropic linear elasticity.

When isotropic viscoelasticity is defined, if the specified shear (deviatoric) and volumetric viscoelastic behaviors are different, Abaqus must split the stiffness matrix into the corresponding deviatoric and volumetric parts to account for the frequency dependence of the material. If the stiffness matrix is defined in a manner such that it cannot be separated into purely deviatoric and volumetric parts, the analysis ends with an error. This restriction does not apply when the specified shear and volumetric viscoelastic behavior are identical because the stiffness matrix does not have to be split in this case.

Transversely isotropic viscoelasticity can only be combined with transversely isotropic linear elasticity model, and orthotropic viscoelasticity can only be combined with orthotropic linear elasticity model.

Elements

You can use the isotropic frequency domain viscoelastic material model with any stress/displacement element in Abaqus/Standard. When nonisotropic viscoelasticity is defined, Abaqus/Standard supports only three-dimensional, plane strain, and axisymmetric continuum stress/displacement elements.

Structural Relaxation in Glass

Products: Abaqus/Standard

References:

- **EXPANSION*
- **TRS*

Overview

The structural relaxation model:

- can be isotropic, orthotropic, or fully anisotropic; and
- uses fictive temperature to describe the structure of glass.

Structural Relaxation

At high temperatures glass behaves like a liquid, while at low temperatures glass behaves like a thermoelastic solid. In both temperature ranges the physical properties of glass depend only on the instantaneous value of the temperature. However, at temperatures in the vicinity of the glass transition temperature the behavior of glass is different. In this temperature range, the molecular structure of glass changes gradually with temperature and a noticeable delay is observed before the equilibrium state is reached. In this case the physical properties depend on both the temperature and the thermal history.

Tool-Narayanaswamy-Moynihan Model

In Abaqus/Standard the Tool-Narayanaswamy-Moynihan (TNM) model can be used to predict structural relaxation in glass. It uses the fictive temperature, θ_f, originally proposed by Tool to describe the structure of the material.

In the TNM model the thermal strains are obtained from the following relation:

$$\epsilon^{th} = \alpha_g(\theta, f_\beta)(\theta - \theta^0) + \left[\alpha_l(\theta, f_\beta) - \alpha_g(\theta, f_\beta)\right](\theta_f - \theta^0)$$
$$-\alpha_g(\theta^I, f_\beta^I)(\theta^I - \theta^0) - \left[\alpha_l(\theta^I, f_\beta^I) - \alpha_g(\theta^I, f_\beta^I)\right](\theta_f^I - \theta^0)$$

where

$\alpha_g(\theta, f_\beta)$

is the thermal expansion coefficient for the glassy state;

$\alpha_l(\theta, f_\beta)$

is the thermal expansion coefficient for the liquid state;

θ

is the current temperature;

θ^I

is the initial temperature;

θ_f^I

is the initial fictive temperature;

f_β

are the current values of the predefined field variables;

f_β^I

are the initial values of the predefined field variables; and

θ^0

is the reference temperature for the thermal expansion coefficients.

For a special case when the thermal expansion coefficients are not temperature- or field variable-dependent, it can be shown that the thermal strain increment can be expressed as

$$\Delta\epsilon^{th} = \alpha_g \Delta\theta + (\alpha_l - \alpha_g)\Delta\theta_f.$$

In the TNM model we must specify the coefficients of thermal expansion for both glassy and liquid states.

Input File Usage: Use the following option to define the thermal expansion coefficient for the glassy state:

**EXPANSION*

Use the following option to define the thermal expansion coefficient for the liquid state:

**EXPANSION*, LIQUID

Defining the Reference Temperature

If the coefficients of thermal expansion are not functions of temperature or field variables, the value of the reference temperature, θ^0, is not required. If α_g or α_l is a function of temperature or field variables, you can define θ^0.

Input File Usage: Use the following option to define the reference temperature value:

**EXPANSION*, ZERO = θ^0

Fictive Temperature

The fictive temperature is commonly expressed by introducing an equilibrium response function $M(t)$:

$$\theta_f\left(t\right) = \theta\left(t\right) - \int_0^t M\left(t - t'\right)\frac{d\theta}{dt'}dt'.$$

The response function, $M(t)$, is a decreasing function in time, such that $M(0) = 1$ and $M(\infty) = 0$.

In the Tool-Narayanaswamy-Moynihan model the response function is expressed with a series of exponential functions:

$$M\left(t\right) = \sum_{i=1}^{N} C_i e^{\frac{-t}{\tau_i}}$$

where

C_i

is the material parameter for the i^{th} term of the response function;

τ_i

is the relaxation time for the i^{th} term of the response function; and

N

is the number of terms in the response function.

In addition, the material parameters C_i must satisfy the relation $\sum_{i=1}^{N} C_i = 1$.

Combining the equations above, the expression for the fictive temperature has the form

$$\theta_f = \sum_{i=1}^{n} C_i \theta_{f,i}\left(\xi\left(t\right)\right),$$

where

$$\theta_{f,i} = \theta\left(t\right) - \int_0^t e^{-\frac{\xi(t)-\xi(t')}{\tau_i}}\frac{d\theta}{dt'}dt',$$

and $\xi = \int_0^t \frac{1}{A(t')}dt'$ is the reduced time, which is computed using the following shift function:

$$A = e^{\frac{E_0}{R}\left(\frac{\chi}{\theta-\theta^Z} + \frac{1-\chi}{\theta_f-\theta^Z} - \frac{1}{\theta_R-\theta^Z}\right)},$$

where

E_0

is the activation energy;

R

is the universal gas constant;

θ_R

is the reference temperature; and

θ^Z

is the absolute zero in the temperature scale used.

Input File Usage: Use the following option to define the response function:

**TRS*, DEFINITION=TNM

Computation of Fictive Temperature

The fictive temperature at time $t + \Delta t$ is computed using the algorithm proposed by Markovsky and Soules:

$$\theta_f\left(t+\Delta t\right) = \sum_{i=1}^{n} C_i \theta_{f,i}\left(t+\Delta t\right)$$

and

$$\theta_{f,i}\left(t+\Delta t\right) = \frac{\tau_i \theta_{f,i}(t) + \theta(t+\Delta t)\frac{d\xi}{dt}\Delta t}{\tau_i + \frac{d\xi}{dt}\Delta t}.$$

Elements

The structural relaxation model can be used with any stress/displacement element in Abaqus/Standard.

Procedures

The model can be used with all stress/displacement procedure types. However, the time effects are taken into account only in quasi-static (see *Quasi-Static Analysis*), coupled temperature-displacement (see *Fully Coupled Thermal-Stress Analysis*), and direct-integration implicit dynamic (see *Implicit Dynamic Analysis Using Direct Integration*) analyses. In other stress/displacement procedure types the evolution of the fictive temperature is suppressed, so its value remains unchanged.

Output

In addition to the standard output identifiers available in Abaqus/Standard(*Using Abaqus/Standard Output Variable Identifiers*), the following variable has special meaning for the TNM model:

TFICT	Fictive temperature.

References

- Markovsky, A., and T. F. Soules, “An Efficient and Stable Algorithm for Calculating Fictive Temperatures,” *Journal of the American Ceramic Society*, vol. 67, no. 4, pp. 56–57, 1984.
- Narayanaswamy, O. S., “A Model of Structural Relaxation in Glass,” *Journal of the American Ceramic Society*, vol. 54, no. 10, pp. 491–498, 1971.
- Tool, A. Q., “Relation Between Inelastic Deformation and Thermal Expansion of Glass in Its Annealing Range,” *Journal of the American Ceramic Society*, vol. 29, no. 9, pp. 240–253, 1946.

Nonlinear Viscoelasticity

The nonlinear viscoelasticity models in Abaqus are intended for modeling nonlinear rate-dependent behavior of materials subjected to large strains, such as elastomers and polymers.

In this section:

- *Hysteresis in Elastomers*
- *Parallel Rheological Framework*

Hysteresis in Elastomers

Products: Abaqus/Standard Abaqus/CAE

References:

- *Elastic Behavior*
- **HYSTERESIS*
- *Defining hysteretic behavior for an isotropic hyperelastic material model*

Overview

The hysteresis material model:

- provides inelastic response only for shear distortional behavior—the response to volumetric deformations is purely elastic;
- can be used only in conjunction with *Hyperelastic Behavior of Rubberlike Materials* to define the elastic response of the material—the elasticity can be defined either in terms of the instantaneous moduli or the long-term moduli;
- is active during a static analysis (*Static Stress Analysis*), a quasi-static analysis (*Quasi-Static Analysis*), or a transient dynamic analysis using direct integration (*Implicit Dynamic Analysis Using Direct Integration*)—it cannot be used in fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*), fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*), or steady-state transport analysis (*Steady-State Transport Analysis*);
- cannot be used to model temperature-dependent creep material properties—however, the elastic material properties can be temperature dependent; and
- uses unsymmetric matrix storage and solution by default.

Strain-Rate-Dependent Material Behavior for Elastomers

Nonlinear strain-rate dependence of elastomers is modeled by decomposing the mechanical response into that of an equilibrium network (A) corresponding to the state that is approached in long-time stress relaxation tests and that of a time-dependent network (B) that captures the nonlinear rate-dependent deviation from the equilibrium state. The total stress is assumed to be the sum of the stresses in the two networks. The deformation gradient, $\mathbf{F}$, is assumed to act on both networks and is decomposed into elastic and inelastic parts in network B according to the multiplicative decomposition $\mathbf{F} = \mathbf{F}_B^e \cdot \mathbf{F}_B^{cr}$. The nonlinear rate-dependent material model is capable of reproducing the hysteretic behavior of elastomers subjected to repeated cyclic loading. It does not model "Mullins effect"—the initial softening of an elastomer when it is first subjected to a load.

The material model is defined completely by:

- a hyperelastic material model that characterizes the elastic response of the model;
- a stress scaling factor, S, that defines the ratio of the stress carried by network B to the stress carried by network A under instantaneous loading; that is, identical elastic stretching in both networks;
- a positive exponent, m, generally greater than 1, characterizing the effective stress dependence of the effective creep strain rate in network B;
- an exponent, C, restricted to lie in $[-1, 0]$, characterizing the creep strain dependence of the effective creep strain rate in network B;

- a nonnegative constant, A, in the expression for the effective creep strain rate—this constant also maintains dimensional consistency in the equation; and
- a constant, E, in the expression for the effective creep strain rate—this constant regularizes the creep strain rate near the undeformed state.

The effective creep strain rate in network B is given by the expression

$$\dot{\epsilon}_B^{cr} = A\left[\lambda_B^{cr} - 1 + E\right]^C (\sigma_B)^m,$$

where $\dot{\epsilon}_B^{cr}$ is the effective creep strain rate in network B, $\lambda_B^{cr} - 1$ is the nominal creep strain in network B, and σ_B is the effective stress in network B. The chain stretch in network B, λ_B^{cr}, is defined as

$$\lambda_B^{cr} = \sqrt{\frac{1}{3}\mathbf{I} : \mathbf{C}_B^{cr}},$$

where $\mathbf{C}_B^{cr} = \mathbf{F}_B^{crT} \cdot \mathbf{F}_B^{cr}$. The effective stress in network B is defined as $\sigma_B = \sqrt{\frac{3}{2}\mathbf{S}_B : \mathbf{S}_B}$, where $\mathbf{S}_B$ is the deviatoric Cauchy stress tensor.

Defining Strain-Rate-Dependent Material Behavior for Elastomers

The elasticity of the model is defined by a hyperelastic material model. You input the stress scaling factor and the creep parameters for network B directly when you define the hysteresis material model. Typical values of the material parameters for a common elastomer are $S = 1.6$, $A = \frac{5}{(\sqrt{3})^m}(\text{sec})^{-1}(\text{MPa})^{-m}$, $m = 4$, $C = -1.0$, and $E = 0.01$ (Bergstrom and Boyce, 1998; 2001).

Input File Usage: Use both of the following options within the same material data block:

**HYSTERESIS*
**HYPERELASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Hyperelastic**: **Suboptions->Hysteresis**

The input of the parameter E is not supported in Abaqus/CAE.

Elements

The use of the hysteresis material model is restricted to elements that can be used with hyperelastic materials (*Hyperelastic Behavior of Rubberlike Materials*). In addition, this model cannot be used with elements based on the plane stress assumption (shell, membrane, and continuum plane stress elements). Hybrid elements can be used with this model only when the accompanying hyperelasticity definition is completely incompressible. When this model is used with reduced-integration elements, the instantaneous elastic moduli are used to calculate the default hourglass stiffness.

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning if hysteretic behavior is defined:

EE Elastic strain corresponding to the stress state at time t and the instantaneous elastic material properties.

CE Equivalent creep strain defined as the difference between the total strain and the elastic strain.

These strain measures are used to approximate the strain energy, SENER, and the viscous dissipation, CENER. These approximations can lead to underestimation of the strain energy and overestimation of the viscous dissipation since the effects of internal stresses on these energy quantities are neglected. This inaccuracies might be particularly noticeable in the case of nonmonotonic loading.

References

- Bergstrom, J. S., , and M. C. Boyce, "Constitutive Modeling of the Large Strain Time-Dependent Behavior of Elastomers," *Journal of the Mechanics and Physics of Solids*, vol. 46, no. 5, pp. 931–954, May 1998.
- Bergstrom, J. S., , and M. C. Boyce, "Constitutive Modeling of the Time-Dependent and Cyclic Loading of Elastomers and Application to Soft Biological Tissues," *Mechanics of Materials*, vol. 33, no. 9, pp. 523–530, 2001.

Parallel Rheological Framework

Products: Abaqus/Standard Abaqus/Explicit

References:

- *About the Material Library*
- *Combining Material Behaviors*
- *Elastic Behavior*
- *UCREEPNETWORK*
- *UTRSNETWORK*
- *VUCREEPNETWORK*
- *VUTRSNETWORK*
- **HYPERELASTIC*
- **MULLINS EFFECT*
- **NETWORK STIFFNESS RATIO*
- **PLASTIC*
- **TRS*
- **VISCOELASTIC*
- **VISCOUS SOFTENING*

Overview

The parallel rheological framework:

- consists of multiple viscoelastic networks and, optionally, an elastic-plastic network in parallel;
- uses a hyperelastic material model to specify the elastic response;
- can be combined with Mullins effect;
- bases the elastic-plastic response on multiplicative split of the deformation gradient and the theory of incompressible isotropic hardening plasticity;
- can include nonlinear kinematic hardening with multiple backstresses in the elastic-plastic response in Abaqus/Standard; and
- uses multiplicative split of the deformation gradient and a flow rule derived from a creep potential to specify the viscous behavior.

Material Behavior

The parallel rheological framework allows definition of a nonlinear viscoelastic-elastoplastic model consisting of multiple networks connected in parallel, as shown in *Figure 1*.

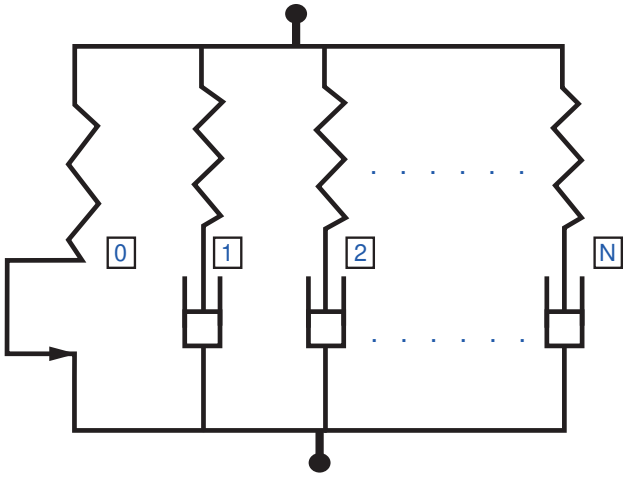


Figure 1: Nonlinear viscoelastic-elastoplastic model with multiple parallel networks.

The number of viscoelastic networks, *N*, can be arbitrary; however, at most one equilibrium network (network **0** in *Figure 1*) is allowed in the model. The equilibrium network response might be purely elastic or elastic-plastic. In addition, it might include Mullins effect to predict material softening. The definition of the equilibrium network is optional. If it is not defined, the stress in the material relaxes completely over time.

The model can be used to predict complex behavior of materials subjected to finite strains, which cannot be modeled accurately using other models available in Abaqus. An example of such complex behavior is depicted in *Figure 2*, which shows normalized stress relaxation curves for three different strain levels. This behavior can be modeled accurately using the nonlinear viscoelastic model depicted in *Figure 3*, which can be defined within the framework; but it cannot be captured with the linear viscoelastic model (see *Time Domain Viscoelasticity*). In the latter case, the three curves would coincide.

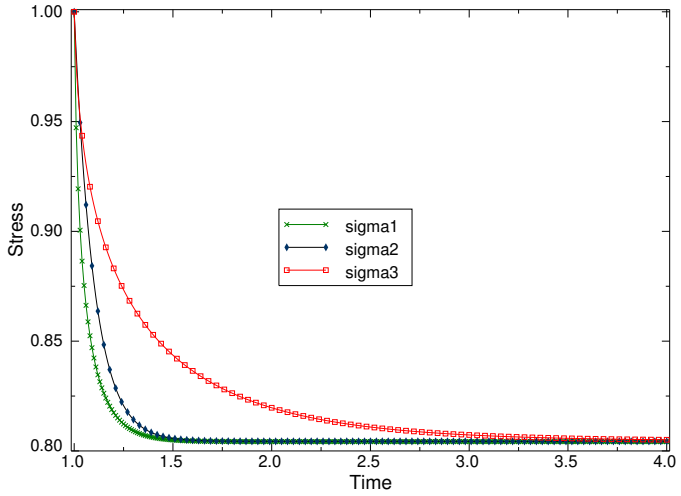


Figure 2: Normalized stress relaxation curves for three different strain levels.

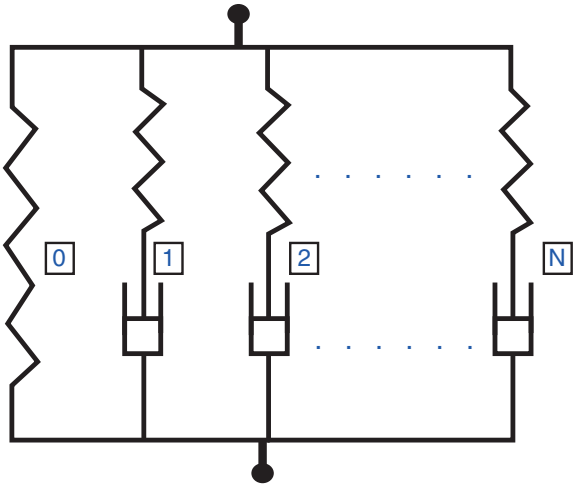


Figure 3: Nonlinear viscoelastic model with multiple parallel networks.

Elastic Behavior

The elastic part of the response for all the networks is specified using the hyperelastic material model. Any of the hyperelastic models available in Abaqus can be used (see *Hyperelastic Behavior of Rubberlike Materials*). The same hyperelastic material definition is used for all the networks, scaled by a stiffness ratio specific to each network. Consequently, only one hyperelastic material definition is required by the model along with the stiffness ratio for each network. The elastic response can be specified by defining either the instantaneous response or the long-term response.

Equilibrium Network Behavior

In addition to the elastic response described above, the response of the equilibrium network can include plasticity and Mullins effect to predict material softening. If the plastic response is defined using isotropic hardening, the response in the equilibrium network is equivalent to that of the permanent set model available in Abaqus (see *Permanent Set in Rubberlike Materials* for a detailed description of the model). In Abaqus/Standard the nonlinear kinematic hardening model with multiple backstresses can be specified in addition to isotropic plastic hardening. The nonlinear kinematic hardening model is a generalization of the model used for metal plasticity. See *Models for Metals Subjected to Cyclic Loading* for a detailed description of the model, with the difference that the Cauchy stress is replaced with the Kirchhoff stress in the current formulation.

Viscous Behavior

Viscous behavior must be defined for each viscoelastic network. It is modeled by assuming the multiplicative split of the deformation gradient and the existence of the creep potential, G^{cr}, from which the flow rule is derived. In the multiplicative split, the deformation gradient is expressed as

$$\mathbf{F} = \mathbf{F}^{e} \cdot \mathbf{F}^{cr},$$

where $\mathbf{F}^{e}$ is the elastic part of the deformation gradient (representing the hyperelastic behavior) and $\mathbf{F}^{cr}$ is the creep part of the deformation gradient (representing the stress-free intermediate configuration). The creep potential is assumed to have the general form

$$G^{cr} = G^{cr}(\boldsymbol{\sigma}),$$

where $\boldsymbol{\sigma}$ is the Cauchy stress. If the potential is specified, the flow rule can be obtained from

$$\mathbf{D}^{cr} = \dot{\lambda}\frac{\partial G^{cr}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}},$$

where $\mathbf{D}^{cr}$ is the symmetric part of the velocity gradient, $\mathbf{L}^{cr}$, expressed in the current configuration and $\dot{\lambda}$ is the proportionality factor. In this model, the creep potential is given by

$$G^{cr} = \bar{q},$$

and the proportionality factor is taken as $\dot{\lambda} = \dot{\bar{\varepsilon}}^{cr}$, where $\bar{q}$ is the equivalent deviatoric Cauchy stress and $\dot{\bar{\varepsilon}}^{cr}$ is the equivalent creep strain rate. In this case, the flow rule has the form

$$\mathbf{D}^{cr} = \frac{3}{2\bar{q}}\dot{\bar{\varepsilon}}^{cr}\overline{\boldsymbol{\sigma}},$$

or, equivalently

$$\mathbf{D}^{cr} = \frac{3}{2\tilde{q}}\dot{\bar{\varepsilon}}^{cr}\overline{\boldsymbol{\tau}},$$

where $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ is the Kirchhoff stress, J is the determinant of $\mathbf{F}$, $\overline{\boldsymbol{\sigma}}$ is the deviatoric Cauchy stress, $\overline{\boldsymbol{\tau}}$ is the deviatoric Kirchhoff stress, and $\tilde{q} = J\bar{q}$. To complete the derivation, the evolution law for $\dot{\bar{\varepsilon}}^{cr}$ must be provided. In this model $\dot{\bar{\varepsilon}}^{cr}$ can be defined by the power law model, the strain hardening model, the hyperbolic-sine law model, the Bergstrom-Boyce model, the Darveaux model, the modified Darveaux model, or a user-defined creep model.

Power Law Model

The power law model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0\left(\left(\frac{\tilde{q}}{q_0 + a\langle p\rangle}\right)^n[(m+1)\,\bar{\varepsilon}^{cr}]^m\right)^{\frac{1}{m+1}},$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the equivalent creep strain rate,

$\bar{\varepsilon}^{cr}$

is the equivalent creep strain,

$\tilde{q}$

is the equivalent deviatoric Kirchhoff stress,

p

is the Kirchhoff pressure, and

q_0, m, n, a, and $\dot{\varepsilon}_0$

are material parameters. It is recommended that you use the power law model rather than the strain hardening model.

Strain Hardening Model

The strain hardening model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = (A\tilde{q}^n[(m+1)\,\bar{\varepsilon}^{cr}]^m)^{\frac{1}{m+1}},$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the equivalent creep strain rate,

$\bar{\varepsilon}^{cr}$

is the equivalent creep strain,

$\tilde{q}$

is the equivalent deviatoric Kirchhoff stress, and

A, m, and n

are material parameters. It is recommended that you use the power law model rather than the strain hardening model. The strain hardening model is a special case of the power law model obtained by setting $\dot{\varepsilon}_0 = 1$, $a = 0$, and $q_0 = A^{-1/n}$.

Hyperbolic-Sine Law Model

The hyperbolic-sine law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A(\sinh B\tilde{q})^n,$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$

are defined above, and

A, B, and n

are material parameters.

Bergstrom-Boyce Model

Abaqus provides two forms to define the Bergstrom-Boyce creep model. The recommended form of the Bergstrom-Boyce model is defined as

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0(\lambda^{cr} - 1 + E)^C \left(\frac{\tilde{q}}{q_0}\right)^m,$$

where

$$\lambda^{cr} = \sqrt{\tfrac{1}{3}\mathbf{I} : \mathbf{C}^{cr}},$$

and

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$

are defined above, and

q_0, m, C, E, and $\dot{\varepsilon}_0$

are material parameters.

The original Bergstrom-Boyce model has the form

$$\dot{\tilde{\varepsilon}}^{cr} = A(\lambda^{cr} - 1 + E)^C(\tilde{q})^m,$$

where

$\dot{\tilde{\varepsilon}}^{cr}$, λ^{cr}, and $\tilde{q}$

are defined above, and

A*, *m*, *C*, and *E

are material parameters.

The recommended form is equivalent to the original form of the Bergstrom-Boyce model. The primary difference between the two formulations is that the recommended form is written in such a way that parameter values do not cause numerical difficulties, which can happen when the original model is calibrated for strain rate applications. In addition, the units of all parameters in the recommended form are physical, which makes unit conversion easier. When the value of the parameter A is very small ($A \leq 10^{-27}$), the recommended form is obtained by setting $q_0 = \left(\frac{\dot{\varepsilon}_0}{A}\right)^{\frac{1}{m}}$ and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

The response of the network defined by the Bergstrom-Boyce model is very similar to the response of the time-dependent network in the hysteresis model (see *Hysteresis in Elastomers*). However, there are also important differences between the models. In the Bergstrom-Boyce model the equivalent Kirchhoff stress is used instead of the equivalent Cauchy stress, which is used in the hysteresis model. (The two stress measures become equivalent for the case of incompressible materials.) In addition, the material parameters, *A*, in the hysteresis model and the original form of the Bergstrom-Boyce model differ by a factor of $3/2$. The parameter in the hysteresis model must be multiplied by $3/2$ to make the parameters equivalent.

Darveaux Model

The original Darveaux model involves both primary and secondary (steady state) creep. It is intended for applications where loading is applied at the beginning of the analysis, while the modified Darveaux model is recommended for arbitrary loading histories. The secondary creep in the Darveaux model is defined by a hyperbolic-sine law:

$$\dot{\tilde{\varepsilon}}_s^{cr} = C_{ss}[\sinh(\alpha\tilde{q})]^n.$$

The secondary law is modified to account for the primary creep effects through

$$\dot{\tilde{\varepsilon}}^{cr} = \dot{\tilde{\varepsilon}}_s^{cr}\left[1 + \epsilon_T B \exp\left(-B\dot{\tilde{\varepsilon}}_s^{cr} t\right)\right],$$

where

$\dot{\tilde{\varepsilon}}^{cr}$ and $\tilde{q}$

are defined above,

$\dot{\tilde{\varepsilon}}_s^{cr}$

is the steady-state equivalent creep strain rate,

C_{ss}

is the steady-state creep prefactor,

α
: is the steady-state power law breakdown, and

n, ϵ_T, and B
: are other material parameters.

Modified Darveaux Model

The modified Darveaux model involves both primary and secondary (steady state) creep, similar to the original Darveaux model. However, unlike the original Darveaux model, which uses total time to describe primary creep effects, the modified Darveaux model uses the accumulated creep strain. 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 Darveaux models provide the same response under constant load and temperature applied at the beginning of the analysis (time zero). However, for arbitrary loading histories, you should use the modified Darveaux model.

The secondary creep is defined by a hyperbolic-sine law:

$$\dot{\bar{\varepsilon}}_s^{cr} = C_{ss}[\sinh(\alpha\tilde{q})]^n.$$

The secondary law is modified to account for the primary creep effects through

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\bar{\varepsilon}}_s^{cr}\left[1 + \epsilon_T B \exp(-B\bar{\varepsilon}_s^{cr})\right],$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$
: are defined above,

$\bar{\varepsilon}_s^{cr}$
: is the accumulated steady-state equivalent creep strain,

C_{ss}, α, n, ϵ_T, and B
: are defined in the Darveaux model.

Anand Model

The Anand model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A\left[\sinh\left(\xi\frac{\tilde{q}}{s}\right)\right]^{\frac{1}{m}},$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$
: are defined above, and

A, m, and ξ
: are material parameters.

The evolution equation for the deformation resistance, s (initially $s = s_0$), is

$$\dot{s} = h_0 \left| 1 - \frac{s}{s^*} \right|^a sign\left(1 - \frac{s}{s^*}\right) \dot{\bar{\varepsilon}}^{cr},$$

with

$$s^* = \hat{s} \, \sinh\left(\xi \frac{\bar{q}}{s}\right)^{\frac{n}{m}},$$

where

$$h_0 = A_0 + A_1\left(\theta - \theta^Z\right) + A_2\left(\theta - \theta^Z\right)^2 + A_3 \dot{\bar{\varepsilon}}^{cr} + A_4\left(\dot{\bar{\varepsilon}}^{cr}\right)^2,$$

and where

θ

is the temperature,

θ^Z

is the user-defined value of absolute zero on the temperature scale used, and

a, n, $\hat{s}$, A_0, A_1, A_2, A_3, and A_4

are material parameters.

In addition, the initial deformation resistance is a function of temperature of the form

$$s_0 = S_1 + S_2\left(\theta - \theta^Z\right) + S_3\left(\theta - \theta^Z\right)^2,$$

where S_1, S_2, and S_3 are material parameters.

User-Defined Model

A user-defined creep model is available of the following general form:

$$\dot{\bar{\varepsilon}}^{cr} = g^{cr}\left(\bar{\varepsilon}^{cr}, I_1^{cr}, \bar{I}_1, \bar{I}_2, J, p, \tilde{q}, t, \theta, FV\right),$$

where

$$I_1^{cr} = \mathbf{I} : \mathbf{C}^{cr},$$

and

$\bar{I}_1$

is the first invariant of $\overline{\mathbf{B}}$,

$\bar{I}_2$

is the second invariant of $\overline{\mathbf{B}}$,

$\dot{\bar{\varepsilon}}^{cr}$, $\bar{\varepsilon}^{cr}$, J, p, and $\tilde{q}$

are defined above,

t

is the time,

θ

is the temperature, and

FV

are field variables.

The tensor $\overline{\mathbf{B}}$ is defined in *UCREEPNETWORK* or in *VUCREEPNETWORK*.

Viscous Softening

Viscous softening can be included for each viscoelastic network. For a given network, viscous softening is modeled by a scaling factor, η^{cr}, multiplying the deviatoric stress of the network. This scaling factor is a function of the equivalent creep strain in the network. Abaqus supports the exponential form and the tabular form of the function.

Exponential Form

The exponential form is available as

$$\eta^{cr}\left(\bar{\varepsilon}^{cr}\right) = \eta_f^{cr} + \left(1 - \eta_f^{cr}\right) \exp\left(-\frac{\bar{\varepsilon}^{cr}}{\bar{\varepsilon}_0^{cr}}\right),$$

where

$\bar{\varepsilon}^{cr}$

is as defined above ($0 \le \eta_f^{cr} \le 1$ and $\bar{\varepsilon}_0^{cr}$ are material parameters),

$0 \le \eta_f^{cr} \le 1$

is the maximum level of softening, and

$\bar{\varepsilon}_0^{cr}$

is the characteristic equivalent creep strain of the viscous softening.

Tabular Form

You can specify the viscous softening in tabular form. In this case, you provide the scaling factor as a function of the equivalent creep strain. The scaling factor must satisfy $0 < \eta^{cr} \le 1$.

Thermal Expansion

Only isotropic thermal expansion is permitted with nonlinear viscoelastic materials (see *Thermal Expansion*).

Defining Viscoelastic Response

The nonlinear viscoelastic response is defined by specifying the identifier, stiffness ratio, and creep law for each viscoelastic network. Viscous softening and thermorheologically simple temperature effects can be optionally specified for each viscoelastic network.

Specifying Network Identifier

Each viscoelastic network in the material model must be assigned a unique network identifier or network id. The network identifiers must be consecutive integers starting with 1. The order in which they are specified is not important.

Input File Usage: Use the following option to specify the network identifier:

**VISCOELASTIC*, NONLINEAR, NETWORKID=*networkId*

Defining the Stiffness Ratio

The contribution of each network to the overall response of the material is determined by the value of the stiffness ratio, r, which is used to scale the elastic response of the network material. The sum of the stiffness ratios of the viscoelastic networks must be smaller than or equal to 1. If the sum of the ratios is equal to 1, the purely elastic equilibrium network is not created. If the sum of the ratios is smaller than 1, the equilibrium network is created with a stiffness ratio, r_0, equal to

$$r_0 = 1 - \sum_{k=1}^{N} r_k,$$

where N denotes the number of viscoelastic networks and r_k is the stiffness ratio of network k. You can specify the stiffness ratio to remain constant during the analysis or to vary as a function of temperature and predefined field variables.

Defining a Constant Stiffness Ratio

You can specify that the stiffness ratio remains constant during the analysis:

Input File Usage: **VISCOELASTIC*, NONLINEAR, SRATIO=*ratio*

Defining a Temperature- and Field-Variable Dependent Stiffness Ratio

Alternatively, you can define the stiffness ratio as a function of temperature and predefined field variables.

Input File Usage: **NETWORK STIFFNESS RATIO*, N=*nNetworks*, DEPENDENCIES=*n*

Specifying the Creep Law

The definition of creep behavior is completed by specifying the creep law.

Power Law Creep Model

The power law model is defined by specifying five material parameters: q_0, n, m, a, and $\dot{\varepsilon}_0$. The parameter $\dot{\varepsilon}_0$ must be positive. It is introduced for dimensional consistency, and its default value is 1.0. For physically reasonable

behavior q_0 and n must be positive, a must be nonnegative (the default is 0.0), and $-1 < m \leq 0$. It is recommended that you use the power law model rather than the strain hardening model.

Input File Usage: **VISCOELASTIC*, NONLINEAR, LAW=POWER LAW

Strain Hardening Creep Model

The strain hardening law is defined by specifying three material parameters: A, n, and m. For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$. It is recommended that you use the power law model rather than the strain hardening model.

Input File Usage: **VISCOELASTIC*, NONLINEAR, LAW=STRAIN

Hyperbolic Sine Creep Model

The hyperbolic sine creep law is specified by providing three nonnegative parameters: A, B, and n.

Input File Usage: **VISCOELASTIC*, NONLINEAR, LAW=HYPERB

Bergstrom-Boyce Creep Model

The recommended form of the Bergstrom-Boyce creep law is specified by providing five parameters: q_0, m, C, E, and $\dot{\varepsilon}_0$. The parameters $\dot{\varepsilon}_0$ and E must be nonnegative, the parameters q_0 and m must be positive, and the parameter C must lie in $[-1, 0]$.

The original form of the Bergstrom-Boyce creep law is specified by providing four parameters: A, m, C, and E. The parameters A and E must be nonnegative, the parameter m must be positive, and the parameter C must lie in $[-1, 0]$.

Input File Usage: Use the following option to define the recommended form of the Bergstrom-Boyce creep law:

**VISCOELASTIC*, NONLINEAR, LAW=BB POWER LAW

Use the following option to define the original form of the Bergstrom-Boyce creep law:

**VISCOELASTIC*, NONLINEAR, LAW=BERGSTROM-BOYCE

Darveaux Creep Model

The Darveaux creep law is specified by providing five parameters: C_{ss}, α, n, ϵ_T, and B.

Input File Usage: Use the following option to define the Darveaux creep law.

**VISCOELASTIC*, NONLINEAR, LAW=DARVEAUX

Modified Darveaux Creep Model

The modified Darveaux creep law is specified by providing five parameters: C_{ss}, α, n, ϵ_T, and B.

Input File Usage: Use the following option to define the Darveaux creep law.

**VISCOELASTIC*, NONLINEAR, LAW=MODIFIED DARVEAUX

Anand Creep Model

The Anand creep law is specified by providing the following parameters: S_1, A, ξ, m, A_0, $\hat{s}$, n, a, S_2, S_3, A_1, A_2, A_3, and A_4.

Input File Usage: Use the following option to define the Anand creep law:

**VISCOELASTIC*, NONLINEAR, LAW=ANAND

User-Defined Creep Model

An alternative method for defining the creep law involves using user subroutine *UCREEPNETWORK* in Abaqus/Standard or *VUCREEPNETWORK* in Abaqus/Explicit. Optionally, you can specify the number of property values required as data in the user subroutine.

Input File Usage: **VISCOELASTIC*, NONLINEAR, LAW=USER, PROPERTIES=*n*

Numerical Difficulties

Depending on the choice of units, the value of A in the creep models might be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations; therefore, a different system of units should be used to avoid such difficulties in the calculation of creep strain increments. In such cases, it is recommended that you use the creep models that do not have the limitation. You can use the power law model rather than the strain hardening model and the recommended form of the Bergstrom-Boyce model rather than the original form.

Specifying Viscous Softening

You can include viscous softening for each viscoelastic network. To complete the definition of viscous softening, you must specify the form of the softening.

Exponential Form

The exponential viscous softening is defined by specifying two parameters: η_f^{cr} and $\bar{\varepsilon}_0^{cr}$.

Input File Usage: Use the following option to define exponential viscous softening:

**VISCOUS SOFTENING*, TYPE=EXPONENTIAL

Tabular Form

The tabular viscous softening is specified by providing the scaling factor as a function of the equivalent creep strain.

Input File Usage: Use the following option to define tabular viscous softening:

**VISCOUS SOFTENING*, TYPE=TABULAR

Thermorheologically Simple Temperature Effects

Thermorheologically simple temperature effects can be included for each viscoelastic network. In this case the creep law is modified and takes the following form:

$$\frac{d\bar{\varepsilon}^{cr}}{d\tau} = \frac{1}{a_T(\theta)} g^{cr}\left(\bar{\varepsilon}^{cr}, I_1^{cr}, \bar{I}_1, \bar{I}_2, J, p, \tilde{q}, \tau\right),$$

where τ and $a_T(\theta)$ denote the reduced time and the shift function, respectively. The reduced time is related to the actual time through the integral differential equation

$$\tau = \int_0^t \frac{dt'}{a_T(\theta)}, \qquad \frac{d\tau}{dt} = \frac{1}{a_T(\theta)}.$$

Abaqus supports the following forms of the shift function: the Williams-Landel-Ferry (WLF) form, the Arrhenius form, and the tabular form (see *Thermorheologically Simple Temperature Effects*). In addition, user-defined forms can be specified.

User-Defined Form

An alternative method for specifying the shift function involves using user subroutine *UTRSNETWORK* in Abaqus/Standard and user subroutine *VUTRSNETWORK* in Abaqus/Explicit. Optionally, you can specify the number of property values required as data in the user subroutine.

Input File Usage: **TRS*, DEFINITION=USER, PROPERTIES=*n*

Material Response in Different Analysis Steps

In Abaqus/Standard the material is active during all stress/displacement procedure types. However, the creep effects are taken into account only in quasi-static (*Quasi-Static Analysis*), coupled temperature-displacement (*Fully Coupled Thermal-Stress Analysis*), direct-integration implicit dynamic (*Implicit Dynamic Analysis Using Direct Integration*), and steady-state transport (*Steady-State Transport Analysis*) analyses. If the material is used in a steady-state transport analysis, it cannot include plasticity. In other stress/displacement procedures, the evolution of the state variables is suppressed and the creep strain remains unchanged.

You can use the fully relaxed long-term elastic solution directly in the following analysis types:

- a static or steady-state transport analysis in Abaqus/Standard (see *Rate-Dependent Yield and Friction*), or
- in or in an explicit dynamic or in an explicit dynamic coupled thermal stress analysis in Abaqus/Explicit (see *Controlling Material Rate Dependence* and *Controlling Material and Friction Rate Dependence*).

When you select this option, Abaqus updates the material state and outputs the creep strain consistent with the long-term response.

Elements

The parallel rheological framework is available with continuum elements that include mechanical behavior (elements that have displacement degrees of freedom), except for one-dimensional elements. The parallel rheological framework is also supported with shells and membrane elements. Abaqus/Explicit also supports the parallel rheological framework with beams in space.

When the parallel rheological framework is used with structural elements (shells, membranes, and beams in space), Abaqus ignores the compressibility of the material.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the nonlinear viscoelastic material model:

CEEQ
The overall equivalent creep strain, defined as $\bar{\varepsilon}^{cr} = \sum_{k=1}^{N} r_k \bar{\varepsilon}_k^{cr}$.

CEEQNET
The equivalent creep strain in all the networks, $\bar{\varepsilon}_k^{cr}$, ($1 \le k \le N$).

CEEQNET*k*
The equivalent creep strain $\bar{\varepsilon}_k^{cr}$ in the k^{th} network ($1 \le k \le N$).

CEEQSSNET
The steady-state equivalent creep strain in all the networks that use the modified Darveaux creep law, $\bar{\varepsilon}_{s,k}^{cr}$, ($1 \le k \le N$).

CEEQSSNET*k*
The steady-state equivalent creep strain $\bar{\varepsilon}_{s,k}^{cr}$ in the k^{th} network ($1 \le k \le N$) if it uses the modified Darveaux creep law.

CE
The overall creep strain, defined as $\boldsymbol{\varepsilon}^{cr} = \sum_{k=1}^{N} r_k \boldsymbol{\varepsilon}_k^{cr}$.

CENER
The overall viscous dissipated energy per unit volume, defined as $E_c = \sum_{k=1}^{N} E_c^k$.

EE
The overall elastic strain, defined as $\boldsymbol{\varepsilon}^{el} = \sum_{k=1}^{N} r_k \boldsymbol{\varepsilon}_k^{el}$.

SDEFRESNET
The deformation resistance in all the networks that use the Anand creep law, s_k, ($1 \le k \le N$).

SDEFRESNET*k*
The deformation resistance s_k in the k^{th} network ($1 \le k \le N$) if it uses the Anand creep law.

SENER
The overall elastic strain energy density per unit volume, defined as $E_s = \sum_{k=0}^{N} E_s^k$.

SNET*k*
All stress components in the k^{th} network ($0 \le k \le 10$).

In the above definitions r_k denotes the stiffness ratio for network k, N denotes the number of viscoelastic networks, the subscript or superscript k is used to denote network quantities, and the network 0 is assumed to be the purely elastic network.

If viscous softening is specified, the following variables have special meaning:

DMENER
Energy dissipated per unit volume by viscous softening.

ELDMD	Total energy dissipated in element by viscous softening.
ALLDMD	Energy dissipated in whole (or partial) model by viscous softening. The contribution from ALLDMD is included in the total strain energy ALLIE.
EDMDDEN	Energy dissipated per unit volume in the element by viscous softening.

If plasticity is specified in the equilibrium network, the standard output identifiers available in Abaqus corresponding to other isotropic and kinematic hardening plasticity models can be obtained for this model as well. In addition, if the Mullins effect is used in the model, the output variables available for the Mullins effect model (see *Stress Softening in Rubberlike Materials*) can be requested. If Mullins effect and viscous softening are both specified in the model, the above damage energy quantities contain contributions from both.

References

- Bergstrom, J. S., and M. C. Boyce, "Constitutive Modeling of the Large Strain Time-Dependent Behavior of Elastomers," *Journal of the Mechanics and Physics of Solids*, vol. 46, pp. 931–954, 1998.
- Bergstrom, J. S., and M. C. Boyce, "Large Strain Time-Dependent Behavior of Filled Elastomers," *Mechanics of Materials*, vol. 32, pp. 627–644, 2000.
- Bergstrom, J. S., and J. E. Bischoff, "An Advanced Thermomechanical Constitutive Model for *UHMWPE*," *International Journal of Structural Changes in Solids*, vol. 2, pp. 31–39, 2010.
- Hurtado, J. A., I. Lapczyk, and S. M. Govindarajan, "Parallel Rheological Framework to Model Non-Linear Viscoelasticity, Permanent Set, and Mullins Effect in Elastomers," *Constitutive Models for Rubber VIII* 95, 2013.
- Lapczyk, I., J. A. Hurtado, and S. M. Govindarajan, "A Parallel Rheological Framework for Modeling Elastomers and Polymers," *182nd Technical Meeting of the Rubber Division of the American Chemical Society*, pp. 1840–1859, October 2012, Cincinnati, Ohio.

Low-Density Foams

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Elastic Behavior*
- **LOW DENSITY FOAM*
- **UNIAXIAL TEST DATA*
- *Creating a low-density foam material model*

Overview

The low-density foam material model:

- is primarily intended for low-density, highly compressible foams with significant rate sensitive behavior (such as polyurethane foam) but can also be used to model nearly incompressible rubberlike materials;
- ignores strain-rate dependence, tension cutoff, and failure in Abaqus/Standard;
- requires the direct specification of uniaxial stress-strain curves at different strain rates for both tension and compression;
- optionally allows the specification of lateral strain data to include Poisson effects;
- allows for the specification of optional unloading stress-strain curves for better representation of the hysteretic behavior and energy absorption during cyclic loading; and
- requires that geometric nonlinearity be accounted for during the analysis step (see *Defining an Analysis* and *General and Perturbation Procedures*), since it is intended for finite-strain applications.

Mechanical Response

Low-density, highly compressible elastomeric foams are widely used in the automotive industry as energy absorbing materials. Foam padding is used in many passive safety systems, such as behind headliners for head impact protection, in door trims for pelvis and thorax protection, etc. Energy absorbing foams are also commonly used in packaging of hand-held and other electronic devices.

The low-density foam material model in Abaqus/Explicit is intended to capture the highly strain-rate sensitive behavior of these materials. The model in Abaqus/Standard is intended to be used in quasi-static analyses where strain-rate sensitivity can be ignored. The model uses a pseudo visco-hyperelastic formulation where the strain energy potential is constructed numerically as a function of principal stretches and a set of internal variables associated with strain rate. By default the Poisson's ratio of the material is assumed to be zero. With this assumption, the evaluation of the stress-strain response becomes uncoupled along the principal deformation directions. Optionally, nonzero Poisson effects can be specified to include coupling along the principal directions.

The model requires as input the stress-strain response of the material for both uniaxial tension and uniaxial compression tests. Poisson effects can be included by also specifying lateral strain data for each of these tests. The tests can be performed at different strain rates. For each test the strain data should be given in nominal strain values (change in length per unit of original length), and the stress data should be given in nominal stress values (force per unit of original cross-sectional area). Uniaxial tension and compression curves are specified separately. The uniaxial stress and strain data are given in absolute values (positive in both tension and compression). On

the other hand, when specified, the lateral strain data must be negative in tension and positive in compression, corresponding to a positive Poisson's effect. The model does not support negative Poisson's effect. The rate-dependent behavior can be specified by providing the uniaxial stress-strain curves for different values of nominal strain rates. Only the response of the material in Abaqus/Explicit is affected. Rates effects are ignored in Abaqus/Standard, and the material response is based on the quasi-static behavior.

Both loading and unloading rate-dependent curves can be specified to better characterize the hysteretic behavior and energy absorption properties of the material during cyclic loading. Use positive values of nominal strain rates for loading curves and negative values for the unloading curves. Currently this option is available only with linear strain rate interpolation (see *Interpolation of Strain Rate–Dependent Data in Abaqus/Explicit*). When the unloading behavior is not specified directly, the model assumes that unloading occurs along the loading curve associated with the smallest deformation rate. A representative schematic of typical rate-dependent uniaxial compression data is shown in *Figure 1* with both loading and unloading curves. It is important that the specified rate-dependent stress-strain curves do not intersect. Otherwise, the material is unstable, and Abaqus issues an error message if an intersection between curves is found.

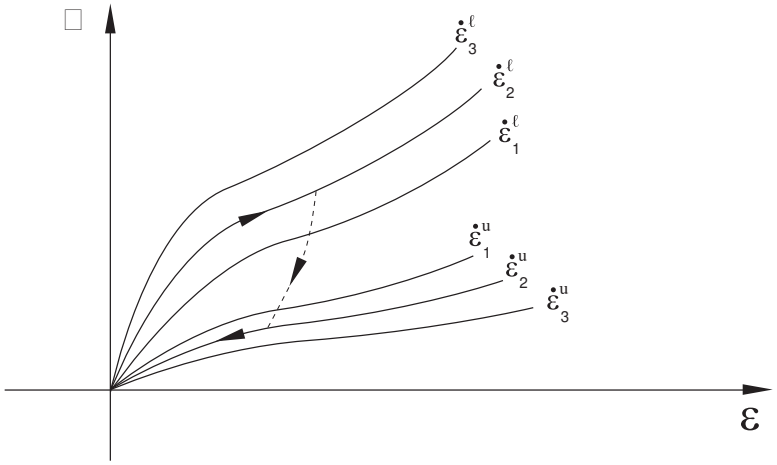


Figure 1: Rate-dependent loading/unloading stress-strain curves.

During the analysis, the stress along each principal deformation direction is evaluated by interpolating the specified loading/unloading stress-strain curves using the corresponding values of principal nominal strain and strain rate. The stress is then corrected by a coupling term if nonzero Poisson effects are included. The representative response of the model for a uniaxial compression cycle is shown in *Figure 1*.

Input File Usage: Use the following options to specify a low-density foam material:

**LOW DENSITY FOAM*
**UNIAXIAL TEST DATA*, DIRECTION=TENSION
**UNIAXIAL TEST DATA*, DIRECTION=COMPRESSION

Use the first option to specify a low-density foam material with zero Poisson's ratio (default), or use the second option to include Poisson effects by defining lateral strains as part of the test data input:

**LOW DENSITY FOAM*,LATERAL STRAIN DATA=NO (default)
**LOW DENSITY FOAM*, LATERAL STRAIN DATA=YES

In addition, use these two options to give the experimental stress-strain data:

**UNIAXIAL TEST DATA*, DIRECTION=TENSION
**UNIAXIAL TEST DATA*, DIRECTION=COMPRESSION

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: **Uniaxial Test Data->Uniaxial Tension Test Data**, **Uniaxial Test Data->Uniaxial Compression Test Data**

Relaxation Coefficients in Abaqus/Explicit

Unphysical jumps in stress due to sudden changes in the deformation rate are prevented using a technique based on viscous regularization. This technique also models stress relaxation effects in a very simplistic manner. In the case of a uniaxial test, for example, the relaxation time is given as $\tau = \mu_0 + \mu_1|\lambda - 1|^{\alpha}$ where μ_0, μ_1, and α are material parameters and λ is the stretch. μ_0 is a linear viscosity parameter that controls the relaxation time when $\lambda \approx 1$, and typically small values of this parameter should be used. μ_1 is a nonlinear viscosity parameter that controls the relaxation time at higher values of deformation. The smaller this value, the shorter the relaxation time. α controls the sensitivity of the relaxation speed to the stretch. The default values of these parameters are $\mu_0 = 0.0001$ (time units), $\mu_1 = 0.005$ (time units), and $\alpha = 2$.

Input File Usage: Use the following option to specify relaxation coefficients:

**LOW DENSITY FOAM*
μ_0, μ_1, α

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: **Relaxation coefficients:** *mu0*, *mu1*, *alpha*

Strain Rate in Abaqus/Explicit

When Poisson's ratio is zero, three different strain rate measures can be used for the evaluation of the stress-strain response along each principal deformation direction for general three-dimensional deformation states: the nominal volumetric strain rate, the nominal strain rate along each principal deformation direction, or the maximum of the nominal strain rates along the principal deformation directions. By default, the nominal volumetric strain rate is used; this approach does not produce rate-sensitive behavior under volume-preserving deformation modes (for example, simple shear). Alternatively, each principal stress can be evaluated based either on the nominal strain rate along the corresponding principal direction or the maximum of the nominal strain rates; both these approaches can provide rate-sensitive behavior for volume-preserving deformation modes. All three strain rate measures produce identical rate-dependent behavior for uniaxial loading conditions when the Poisson's ratio is zero.

When nonzero Poisson effects are defined, the model uses the maximum nominal strain rate along the principal deformation directions for the evaluation of the stress-strain response. This is the default and only strain rate measure available for this case.

Input File Usage: Use the following option to use the volumetric strain rate (default when Poisson's ratio is zero):

**LOW DENSITY FOAM*, STRAIN RATE=VOLUMETRIC

Use the following option to use the nominal strain rate evaluated along each principal direction:

**LOW DENSITY FOAM*, STRAIN RATE=PRINCIPAL

Use the following option to use the maximum of the nominal strain rates along the principal directions (default and only option available when Poisson's ratio is not zero):

**LOW DENSITY FOAM*, STRAIN RATE=MAX PRINCIPAL

Abaqus/CAE Usage: Use the following option to use the volumetric strain rate (default):

Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: **Strain rate measure: Volumetric**

Use the following option to use the strain rate evaluated along each principal direction:

Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: **Strain rate measure: Principal**

Extrapolation of Stress-Strain Curves

By default, for this material model and for strain values beyond the range of specified strains, Abaqus extrapolates the stress-strain curves using the slope at the last data point.

When the strain rate value exceeds the maximum specified strain rate, Abaqus/Explicit uses the stress-strain curve corresponding to the maximum specified strain rate by default. You can override this default and activate strain rate extrapolation based on the slope (with respect to strain rate).

Input File Usage: Use the following option to activate strain rate extrapolation of loading curves in Abaqus/Explicit:

**LOW DENSITY FOAM*, RATE EXTRAPOLATION=YES

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: toggle on **Extrapolate stress-strain curve beyond maximum strain rate**

Compressibility

The low-density foam model is primarily intended for highly compressible material behavior. In this case the model is formulated in terms of total principal stretches such that the deviatoric (distortional) and volumetric response are fully coupled. Optionally, you can use the model with nearly incompressible rubberlike materials. For such materials Abaqus switches to a deviatoric formulation that decouples distortional and volumetric deformation, which improves the robustness of the solution. Abaqus uses this deviatoric formulation automatically if the Poisson's ratio of the material specified is greater than 0.475. In Abaqus/Standard it is recommended that you use solid continuum hybrid elements in this case to avoid potential convergence problems.

Tension Cutoff and Failure in Abaqus/Explicit

Low-density foams have limited strength in tension and can easily rupture under excessive tensile loading. The model in Abaqus/Explicit provides the option to specify a cutoff value for the maximum principal tensile stress that the material can sustain. The maximum principal stresses computed by the program will stay at or below this value. You can also activate deletion (removal) of the element from the simulation when the tension cutoff value is reached, which provides a simple method for modeling rupture.

When specified, the tension cutoff and failure options are ignored in Abaqus/Standard.

Input File Usage: Use the following option to define a tension cutoff value without element deletion:

**LOW DENSITY FOAM*, TENSION CUTOFF=*value*

Use the following option to allow element deletion when the tension cutoff value is met:

**LOW DENSITY FOAM*, TENSION CUTOFF=*value*, FAIL=YES

Abaqus/CAE Usage: Use the following option to define a tension cutoff value:

Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: toggle on **Maximum allowable principal tensile stress:** *value*

Use the following option to allow element deletion when the tension cutoff value is met:

Property module: material editor: **Mechanical->Elasticity->Low Density Foam**: toggle on **Remove elements exceeding maximum**

Thermal Expansion

Only isotropic thermal expansion is permitted with the low-density foam material model.

The elastic volume ratio, J^{el}, relates the total volume ratio (current volume/reference volume), J, and the thermal volume ratio, J^{th}, via the simple relationship:

$$J^{el} = \frac{J}{J^{th}}.$$

J^{th} is given by

$$J^{th} = \left(1 + \varepsilon^{th}\right)^3,$$

where ε^{th} is the linear thermal expansion strain that is obtained from the temperature and the isotropic thermal expansion coefficient (*Thermal Expansion*).

Material Stability

The Drucker stability condition for a compressible material requires that the change in the Kirchhoff stress, $d\boldsymbol{\tau}$, following from an infinitesimal change in the logarithmic strain, $d\boldsymbol{\varepsilon}$, satisfies the inequality

$$d\boldsymbol{\tau} : d\boldsymbol{\varepsilon} > 0,$$

where the Kirchhoff stress $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. Using $d\boldsymbol{\tau} = \mathbf{D} : d\boldsymbol{\varepsilon}$, the inequality becomes

$$d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0.$$

This restriction requires that the tangential material stiffness $\mathbf{D}$ be positive definite.

For an isotropic elastic formulation the inequality can be represented in terms of the principal stresses and strains

$$d\tau_1 d\varepsilon_1 + d\tau_2 d\varepsilon_2 + d\tau_3 d\varepsilon_3 > 0.$$

Thus, the relation between changes in the stress and changes in the strain can be obtained in the form of the matrix equation

$$\begin{Bmatrix} d\tau_1 \\ d\tau_2 \\ d\tau_3 \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{Bmatrix} d\varepsilon_1 \\ d\varepsilon_2 \\ d\varepsilon_3 \end{Bmatrix},$$

where $D_{ij} = \delta\tau_i/\delta\varepsilon_j$. Since $\mathbf{D}$ must be positive definite, it is necessary that

$$D_{11} + D_{22} + D_{33} > 0,$$
$$D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0,$$
$$\det(\mathbf{D}) > 0.$$

When Poisson's ratio is zero, the off diagonal terms of $\mathbf{D}$ become zero. In that case the necessary conditions for a positive definite matrix reduce to $D_{ii} > 0$; that is, the slope of the specified uniaxial stress-strain curves in the space of Kirchhoff stress versus logarithmic strain must be positive.

You should be careful defining the input data for the low-density foam model to ensure stable material response for all strain rates. If an instability is found, Abaqus issues a warning message and prints the lowest value of strain for which the instability is observed. Ideally, no instability should occur. If instabilities are observed at strain levels that are likely to occur in the analysis, it is strongly recommended that you carefully examine and revise the material input data. When nonzero Poisson effects are defined, it is highly recommended that you provide uniaxial test data in tension and compression for the same range of strain rates.

Elements

The low-density foam model can be used with solid (continuum) elements and generalized plane strain elements. One-dimensional solid elements (truss and rebar) are also available for the case when no lateral strains are specified (Poisson's ratio is zero). The model cannot be used with shell elements or membrane elements.

Procedures

The low-density foam model must always be used with geometrically nonlinear analyses (*General and Perturbation Procedures*).

Superelasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Linear Elastic Behavior*
- **ELASTIC*
- **SUPERELASTIC*
- *Defining a superelasticity model*

Overview

The superelasticity model provided in Abaqus:

- can be used to model the stress-induced transformation of austenite to single variant martensite, the stress-induced transformation of martensite to austenite, reorientation of martensite, and the shape setting process; and
- is based on the work of Auricchio and Taylor (1996, 1997).

Material Behavior

The superelastic model is based on the uniaxial stress-strain response of phase transforming materials (*Figure 1*). Such materials (for example, Nitinol) are in the austenite phase under no loading conditions. Austenite is assumed to follow isotropic linear elasticity. On loading the material, the austenite phase starts transforming into martensite beyond a certain stress. Martensite is also assumed to follow isotropic linear elasticity. During the phase transformation, elastic properties are calculated from the elastic constants of austenite and martensite, following the rule of mixtures:

$$E = E_A + \zeta(E_M - E_A),$$
$$\nu = \nu_A + \zeta(\nu_M - \nu_A),$$

where ζ is the fraction of martensite, E_A is the Young's modulus of austenite, E_M is the Young's modulus of martensite, ν_A is the Poisson's ratio of austenite, and ν_M is the Poisson's ratio of martensite. After a certain stress, austenite is completely transformed into martensite, which deforms elastically thereafter. Therefore, the deformation follows the elastic constants of austenite when the fraction of martensite is zero and follows the elastic constants of martensite if the fraction of martensite is one (full transformation). On unloading, martensite transforms back into austenite and the transformation strain is fully recovered. However, the stress at which the reverse transformation occurs is different from the stress at which the austenite to martensite transformation occurred.

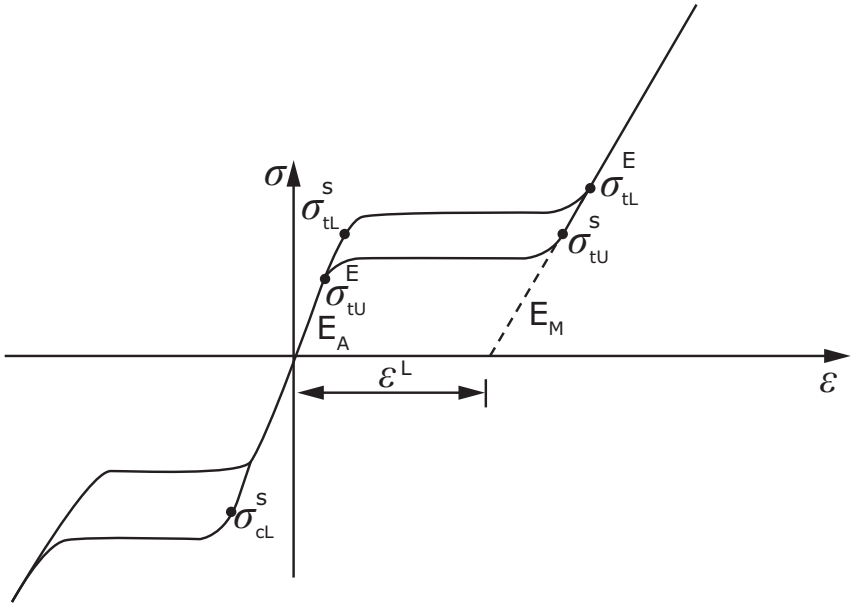


Figure 1: Uniaxial response of a superelastic material.

In this model the total strain increment, $\Delta\boldsymbol{\varepsilon}$, is assumed to be the sum of the elastic strain increment, $\Delta\boldsymbol{\varepsilon}^{el}$, and the increment in transformation strain, $\Delta\boldsymbol{\varepsilon}^{tr}$:

$$\Delta\boldsymbol{\varepsilon} = \Delta\boldsymbol{\varepsilon}^{el} + \Delta\boldsymbol{\varepsilon}^{tr}.$$

The increment in transformation strain is calculated using the following flow rule:

$$\Delta\boldsymbol{\varepsilon}^{tr} = \Delta\zeta\frac{\partial G^{tr}}{\partial\boldsymbol{\sigma}},$$

where G^{tr} is the transformation flow potential. The transformation potential is assumed to follow the Drucker-Prager form:

$$G^{tr} = q - p\,\tan\psi,$$

where $p = -\frac{1}{3}\,\mathbf{trace}\,\boldsymbol{\sigma}$ is the equivalent pressure stress, and $q = \sqrt{\frac{3}{2}\boldsymbol{S}:\boldsymbol{S}}$ is the Mises equivalent stress.

The transformation surface, F^{tr}, is also assumed to follow the Drucker-Prager form. It varies linearly with temperature, $\boldsymbol{T}$:

$$F^{tr} = q - p\,\tan\beta.$$

In the case of tensile loading, F^{tr} should lie between σ^{S}_{tL} and σ^{E}_{tL}; during unloading, F^{tr} should be between σ^{S}_{tU} and σ^{E}_{tU}. A difference in the response of the specimen during loading in tension and compression can be obtained by specifying a value of σ^{S}_{cL}, which is different from σ^{S}_{tL}. The angles β and ψ are calculated from the tensile and compressive transformation stress levels, the uniaxial transformation strain, and the user-specified volumetric transformation strain. The transformation stress levels must be positive for physically meaningful and converged results. For example, if the unloading transformation stress levels become negative, unrealistic results (due to incomplete reverse transformation) or even nonconvergence can occur during unloading. Nonconvergence can also occur if the transformation is not fully reversed by the time q reaches 0, which could happen for tensile hydrostatic pressures $p < -\sigma^{E}_{tU}\left(1/\tan\beta + 1/3\right)$. It is recommended that you avoid specifying parameters leading to a large value of $\tan\beta$ that can compromise the performance of the model under hydrostatic tension.

You can specify the stress levels at which the transformation takes place at the reference temperature, T_0. These stress levels are assumed to vary linearly with temperature, as shown in *Figure 2*. The stress levels should stay positive for all operating temperatures.

The material data cannot be specified as a function of temperature or field variables.

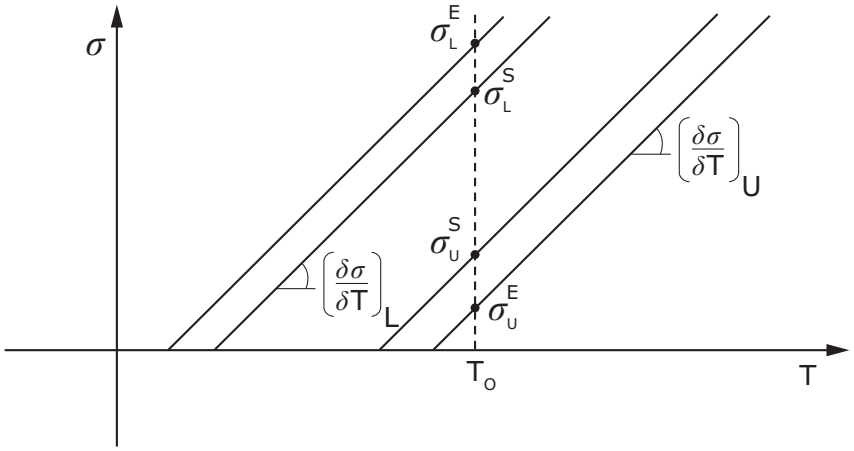


Figure 2: Stress vs. temperature curve.

Defining the Elastic Constants of Austenite

You can define the Young's modulus and Poisson's ratio of austenite.

Input File Usage: Use the following option to define the elastic constants of austenite:

**ELASTIC*

Abaqus/CAE Usage: Use the following option to define the elastic constants of austenite:

Property module: material editor: **Mechanical->Elasticity**: **Elastic**

Defining the Properties of Martensite and the Transformation Plateau

You can define the elastic properties of martensite, the critical stress levels for forward and reverse transformation, and the variation of transformation plateau with temperature.

Input File Usage: Use the following option to define the elastic properties of martensite and the transformation plateau:

**SUPERELASTIC*

Abaqus/CAE Usage: Use the following option to define the elastic properties of martensite and the transformation plateau:

Property module: material editor: **Mechanical->Superelasticity**

Nonassociated Transformation Flow

If the value of the volumetric transformation strain is not equal to the uniaxial transformation strain, Abaqus uses a nonassociated flow rule. This approach results in an unsymmetric material stiffness matrix. In such cases, Abaqus/Standard automatically activates the unsymmetric solution scheme. If desired, you can turn off the unsymmetric solution scheme for a particular step (see *Defining an Analysis*).

Input File Usage: Use the following option to use the nonassociated transformation flow rule:

**SUPERELASTIC*, NONASSOCIATED= ε_V^L

Abaqus/CAE Usage: Use the following option to use the nonassociated transformation flow rule:

Property module: material editor: **Mechanical->Superelasticity**: toggle on **Nonassociated**, and enter the volumetric transformation strain, ε_V^L

Shape Setting

Shape setting is done by setting all the state variables to zero. The procedure varies between Abaqus/Standard and Abaqus/Explicit. In Abaqus/Standard you must remove and reactivate elements. In Abaqus/Explicit you only need to specify the annealing process.

Input File Usage: Use the following options to define shape setting in Abaqus/Standard:

**SUPERELASTIC*
**MODEL CHANGE*, REMOVE
**MODEL CHANGE*, ADD=STRAIN FREE

Use the following options to define shape setting in Abaqus/Explicit:

**SUPERELASTIC*
**ANNEAL*

Abaqus/CAE Usage: Use the following options to define shape setting in Abaqus/Standard:

Property module: material editor: **Mechanical->Superelasticity**
Interaction module: **Create Interaction**: **Model Change**: **Definition**: **Region, Activation state of region elements: Deactivated in this step**
Interaction module: **Create Interaction**: **Model Change**: **Definition**: **Region, Activation state of region elements: Reactivated in this step**

Use the following options to define shape setting in Abaqus/Explicit:

Property module: material editor: **Mechanical->Superelasticity**
Step module: **Create Step**: **General**: **Anneal**

Elements

Superelasticity can be used with any stress/displacement element in Abaqus.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables are available for the superelasticity model:

TE	Transformation strain tensor.
TEEQ	Equivalent transformation strain.
TEVOL	Volumetric transformation strain.
MVF	Fraction of martensite.
SEQUT	Equivalent uniaxial tensile stress.
EEQUT	Equivalent uniaxial tensile total strain.
EE	Elastic strain tensor.

References

- Auricchio, F., and R.L. Taylor, "Shape-Memory Alloys: Modeling and Numerical Simulations of the Finite-Strain Superelastic Behavior," *Computer Methods in Applied Mechanics and Engineering*, vol. 143, pp. 175–194, 1996.
- Auricchio, F., and J. Lubliner, "Shape-Memory Alloys: Macromodelling and Numerical Simulations of the Superelastic Behavior," *Computer Methods in Applied Mechanics and Engineering*, vol. 146, pp. 281–312, 1997.

Inelastic Mechanical Properties

In this section:

- *Inelastic Behavior*
- *Metal Plasticity*
- *Other Plasticity Models*
- *Fabric Material Behavior*
- *Jointed Material Model*
- *Concrete*
- *Permanent Set in Rubberlike Materials*
- *Plasticity Model for Superelastic Materials*
- *Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials*
- *Paperboard Material Model*

Inelastic Behavior

Overview

The material library in Abaqus includes several models of inelastic behavior.

Classical metal plasticity

You can typically use the classical metal plasticity models (*Classical Metal Plasticity*) to describe the yield and inelastic flow of a metal at relatively low temperatures, where loading is relatively monotonic and creep effects are not important. In Abaqus these models use standard Mises or quadratic Hill yield surfaces with associated plastic flow. Perfect plasticity and isotropic hardening definitions are both available in the classical metal plasticity models. Common applications include crash analyses, metal forming, and general collapse studies; the models are simple and adequate for such cases.

Models for metals subjected to cyclic loading

You can use a linear kinematic hardening model or a nonlinear isotropic/kinematic hardening model (*Models for Metals Subjected to Cyclic Loading*) in Abaqus to simulate the behavior of materials that are subjected to cyclic loading. The evolution law in these models consists of:

- a kinematic hardening component (which describes the translation of the yield surface in stress space),
- and, for the nonlinear isotropic/kinematic hardening model, of an isotropic component (which describes the change of the elastic range).

You can simulate the Bauschinger effect and plastic shakedown with both models, but the nonlinear isotropic/kinematic hardening model provides more accurate predictions. Ratcheting and relaxation of the mean stress are accounted for only by the nonlinear isotropic/kinematic model. In addition to these two models, you can use the ORNL model in Abaqus/Standard if you require a simple life estimation for the design of stainless steels subjected to low-cycle fatigue and creep fatigue (see below).

Rate-dependent yield

As strain rates increase, many materials show an increase in their yield strength. You can define rate dependence (*Rate-Dependent Yield*) in Abaqus for a number of plasticity models. Rate dependence is available in both static and dynamic procedures. Applicable models include classical metal plasticity, extended Drucker-Prager plasticity, and crushable foam plasticity.

Creep and swelling

Abaqus/Standard provides a material model for classical metal creep behavior and time-dependent volumetric swelling behavior (*Rate-Dependent Plasticity: Creep and Swelling*). This model is intended for relatively slow (quasi-static) inelastic deformation of a model such as the high-temperature creeping flow of a metal or a piece of glass. The creep strain rate is assumed to be purely deviatoric, meaning that there is no volume change associated with this part of the inelastic straining. You can use creep with the classical metal plasticity model, with the ORNL model, and to define rate-dependent gasket behavior (*Defining the Gasket Behavior Directly Using a Gasket Behavior Model*). Swelling is available with the classical metal plasticity model. (Usage with the Drucker-Prager models is explained below.)

Annealing or melting

Abaqus provides a modeling capability for situations in which a loss of memory related to hardening occurs above a certain user-defined temperature, known as the annealing temperature (*Annealing or Melting*). It is intended for use with metals subjected to high-temperature deformation processes, in which the material might undergo melting and possibly resolidification or some other form of annealing. In Abaqus, you can model annealing or melting with classical metal plasticity (Mises and Hill); in Abaqus/Explicit, you can also model annealing or melting with Johnson-Cook plasticity. The annealing temperature is assumed to be a material property. See *Annealing* for information on an alternative method for simulating annealing in Abaqus/Explicit.

Hill anisotropic yield and creep

Abaqus provides an anisotropic yield model using Hill's quadratic potential (*Hill Anisotropic Yield/Creep*), which is available for use with materials modeled with classical metal plasticity (*Classical Metal Plasticity*), kinematic hardening (*Models for Metals Subjected to Cyclic Loading*), and creep (*Rate-Dependent Plasticity: Creep and Swelling*) that exhibit different yield stresses in different directions. The Abaqus/Standard model includes creep; creep behavior is not available in Abaqus/Explicit. The model allows for the specification of different stress ratios for each stress component to define the initial anisotropy. The model is not adequate for cases in which the anisotropy changes significantly as the material deforms because of loading.

Nonquadratic yield

Abaqus provides yield models using nonquadratic stress potential functions (*Nonquadratic Yield*), such as Tresca and Hosford yield surfaces for isotropic yield and Barlat yield surfaces for anisotropic yield. Compared with Hill anisotropic plasticity, Barlat anisotropic plasticity provides flexibility for describing more complex anisotropic yield behavior. Abaqus uses the anisotropy coefficients specified for Barlat yield surfaces to define the initial anisotropy. The model is not adequate for cases in which the anisotropy changes significantly as the material deforms because of loading.

Johnson-Cook plasticity

The Johnson-Cook plasticity model in Abaqus/Explicit (*Johnson-Cook Plasticity*) is particularly suited to model high-strain-rate deformation of metals. This model is a particular type of Mises plasticity that includes analytical forms of the hardening law and rate dependence. It is generally used in adiabatic transient dynamic analysis.

Dynamic failure models

Two types of dynamic failure models are available in Abaqus/Explicit for the Mises and Johnson-Cook plasticity models (*Dynamic Failure Models*):

- the shear failure model, where the failure criterion is based on the accumulated equivalent plastic strain,
- and the tensile failure model, which uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff.

Both models offer a number of failure choices including element removal and are applicable mainly in truly dynamic situations. In contrast, the progressive failure and damage models (*Progressive Damage and Failure*) are suitable for both quasi-static and dynamic situations and have other significant advantages.

Porous metal plasticity

The porous metal plasticity model (*Porous Metal Plasticity*) models materials that exhibit damage in the form of void initiation and growth. You can also use it for some powder metal process simulations at high relative densities (relative density is defined as the ratio of the volume of solid material to the total volume of the material).

The model is based on Gurson's porous metal plasticity theory with void nucleation and is intended for use with materials that have a relative density that is greater than 0.9. The model is adequate for relatively monotonic loading.

Cast iron plasticity

The cast iron plasticity model (*Cast Iron Plasticity*) is used to model gray cast iron, which exhibits markedly different inelastic behavior in tension and compression. The microstructure of gray cast iron consists of a distribution of graphite flakes in a steel matrix. In tension, the graphite flakes act as stress concentrators; while in compression, the flakes serve to transmit stresses. The resulting material is brittle in tension, but in compression it is similar in behavior to steel. The differences in tensile and compressive plastic response include: (i) a yield stress in tension that is three to five times lower than the yield stress in compression; (ii) permanent volume increase in tension, but negligible inelastic volume change in compression; (iii) different hardening behavior in tension and compression. The model is adequate for relatively monotonic loading.

Two-layer viscoplasticity

The two-layer viscoplasticity model in Abaqus/Standard (*Two-Layer Viscoplasticity*) is useful for modeling materials in which significant time-dependent behavior and plasticity occur. For metals this behavior typically occurs at elevated temperatures. The model has been shown to provide good results for thermomechanical loading.

ORNL constitutive model

The ORNL plasticity model in Abaqus/Standard (*ORNL – Oak Ridge National Laboratory Constitutive Model*) is intended for cyclic loading and high-temperature creep of type 304 and 316 stainless steel. Plasticity and creep calculations are provided according to the specification in Nuclear Standard NEF 9-5T, “Guidelines and Procedures for Design of Class I Elevated Temperature Nuclear System Components.” This model is an extension of the linear kinematic hardening model (discussed above), which attempts to provide for simple life estimation for design purposes when low-cycle fatigue and creep fatigue are critical issues.

Deformation plasticity

Abaqus/Standard provides a deformation theory Ramberg-Osgood plasticity model (*Deformation Plasticity*) for use in developing fully plastic solutions for fracture mechanics applications in ductile metals. The model is most commonly applied in static loading with small-displacement analysis for which the fully plastic solution must be developed in a part of the model.

Extended Drucker-Prager plasticity and creep

The extended Drucker-Prager family of plasticity models (*Extended Drucker-Prager Models*) describes the behavior of granular materials or polymers in which the yield behavior depends on the equivalent pressure stress. The inelastic deformation can sometimes be associated with frictional mechanisms such as sliding of particles across each other.

This class of models provides a choice of three different yield criteria. The differences in criteria are based on the shape of the yield surface in the meridional plane, which can be a linear form, a hyperbolic form, or a general exponent form. Inelastic time-dependent (creep) behavior coupled with the plastic behavior is also available in Abaqus/Standard for the linear form of the model. Creep behavior is not available in Abaqus/Explicit.

Modified Drucker-Prager/Cap plasticity and creep

The modified Drucker-Prager/Cap plasticity model (*Modified Drucker-Prager/Cap Model*) can be used to simulate geological materials that exhibit pressure-dependent yield. The addition of a cap yield surface helps control volume dilatancy when the material yields in shear and provides an inelastic hardening mechanism to represent plastic compaction. In Abaqus/Standard, inelastic time-dependent (creep) behavior coupled with the plastic behavior is also available for this model, with two possible creep mechanisms are possible: a cohesion, Drucker-Prager–like mechanism; and a consolidation, cap-like mechanism.

Mohr-Coulomb plasticity

The Mohr-Coulomb plasticity model (*Mohr-Coulomb Plasticity*) can be used for design applications in the geotechnical engineering area. The model uses the classical Mohr-Coloumb yield criterion: a straight line in the meridional plane and an irregular hexagonal section in the deviatoric plane. However, the Abaqus Mohr-Coulomb model has a completely smooth flow potential instead of the classical hexagonal pyramid: the flow potential is a hyperbola in the meridional plane, and it uses the smooth deviatoric section proposed by Menétrey and Willam.

Critical state (clay) plasticity

The clay plasticity model (*Critical State (Clay) Plasticity Model*) describes the inelastic response of cohesionless soils. The model provides a reasonable match to the experimentally observed behavior of saturated clays. This model defines the inelastic behavior of a material by a yield function that depends on the three stress invariants, an associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain.

Crushable foam plasticity

The foam plasticity model (*Crushable Foam Plasticity Models*) is intended for modeling crushable foams that are typically used as energy absorption structures; however, other crushable materials such as balsa wood can also be simulated with this model. This model is most appropriate for relatively monotonic loading. The crushable foam model with isotropic hardening is applicable to polymeric foams as well as metallic foams.

Jointed material

The jointed material model in Abaqus/Standard (*Jointed Material Model*) is intended to provide a simple, continuum model for a material that contains a high density of parallel joint surfaces in different orientations, such as sedimentary rock. This model is intended for applications where stresses are mainly compressive, and it provides a joint opening capability when the stress normal to the joint tries to become tensile.

Concrete

Three different constitutive models are available in Abaqus for the analysis of concrete at low confining pressures: the smeared crack concrete model in Abaqus/Standard (*Concrete Smeared Cracking*); the brittle cracking model in Abaqus/Explicit (*Cracking Model for Concrete*); and the concrete damaged plasticity model in both Abaqus/Standard and Abaqus/Explicit (*Concrete Damaged Plasticity*). Each model is designed to provide a general capability for modeling plain and reinforced concrete (as well as other similar quasi-brittle materials) in all types of structures: beams, trusses, shells, and solids.

The smeared crack concrete model in Abaqus/Standard is intended for applications in which the concrete is subjected to essentially monotonic straining and a material point exhibits either tensile cracking or compressive crushing. Plastic straining in compression is controlled by a “compression” yield surface. Cracking is assumed to be the most important aspect of the behavior, and the representation of cracking and postcracking anisotropic behavior dominates the modeling.

The brittle cracking model in Abaqus/Explicit is intended for applications in which the concrete behavior is dominated by tensile cracking and compressive failure is not important. The model includes consideration of the anisotropy induced by cracking. In compression, the model assumes elastic behavior. A simple brittle failure criterion is available to allow the removal of elements from a mesh.

The concrete damaged plasticity model in Abaqus/Standard and Abaqus/Explicit is based on the assumption of scalar (isotropic) damage and is designed for applications in which the concrete is subjected to arbitrary loading conditions, including cyclic loading. The model takes into consideration the degradation of the elastic stiffness induced by plastic straining both in tension and compression. It also accounts for stiffness recovery effects under cyclic loading.

Progressive damage and failure

Abaqus/Explicit offers a general capability for modeling progressive damage and failure in ductile metals and fiber-reinforced composites (*Progressive Damage and Failure*).

References:

Plasticity Theories

Most materials of engineering interest initially respond elastically. Elastic behavior means that the deformation is fully recoverable: when the load is removed, the specimen returns to its original shape. If the load exceeds some limit (the “yield load”), the deformation is no longer fully recoverable. Some part of the deformation remains when the load is removed; for example, when a paperclip is bent too much or when a billet of metal is rolled or forged in a manufacturing process. Plasticity theories model the material's mechanical response as it undergoes such nonrecoverable deformation in a ductile fashion. The theories have been developed most intensively for metals, but they are also applied to soils, concrete, rock, ice, crushable foam, and other materials. These materials behave in very different ways. For example, large values of pure hydrostatic pressure cause very little inelastic deformation in metals, but quite small hydrostatic pressure values can cause a significant, nonrecoverable volume change in a soil sample. Nonetheless, the fundamental concepts of plasticity theories are sufficiently general that models based on these concepts have been developed successfully for a wide range of materials.

Most of the plasticity models in Abaqus are “incremental” theories in which the mechanical strain rate is decomposed into an elastic part and a plastic (inelastic) part. Incremental plasticity models are usually formulated in terms of

- a *yield surface*, which generalizes the concept of “yield load” into a test function that can be used to determine if the material has a purely elastic response at a particular state of stress, temperature, and other values;
- a *flow rule*, which defines the inelastic deformation that occurs if the material point is no longer responding purely elastically; and
- evolution laws that define the *hardening*—the way in which the yield or flow definitions change as inelastic deformation occurs.

Abaqus/Standard also has a “deformation” plasticity model, in which the stress is defined from the total mechanical strain. This is a Ramberg-Osgood model (*Deformation Plasticity*) and is intended primarily for ductile fracture mechanics applications, where fully plastic solutions are often required.

Elastic Response

The Abaqus plasticity models also need an elasticity definition to deal with the recoverable part of the strain. In Abaqus the elasticity is defined by including linear elastic behavior or, if relevant for some plasticity models, porous elastic behavior in the same material definition (see *Material Data Definition*). In the case of the Mises

and Johnson-Cook plasticity models in Abaqus/Explicit the elasticity can alternatively be defined using an equation of state with associated deviatoric behavior (see *Equation of State*).

When performing an elastic-plastic analysis at finite strains, Abaqus assumes that the plastic strains dominate the deformation and that the elastic strains are small. This restriction is imposed by the elasticity models that Abaqus uses. It is justified because most materials have a well-defined yield point that is a very small percentage of their Young's modulus; for example, the yield stress of metals is typically less than 1% of the Young's modulus of the material. Therefore, the elastic strains will also be less than this percentage, and the elastic response of the material can be modeled quite accurately as being linear.

In Abaqus/Explicit the elastic strain energy reported is updated incrementally. The incremental change in elastic strain energy (ΔE_s) is computed as $\Delta E_s = \Delta E_t - \Delta E_p$, where ΔE_t is the incremental change in total strain energy and ΔE_p is the incremental change in plastic energy dissipation. ΔE_s is much smaller than ΔE_t and ΔE_p for increments in which the deformation is almost all plastic. Approximations in the calculations of ΔE_t and ΔE_p result in deviations from the true solutions that are insignificant compared to ΔE_t and ΔE_p but can be significant relative to ΔE_s. Typically, the elastic strain energy solution is quite accurate, but in some rare cases the approximations in the calculations of ΔE_t and ΔE_p can lead to a negative value reported for the elastic strain energy. These negative values are most likely to occur in an analysis that uses rate-dependent plasticity. If the absolute value of the elastic strain energy is very small compared to the total strain energy, a negative value for the elastic strain energy should not be considered an indication of a serious solution problem.

Stress and Strain Measures

Most materials that exhibit ductile behavior (large inelastic strains) yield at stress levels that are orders of magnitude less than the elastic modulus of the material, which implies that the relevant stress and strain measures are “true” stress (Cauchy stress) and logarithmic strain. Material data for all of these models should, therefore, be given in these measures.

If you have nominal stress-strain data for a uniaxial test and the material is isotropic, a simple conversion to true stress and logarithmic plastic strain is

$$\sigma_{\text{true}} = \sigma_{\text{nom}} \left(1 + \varepsilon_{\text{nom}}\right),$$
$$\varepsilon_{\text{ln}}^{\text{pl}} = \ln(1 + \varepsilon_{\text{nom}}) - \frac{\sigma_{\text{true}}}{E},$$

where E is the Young's modulus.

Example of Stress-Strain Data Input

The example below illustrates the input of material data for the classical metal plasticity model with isotropic hardening (*Classical Metal Plasticity*). Stress-strain data representing the material hardening behavior are necessary to define the model. An experimental hardening curve might appear as that shown in *Figure 1*.

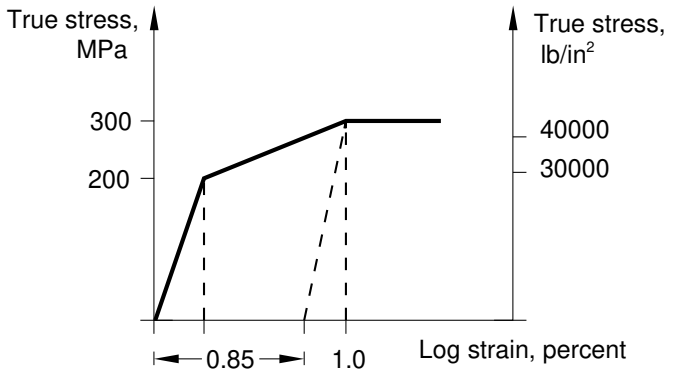


Figure 1: Experimental hardening curve.

First yield occurs at 200 MPa (29000 lb/in^2). The material then hardens to 300 MPa (43511 lb/in^2) at one percent strain, after which it is perfectly plastic. Assuming that the Young's modulus is 200000 MPa (29×10^6 lb/in^2), the plastic strain at the one percent strain point is .01 − 300/200000=.0085. When the units are newtons and millimeters, the input is

Yield Stress	Plastic Strain
200.	0.
300.	.0085

Plastic strain values, not total strain values, are used in defining the hardening behavior. Furthermore, the first data pair must correspond with the onset of plasticity (the plastic strain value must be zero in the first pair). These concepts are applicable when hardening data are defined in a tabular form for any of the following plasticity models:

- *Classical Metal Plasticity*
- *Models for Metals Subjected to Cyclic Loading*
- *Porous Metal Plasticity* (isotropic hardening classical metal plasticity must be defined for use with this model)
- *Cast Iron Plasticity*
- *ORNL – Oak Ridge National Laboratory Constitutive Model*
- *Extended Drucker-Prager Models*
- *Modified Drucker-Prager/Cap Model*
- *Mohr-Coulomb Plasticity*
- *Critical State (Clay) Plasticity Model*
- *Crushable Foam Plasticity Models*
- *Concrete Smeared Cracking*

The input required to define hardening is discussed in the referenced sections.

Specifying Initial Equivalent Plastic Strains

Initial values of equivalent plastic strain can be specified in Abaqus for elements that use classical metal plasticity (*Classical Metal Plasticity*) or Drucker-Prager plasticity (*Extended Drucker-Prager Models*) by defining initial hardening conditions (*Initial Conditions*). The equivalent plastic strain (output variable PEEQ) then contains the initial value of equivalent plastic strain plus any additional equivalent plastic strain because of plastic straining

during the analysis. However, the plastic strain tensor (output variable PE) contains only the amount of straining due to deformation during the analysis.

The simple one-dimensional example shown in *Figure 2* illustrates the concept.

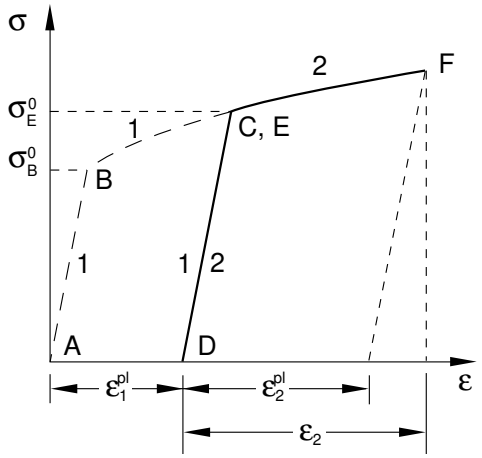


Figure 2: Initial equivalent plastic strain example.

The material is in an annealed configuration at point *A*; its yield stress is σ_B^0. It is then hardened by loading it along the path (A, B, C, D); the new yield stress is σ_E^0. A new analysis that uses the same hardening curve as the first analysis takes this material along the path (D, E, F), starting from point *D*, by specifying a total strain, ε_2. Plastic strain ε_2^{pl} will result and can be output (for example) using output variable PE11. To obtain the correct yield stress, σ_E^0, the equivalent plastic strain at point *E*, ε_1^{pl}, should be provided as an initial condition. Likewise, the correct yield stress at point *F* is obtained from an equivalent plastic strain PEEQ $= \varepsilon_1^{pl} + \varepsilon_2^{pl}$.

Metal Plasticity

In this section:

- *Classical Metal Plasticity*
- *Models for Metals Subjected to Cyclic Loading*
- *Rate-Dependent Yield*
- *Rate-Dependent Plasticity: Creep and Swelling*
- *Annealing or Melting*
- *Hill Anisotropic Yield/Creep*
- *Nonquadratic Yield*
- *Johnson-Cook Plasticity*
- *Dynamic Failure Models*
- *Porous Metal Plasticity*
- *Cast Iron Plasticity*
- *Two-Layer Viscoplasticity*
- *ORNL – Oak Ridge National Laboratory Constitutive Model*
- *Deformation Plasticity*

Classical Metal Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Rate-Dependent Yield*
- *Hill Anisotropic Yield/Creep*
- *Johnson-Cook Plasticity*
- *Progressive Damage and Failure*
- *Dynamic Failure Models*
- *About the Material Library*
- *Inelastic Behavior*
- *UHARD*
- **PLASTIC*
- **RATE DEPENDENT*
- **POTENTIAL*
- *Defining classical metal plasticity*

Overview

The classical metal plasticity models:

- use Mises or Hill yield surfaces with associated plastic flow, which allow for isotropic and anisotropic yield, respectively;
- use perfect plasticity or isotropic hardening behavior;
- can be used when rate-dependent effects are important;
- are intended for applications such as crash analyses, metal forming, and general collapse studies (Plasticity models that include kinematic hardening and are, therefore, more suitable for cases involving cyclic loading are also available in Abaqus: see *Models for Metals Subjected to Cyclic Loading*.);
- can be used in any procedure that uses elements with displacement degrees of freedom;
- can be used in a fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*), a fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*), or an adiabatic thermal-stress analysis (*Adiabatic Analysis*) such that plastic dissipation results in the heating of a material;
- can be used in conjunction with the models of progressive damage and failure in Abaqus (*About Damage and Failure for Ductile Metals*) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh;
- can be used in conjunction with the shear failure model in Abaqus/Explicit to provide a simple ductile dynamic failure criterion that allows for the removal of elements from the mesh, although the progressive damage and failure methods discussed above are generally recommended instead;
- can be used in conjunction with the tensile failure model in Abaqus/Explicit to provide a tensile spall criterion offering a number of failure choices and removal of elements from the mesh; and
- must be used in conjunction with either the linear elastic material model (*Linear Elastic Behavior*) or the equation of state material model (*Equation of State*).

Yield Surfaces

The Mises and Hill yield surfaces assume that yielding of the metal is independent of the equivalent pressure stress: this observation is confirmed experimentally for most metals (except voided metals) under positive pressure stress but may be inaccurate for metals under conditions of high triaxial tension when voids may nucleate and grow in the material. Such conditions can arise in stress fields near crack tips and in some extreme thermal loading cases such as those that might occur during welding processes. A porous metal plasticity model is provided in Abaqus for such situations. This model is described in *Porous Metal Plasticity*.

Mises Yield Surface

The Mises yield surface is used to define isotropic yielding. It is defined by giving the value of the uniaxial yield stress as a function of uniaxial equivalent plastic strain, temperature, and/or field variables. In Abaqus/Standard the yield stress can alternatively be defined in user subroutine *UHARD*.

Input File Usage: **PLASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**

Hill Yield Surface

The quadratic Hill yield surface allows anisotropic yielding to be modeled. You must specify a reference yield stress, σ^0, for the metal plasticity model and define a set of yield ratios, R_{ij}, separately. These data define the yield stress corresponding to each stress component as $R_{ij}\sigma^0$. Hill's potential function is discussed in detail in *Hill Anisotropic Yield/Creep*. Yield ratios can be used to define three common forms of anisotropy associated with sheet metal forming: transverse anisotropy, planar anisotropy, and general anisotropy.

The plasticity model using the Hill yield surface is applicable to strains up to about 25%–30%, and it is not recommended for analyses in which these values are exceeded.

Input File Usage: Use both of the following options:

**PLASTIC (to specify the reference yield stress, σ^0)*

**POTENTIAL*, TYPE=HILL (default) *(to specify the yield ratios, R_{ij})*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Potential**

Hardening

In Abaqus a perfectly plastic material (with no hardening) can be defined, or work hardening can be specified. Isotropic hardening, including Johnson-Cook hardening, is available in both Abaqus/Standard and Abaqus/Explicit. In addition, Abaqus provides kinematic hardening for materials subjected to cyclic loading.

Perfect Plasticity

Perfect plasticity means that the yield stress does not change with plastic strain. It can be defined in tabular form for a range of temperatures and/or field variables; a single yield stress value per temperature and/or field variable specifies the onset of yield.

Input File Usage: **PLASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**

Isotropic Hardening

Isotropic hardening means that the yield surface changes size uniformly in all directions such that the yield stress increases (or decreases) in all stress directions as plastic straining occurs. Abaqus provides an isotropic hardening model, which is useful for cases involving gross plastic straining or in cases where the straining at each point is essentially in the same direction in strain space throughout the analysis. Although the model is referred to as a "hardening" model, strain softening or hardening followed by softening can be defined. Isotropic hardening plasticity is discussed in more detail in *Isotropic elasto-plasticity*.

If isotropic hardening is defined, the yield stress, σ^0, can be given as a tabular function of equivalent plastic strain and, if required, of temperature and/or other predefined field variables. The yield stress at a given state is interpolated from this table of data, and it is assumed to remain constant outside the range of the independent variables other than equivalent plastic strain. Outside the range of equivalent plastic strains, you can choose if the yield stress remains constant (default) or is extrapolated linearly (see *Extrapolation of Material Data*).

Abaqus/Explicit regularizes the data into tables that are defined in terms of even intervals of the independent variables. In some cases where the yield stress is defined at uneven intervals of the independent variable (plastic strain) and the range of the independent variable is large compared to the smallest interval, Abaqus/Explicit may fail to obtain an accurate regularization of your data in a reasonable number of intervals. In this case the program will stop after all data are processed with an error message that you must redefine the material data. See *Material Data Definition* for a more detailed discussion of data regularization.

Input File Usage: Use the following option to define isotropic hardening with tabular data:

**PLASTIC*, HARDENING=ISOTROPIC (default if parameter is omitted)

Use the following option to request that the yield stress remains constant outside the specified range of the equivalent plastic strains:

**PLASTIC*, EXTRAPOLATION=CONSTANT (default)

Use the following option to request linear extrapolation of the yield stress outside the specified range of the equivalent plastic strains:

**PLASTIC*, EXTRAPOLATION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Isotropic**: **Extrapolation**: **Constant** or **Linear**

Johnson-Cook Isotropic Hardening

Johnson-Cook hardening is a particular type of isotropic hardening where the yield stress is given as an analytical function of equivalent plastic strain, strain rate, and temperature. This hardening law is suited for modeling high-rate deformation of many materials including most metals. Hill's potential function (see *Hill Anisotropic Yield/Creep*) cannot be used with Johnson-Cook hardening. For more details, see *Johnson-Cook Plasticity*.

Input File Usage: **PLASTIC*, HARDENING=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Johnson-Cook**

User Subroutine

In Abaqus/Standard the yield stress for isotropic hardening, σ^0, can alternatively be described through user subroutine *UHARD*.

Input File Usage: **PLASTIC*, HARDENING=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: User**

Kinematic Hardening

Three kinematic hardening models are provided in Abaqus to model the cyclic loading of metals. The linear kinematic model approximates the hardening behavior with a constant rate of hardening. The more general nonlinear isotropic/kinematic model will give better predictions but requires more detailed calibration. The multilinear kinematic model combines several piecewise linear hardening curves to predict the complex response of metals under thermomechanical load cycles. This model is based on Besseling (1958) and is available only in Abaqus/Standard. For more details, see *Models for Metals Subjected to Cyclic Loading*.

Input File Usage: Use the following option to specify the linear kinematic model:

**PLASTIC*, HARDENING=KINEMATIC

Use the following option to specify the nonlinear combined isotropic/kinematic model:

**PLASTIC*, HARDENING=COMBINED

Use the following option to specify the multilinear kinematic hardening model:

**PLASTIC*, HARDENING=MULTILINEAR KINEMATIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Kinematic** or **Multilinear-Kinematic** or **Combined**

Flow Rule

Abaqus uses associated plastic flow. Therefore, as the material yields, the inelastic deformation rate is in the direction of the normal to the yield surface (the plastic deformation is volume invariant). This assumption is generally acceptable for most calculations with metals; the most obvious case where it is not appropriate is the detailed study of the localization of plastic flow in sheets of metal as the sheet develops texture and eventually tears apart. So long as the details of such effects are not of interest (or can be inferred from less detailed criteria, such as reaching a forming limit that is defined in terms of strain), the associated flow models in Abaqus used with the smooth Mises or Hill yield surfaces generally predict the behavior accurately.

Rate Dependence

As strain rates increase, many materials show an increase in their yield strength. This effect becomes important in many metals when the strain rates range between 0.1 and 1 per second; and it can be very important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes.

There are multiple ways to introduce a strain-rate-dependent yield stress.

Direct Tabular Data

Test data can be provided as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates ($\dot{\bar{\varepsilon}}^{pl}$); one table per strain rate. Direct tabular data cannot be used with Johnson-Cook hardening. The guidelines that govern the entry of this data are provided in *Rate-Dependent Yield*.

Input File Usage: **PLASTIC*, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Use strain-rate-dependent data**

Yield Stress Ratios

Alternatively, you can specify the strain rate dependence by means of a scaling function. In this case you enter only one hardening curve, the static hardening curve, and then express the rate-dependent hardening curves in terms of the static relation; that is, we assume that

$$\bar{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}\right) = \sigma^0\left(\bar{\varepsilon}^{pl}\right) R\left(\dot{\bar{\varepsilon}}^{pl}\right),$$

where σ^0 is the static yield stress, $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, and R is a ratio, defined as $R = 1.0$ at $\dot{\bar{\varepsilon}}^{pl} = 0.0$. This method is described further in *Rate-Dependent Yield*.

Input File Usage: Use both of the following options:

**PLASTIC (to specify the static yield stress $\sigma^0\left(\bar{\varepsilon}^{pl}\right)$)*

**RATE DEPENDENT (to specify the ratio $R\left(\dot{\bar{\varepsilon}}^{pl}\right)$)*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Rate Dependent**

User Subroutine

In Abaqus/Standard user subroutine *UHARD* can be used to define a rate-dependent yield stress. You are provided the current equivalent plastic strain and equivalent plastic strain rate and are responsible for returning the yield stress and derivatives.

Input File Usage: **PLASTIC*, HARDENING=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: User**

Progressive Damage and Failure

In Abaqus the metal plasticity material models can be used in conjunction with the progressive damage and failure models discussed in *About Damage and Failure for Ductile Metals*. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and, in Abaqus/Explicit, Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations. This is a great advantage over the dynamic failure models discussed next.

Input File Usage: Use the following options:

**PLASTIC*
**DAMAGE INITIATION*
**DAMAGE EVOLUTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**

Shear and Tensile Dynamic Failure in Abaqus/Explicit

In Abaqus/Explicit the metal plasticity material models can be used in conjunction with the shear and tensile failure models (*Dynamic Failure Models*) that are applicable in truly dynamic situations; however, the progressive damage and failure models discussed above are generally preferred.

Shear Failure

The shear failure model provides a simple failure criterion that is suitable for high-strain-rate deformation of many materials including most metals. It offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The shear failure criterion is based on the value of the equivalent plastic strain and is applicable mainly to high-strain-rate, truly dynamic problems. For more details, see *Dynamic Failure Models*.

Input File Usage: Use both of the following options:

**PLASTIC*
**SHEAR FAILURE*

Abaqus/CAE Usage: The shear failure model is not supported in Abaqus/CAE.

Tensile Failure

The tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff. It offers a number of failure choices including element removal. Similarly to the shear failure model, the tensile failure model is suitable for high-strain-rate deformation of metals and is applicable to truly dynamic problems. For more details, see *Dynamic Failure Models*.

Input File Usage: Use both of the following options:

**PLASTIC*
**TENSILE FAILURE*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Tensile Failure**

Heat Generation by Plastic Work

Abaqus optionally allows for plastic dissipation to result in the heating of a material. Heat generation is typically used in the simulation of bulk metal forming or high-speed manufacturing processes involving large amounts of inelastic strain where the heating of the material caused by its deformation is an important effect because of temperature dependence of the material properties. It is applicable only to adiabatic thermal-stress analysis (*Adiabatic Analysis*), fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*), or fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*).

This effect is introduced by defining the fraction of the rate of inelastic dissipation that appears as a heat flux per volume.

Input File Usage: Use all of the following options in the same material data block:

**PLASTIC*
**SPECIFIC HEAT*
**DENSITY*
**INELASTIC HEAT FRACTION*

Abaqus/CAE Usage: Use all of the following options for the same material:

Property module: material editor:
Mechanical->Plasticity->Plastic
Thermal->Specific Heat
General->Density
Thermal->Inelastic Heat Fraction

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state (see *Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

User Subroutine Specification in Abaqus/Standard

For more complicated cases, initial conditions can be defined in Abaqus/Standard through user subroutine *HARDINI*.

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

Classical metal plasticity can be used with any elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the classical metal plasticity models:

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

YIELDS

Yield stress, σ^0.

References

- Besseling, J. F., "A Theory of Elastic, Plastic, and Creep Deformations of an Initially Isotropic Material Showing Anisotropic Strain-Hardening, Creep Recovery, and Secondary Creep," *Journal of Applied Mechanics*, pp. 529–536, 1958.

Models for Metals Subjected to Cyclic Loading

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Hill Anisotropic Yield/Creep*
- *UHARD*
- **CYCLIC HARDENING*
- **PLASTIC*
- **POTENTIAL*
- *Defining classical metal plasticity*

Overview

The kinematic hardening models:

- are used to simulate the inelastic behavior of materials that are subjected to cyclic loading;
- include a linear kinematic hardening model, a nonlinear isotropic/kinematic hardening model, and, in Abaqus/Standard, a multilinear kinematic hardening model;
- include a nonlinear isotropic/kinematic hardening model with multiple backstresses and, optionally, static recovery effects;
- can be used in any procedure that uses elements with displacement degrees of freedom;
- in Abaqus/Standard cannot be used in adiabatic analyses;
- can be used to model rate-dependent yield, except the multilinear kinematic hardening;
- can be used with creep and swelling in Abaqus/Standard, except the multilinear kinematic hardening model; and
- require the use of the linear elasticity material model to define the elastic part of the response.

Yield Surfaces

The kinematic hardening models used to model the behavior of metals subjected to cyclic loading are pressure-independent plasticity models; in other words, yielding of the metals is independent of the equivalent pressure stress. These models are suited for most metals subjected to cyclic loading conditions, except voided metals.

The linear kinematic hardening model can be used with the Mises or Hill yield surface. The nonlinear isotropic/kinematic model can be used only with the Mises yield surface in Abaqus/Standard and with the Mises or Hill yield surface in Abaqus/Explicit. The pressure-independent yield surface used in the linear kinematic hardening model and the nonlinear isotropic/kinematic hardening model is defined by the function

$$F = f(\boldsymbol{\sigma} - \boldsymbol{\alpha}) - \sigma^0 = 0,$$

where σ^0 is the yield stress and $f(\boldsymbol{\sigma} - \boldsymbol{\alpha})$ is the equivalent Mises stress or Hill's potential with respect to the backstress $\boldsymbol{\alpha}$. For example, the equivalent Mises stress is defined as

$$f(\boldsymbol{\sigma}-\boldsymbol{\alpha})=\sqrt{\tfrac{3}{2}\,(\mathbf{S}-\boldsymbol{\alpha}^{dev}):(\mathbf{S}-\boldsymbol{\alpha}^{dev})},$$

where $\mathbf{S}$ is the deviatoric stress tensor (defined as $\mathbf{S}=\boldsymbol{\sigma}+p\mathbf{I}$, where $\boldsymbol{\sigma}$ is the stress tensor, p is the equivalent pressure stress, and $\mathbf{I}$ is the identity tensor) and $\boldsymbol{\alpha}^{dev}$ is the deviatoric part of the backstress tensor.

The response in the multilinear kinematic hardening model is assumed to be a weighted sum of various elastic-perfectly plastic elements. Each of these elements, also referred to as subvolumes, uses the Mises yield surface with a different yield strength.

Flow Rule

The kinematic hardening models assume associated plastic flow. The assumption of associated plastic flow is acceptable for metals subjected to cyclic loading as long as microscopic details, such as localization of plastic flow occurring as a metal component ruptures due to cyclic fatigue loads, are not of interest.

Linear Kinematic Hardening Model and Nonlinear Isotropic/Kinematic Hardening Model

In the case of the linear kinematic hardening model and the nonlinear isotropic/kinematic hardening model:

$$\dot{\boldsymbol{\varepsilon}}^{pl}=\dot{\bar{\varepsilon}}^{pl}\,\frac{\partial F}{\partial\boldsymbol{\sigma}},$$

where $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the rate of plastic flow and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate.

The evolution of the equivalent plastic strain is obtained from the following equivalent plastic work expression:

$$\sigma^{0}\dot{\bar{\varepsilon}}^{pl}=\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{pl},$$

which yields $\dot{\bar{\varepsilon}}^{pl}=\sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl}:\dot{\boldsymbol{\varepsilon}}^{pl}}$ for isotropic Mises plasticity.

Multilinear Kinematic Hardening Model

The flow rule for the multilinear kinematic hardening model is as follows:

$$\dot{\varepsilon}^{pl}=\sum_{k=1}^{N}w_{k}\dot{\bar{\varepsilon}}_{k}^{pl}\,\frac{\partial F_{k}}{\partial\sigma_{k}},$$

where N is the total number of subvolumes and w_k, F_k, $\dot{\bar{\varepsilon}}_k^{pl}$, and σ_k are the weights, Mises yield surface, equivalent plastic strain rate, and the stress of the kth subvolume, respectively. Every subvolume follows an associated flow rule.

Hardening

The linear kinematic hardening model has a constant hardening modulus, and the nonlinear isotropic/kinematic hardening model has both nonlinear kinematic and nonlinear isotropic hardening components. The multilinear kinematic hardening model has a piecewise constant hardening modulus.

Linear Kinematic Hardening Model

The evolution law of this model consists of a linear kinematic hardening component that describes the translation of the yield surface in stress space through the backstress, $\boldsymbol{\alpha}$. When temperature dependence is omitted, this evolution law is the linear Ziegler hardening law

$$\dot{\boldsymbol{\alpha}}=C\tfrac{1}{\sigma^{0}}\,(\boldsymbol{\sigma}-\boldsymbol{\alpha})\,\dot{\bar{\varepsilon}}^{pl},$$

where $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate and C is the kinematic hardening modulus. In this model the equivalent stress defining the size of the yield surface, σ^0, remains constant, $\sigma^0 = \sigma|_0$, where $\sigma|_0$ is the equivalent stress defining the size of the yield surface at zero plastic strain.

Nonlinear Isotropic/Kinematic Hardening Model

The evolution law of this model consists of two components: a nonlinear kinematic hardening component, which describes the translation of the yield surface in stress space through the backstress, $\boldsymbol{\alpha}$; and an isotropic hardening component, which describes the change of the equivalent stress defining the size of the yield surface, σ^0, as a function of plastic deformation.

The kinematic hardening component is defined to be an additive combination of a purely kinematic term (linear Ziegler hardening law), a relaxation term (the *recall* or dynamic recovery term), and, optionally, a static recovery term. The last two terms introduce the nonlinearity. In addition, several kinematic hardening components (backstresses) can be superposed, which may considerably improve results in some cases. When temperature and field variable dependencies are omitted, the hardening laws for each backstress are

$$\dot{\boldsymbol{\alpha}}_k = C_k \frac{1}{\sigma^0} (\boldsymbol{\sigma} - \boldsymbol{\alpha}) \dot{\bar{\varepsilon}}^{pl} - \gamma_k \boldsymbol{\alpha}_k \dot{\bar{\varepsilon}}^{pl} - \xi_k \left(\frac{|\boldsymbol{\alpha}_k|}{\overline{R}_k} \right)^{m_k - 1} \boldsymbol{\alpha}_k ,$$

where

$$\left| \boldsymbol{\alpha}_k \right| = \sqrt{\tfrac{3}{2} \boldsymbol{\alpha}_k^{dev} : \boldsymbol{\alpha}_k^{dev}},$$

and the overall backstress is computed from the relation

$$\boldsymbol{\alpha} = \sum_{k=1}^{N} \boldsymbol{\alpha}_k,$$

where N is the number of backstresses, and C_k, γ_k, ξ_k, $\overline{R}_k$, and m_k are material parameters that must be calibrated from cyclic test data. C_k are the initial kinematic hardening moduli, and γ_k determine the rate at which the kinematic hardening moduli decrease with increasing plastic deformation. ξ_k, $\overline{R}_k$, and m_k control the static recovery term. The kinematic hardening law can be separated into a deviatoric part and a hydrostatic part; only the deviatoric part has an effect on the material behavior. When ξ_k, C_k, and γ_k are zero, the model reduces to an isotropic hardening model. When all ξ_k and γ_k equal zero, the linear Ziegler hardening law is recovered. Calibration of the material parameters is discussed in *Usage and Calibration of the Kinematic Hardening Models* below. *Figure 1* shows an example of nonlinear kinematic hardening with three backstresses.

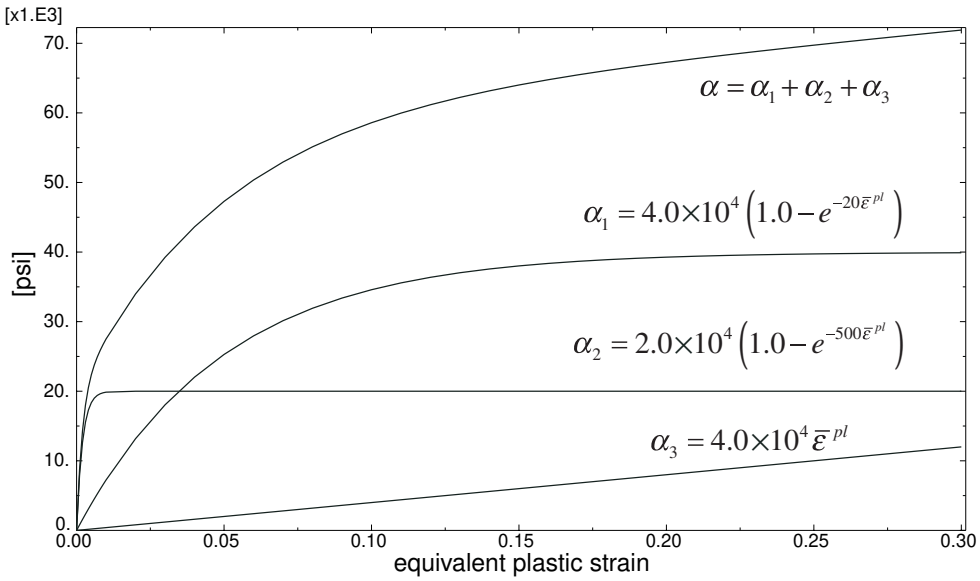


Figure 1: Kinematic hardening model with three backstresses.

Each of the backstresses covers a different range of strains, and the linear hardening law is retained for large strains.

The isotropic hardening behavior of the model defines the evolution of the yield surface size, σ^0, as a function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$. This evolution can be introduced by specifying σ^0 directly as a function of $\bar{\varepsilon}^{pl}$ in tabular form, by specifying σ^0 in user subroutine *UHARD* (in Abaqus/Standard only), or by using the simple exponential law

$$\sigma^0 = \sigma|_0 + Q_\infty\left(1 - e^{-b\,\bar{\varepsilon}^{pl}}\right),$$

where $\sigma|_0$ is the yield stress at zero plastic strain and Q_∞ and b are material parameters. Q_∞ is the maximum change in the size of the yield surface, and b defines the rate at which the size of the yield surface changes as plastic straining develops. When the equivalent stress defining the size of the yield surface remains constant ($\sigma^0 = \sigma|_0$), the model reduces to a nonlinear kinematic hardening model.

The evolution of the kinematic and the isotropic hardening components is illustrated in *Figure 2* for unidirectional loading and in *Figure 3* for multiaxial loading. The evolution law for the kinematic hardening component implies that the backstress is contained within a cylinder of radius $\sqrt{2/3}\,\alpha^s = \sqrt{2/3}\sum_k^N C_k/\gamma_k$, where α^s is the magnitude of $\boldsymbol{\alpha}$ at saturation (large plastic strains). It also implies that any stress point must lie within a cylinder of radius $\sqrt{2/3}\,\sigma_{max}$ (using the notation of *Figure 2*) since the yield surface remains bounded. At large plastic strain any stress point is contained within a cylinder of radius $\sqrt{2/3}\,(\alpha^s + \sigma^s)$, where σ^s is the equivalent stress defining the size of the yield surface at large plastic strain. If tabular data are provided for the isotropic component, σ^s is the last value given to define the size of the yield surface. If user subroutine *UHARD* is used, this value will depend on your implementation; otherwise, $\sigma^s = \sigma|_0 + Q_\infty$.

The model is applicable to strains up to about 25%–30%, and it is not recommended for analyses in which these values are exceeded.

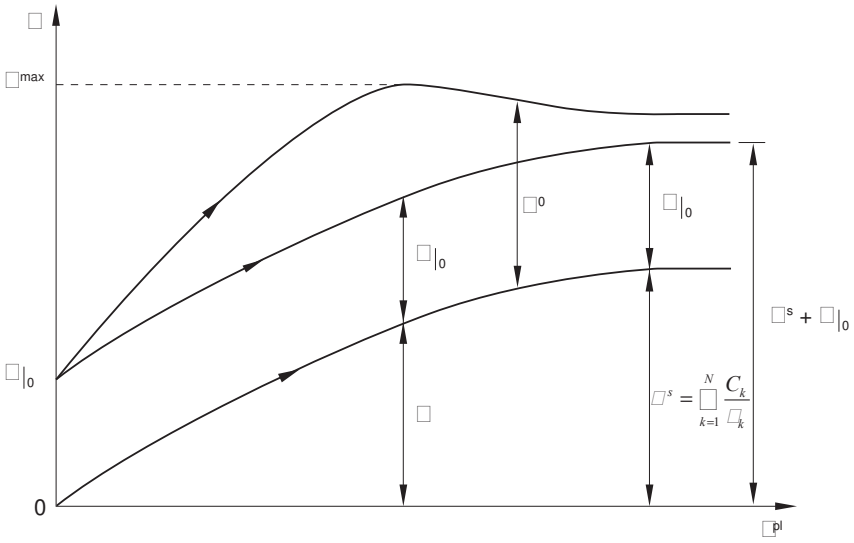


Figure 2: One-dimensional representation of the hardening in the nonlinear isotropic/kinematic model.

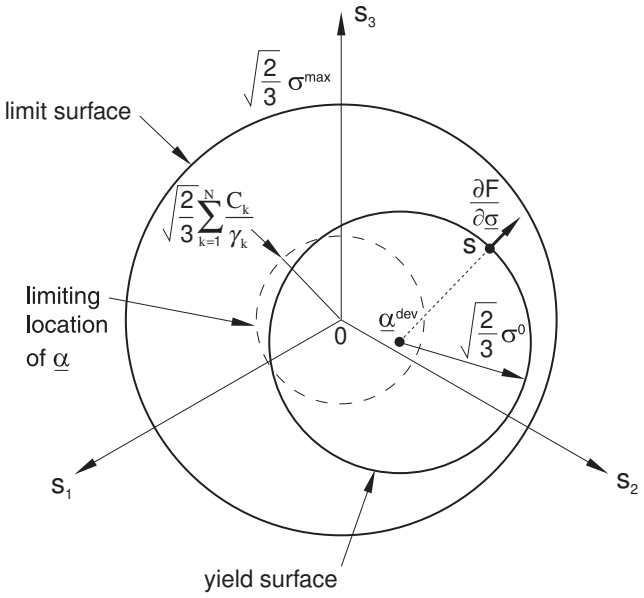


Figure 3: Three-dimensional representation of the hardening in the nonlinear isotropic/kinematic model.

Multilinear Kinematic Hardening Model

The multilinear kinematic hardening model is defined by giving the value of uniaxial yield stress as a function of uniaxial plastic strain (*Figure 4*). The number of stress-plastic strain pairs given on the input curve determines the number of subvolumes, N, in the model. The yield strength of the kth subvolume ($\sigma_{y,k}$) is calculated as

$$\sigma_{y,k} = \sigma_k + 3\mu\epsilon_k^{pl},$$

where $(\sigma_k, \epsilon_k^{pl})$ is the user-specified stress-plastic strain data and μ is the shear modulus of the material. All the subvolumes are subjected to the same total strain, and the total stress is calculated as a weighted sum of stresses in the different subvolumes:

$$\sigma = \sum_{k=1}^{N} w_k \sigma_k.$$

The weight of the kth subvolume is

$$w_k = \frac{3\mu}{3\mu + H_k} - \sum_{i=1}^{k-1} w_i,$$

where H_k is the hardening modulus between the kth data point and the $(k+1)$th data point on the stress versus plastic strain curve. The sum of the weights over all the subvolumes is one. It is assumed that the hardening modulus beyond the last data point is zero.

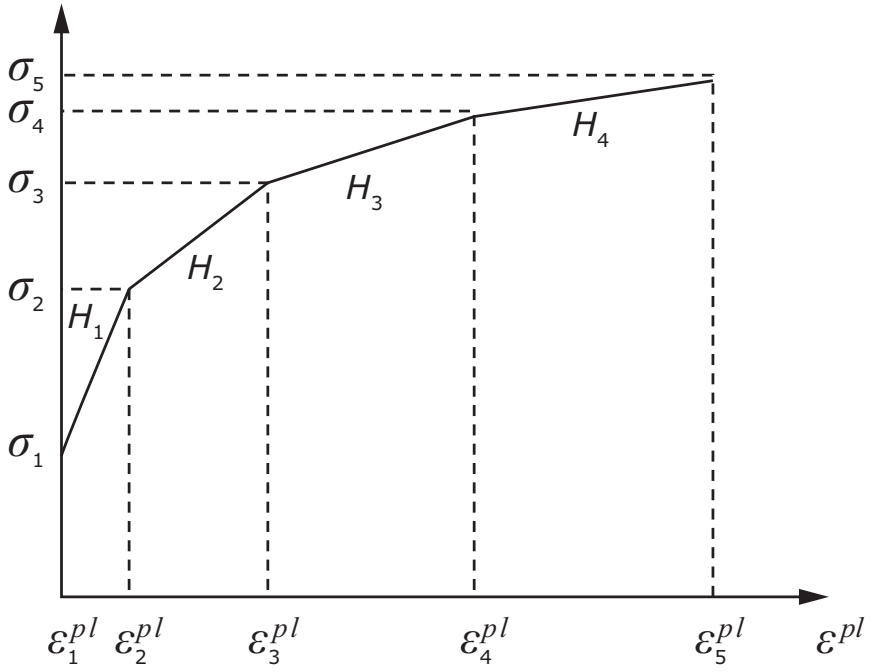


Figure 4: Uniaxial stress versus plastic strain curve in the case of the multilinear kinematic hardening model.

Predicted Material Behavior

In the kinematic hardening models the center of the yield surface moves in stress space due to the kinematic hardening component. In addition, when the nonlinear isotropic/kinematic hardening model is used, the yield surface range may expand or contract due to the isotropic component. These features allow modeling of inelastic deformation in metals that are subjected to cycles of load or temperature, resulting in significant inelastic deformation and, possibly, low-cycle fatigue failure. These models account for the following phenomena:

Bauschinger effect

This effect is characterized by a reduced yield stress upon load reversal after plastic deformation has occurred during the initial loading. This phenomenon decreases with continued cycling. The linear kinematic hardening component takes this effect into consideration, but a nonlinear component improves the shape of the cycles. Further improvement of the shape of the cycle can be obtained by using a nonlinear model

with multiple backstresses. The multilinear kinematic hardening model can also be used to model this effect.

Cyclic hardening with plastic shakedown

This phenomenon is characteristic of symmetric stress- or strain-controlled experiments. Soft or annealed metals tend to harden toward a stable limit, and initially hardened metals tend to soften. *Figure 5* illustrates the behavior of a metal that hardens under prescribed symmetric strain cycles.

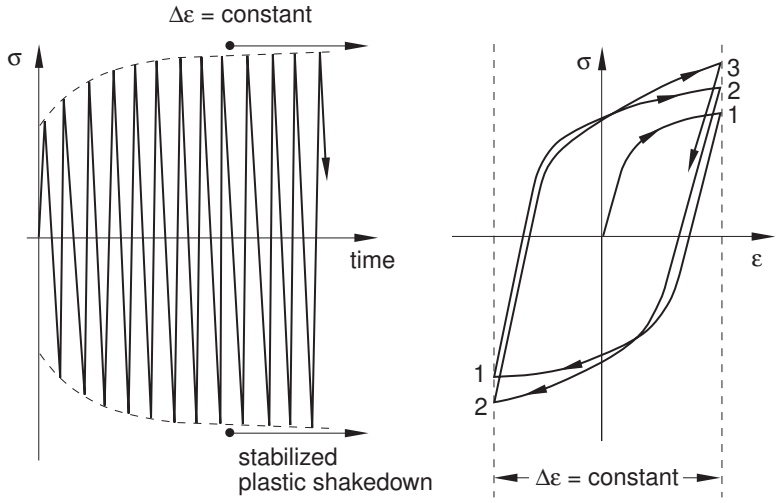


Figure 5: Plastic shakedown.

The kinematic hardening component of the models used alone predicts plastic shakedown after one stress cycle. The combination of the isotropic component together with the nonlinear kinematic component predicts shakedown after several cycles.

Ratcheting

Unsymmetric cycles of stress between prescribed limits will cause progressive “creep” or “ratcheting” in the direction of the mean stress (*Figure 6*).

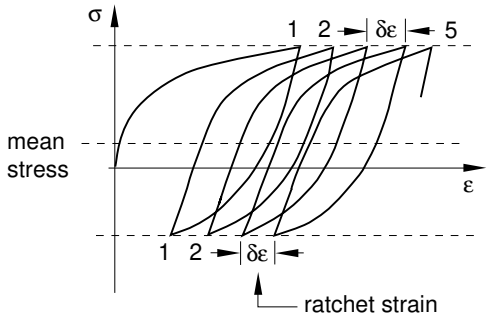


Figure 6: Ratcheting.

Typically, transient ratcheting is followed by stabilization (zero ratchet strain) for low mean stresses, while a constant increase in the accumulated ratchet strain is observed at high mean stresses. The nonlinear kinematic hardening component, used without the isotropic hardening component, predicts constant ratchet strain. The prediction of ratcheting is improved by adding isotropic hardening, in which case the ratchet strain may decrease until it becomes constant. However, in general the nonlinear hardening model with a single backstress predicts a too significant ratcheting effect. A considerable improvement in modeling

ratcheting can be achieved by superposing several kinematic hardening models (backstresses) and choosing one of the models to be linear or nearly linear ($\gamma_k \ll C_k$), which results in a less pronounced ratcheting effect. The multilinear kinematic hardening model cannot predict this effect.

Relaxation of the mean stress

This phenomenon is characteristic of an unsymmetric strain experiment, as shown in *Figure 7*.

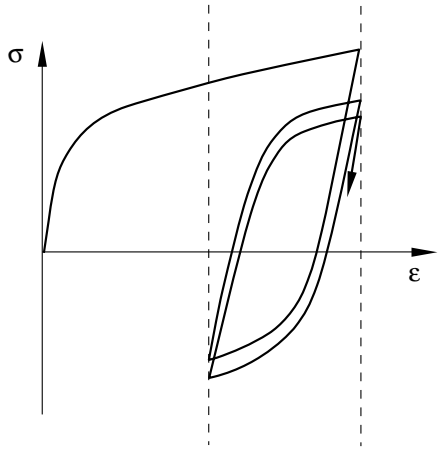


Figure 7: Relaxation of the mean stress.

As the number of cycles increases, the mean stress tends to zero. The nonlinear kinematic hardening component of the nonlinear isotropic/kinematic hardening model accounts for this behavior.

Limitations

The linear kinematic model is a simple model that gives only a first approximation of the behavior of metals subjected to cyclic loading, as explained above. The nonlinear isotropic/kinematic hardening model can provide more accurate results in many cases involving cyclic loading, but it still has the following limitations:

- The isotropic hardening is the same at all strain ranges. Physical observations, however, indicate that the amount of isotropic hardening depends on the magnitude of the strain range. Furthermore, if the specimen is cycled at two different strain ranges, one followed by the other, the deformation in the first cycle affects the isotropic hardening in the second cycle. Thus, the model is only a coarse approximation of actual cyclic behavior. It should be calibrated to the expected size of the strain cycles of importance in the application.
- The same cyclic hardening behavior is predicted for proportional and nonproportional load cycles. Physical observations indicate that the cyclic hardening behavior of materials subjected to nonproportional loading may be very different from uniaxial behavior at a similar strain amplitude.
- If the static recovery term is included in the model, Abaqus uses an explicit scheme to integrate this term. Therefore, in this case, a small time increment might be required to obtain accurate results.

The example problems *Simple proportional and nonproportional cyclic tests*, *Notched beam under cyclic loading*, and *Uniaxial ratcheting under tension and compression* illustrate the phenomena of cyclic hardening with plastic shakedown, ratcheting, and relaxation of the mean stress for the nonlinear isotropic/kinematic hardening model, as well as its limitations.

Usage and Calibration of the Kinematic Hardening Models

The linear kinematic model approximates the hardening behavior with a constant rate of hardening. This hardening rate should be matched to the average hardening rate measured in stabilized cycles over a strain range

corresponding to that expected in the application. A stabilized cycle is obtained by cycling over a fixed strain range until a steady-state condition is reached; that is, until the stress-strain curve no longer changes shape from one cycle to the next. The multilinear kinematic hardening model does not have the restriction of a constant hardening rate. The more general nonlinear model will give better predictions but requires more detailed calibration.

Linear Kinematic Hardening Model

The test data obtained from a half cycle of a unidirectional tension or compression experiment must be linearized, since this simple model can predict only linear hardening. The data are usually based on measurements of the stabilized behavior in strain cycles covering a strain range corresponding to the strain range that is anticipated to occur in the application. Abaqus expects you to provide only two data pairs to define this linear behavior: the yield stress, $\sigma|_0$, at zero plastic strain and a yield stress, σ, at a finite plastic strain value, ε^{pl}. The linear kinematic hardening modulus, C, is determined from the relation

$$C = \frac{\sigma - \sigma|_0}{\varepsilon^{pl}}.$$

You can provide several sets of two data pairs as a function of temperature to define the variation of the linear kinematic hardening modulus with respect to temperature. If the Hill yield surface is desired for this model, you must specify a set of yield ratios, R_{ij}, independently (see *Hill Anisotropic Yield/Creep* for information on how to specify the yield ratios).

This model gives physically reasonable results for only relatively small strains (less than 5%).

Input File Usage: **PLASTIC*, HARDENING=KINEMATIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Kinematic**

Nonlinear Isotropic/Kinematic Hardening Model

The evolution of the equivalent stress defining the size of the yield surface, σ^0, as a function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, defines the isotropic hardening component of the model. You can define this isotropic hardening component through an exponential law or directly in tabular form. It need not be defined if the yield surface remains fixed throughout the loading. In Abaqus/Explicit if the Hill yield surface is desired for this model, you must specify a set of yield ratios, R_{ij}, independently (see *Hill Anisotropic Yield/Creep* for information on how to specify the yield ratios). The Hill yield surface cannot be used with this model in Abaqus/Standard.

The material parameters C_k and γ_k determine the kinematic hardening component of the model. Abaqus offers three different ways of providing data for the kinematic hardening component of the model: the parameters C_k and γ_k can be specified directly, half-cycle test data can be given, or test data obtained from a stabilized cycle can be given. The experiments required to calibrate the model are described below.

Optionally, you can include static recovery by specifying material parameters ξ_k, $\overline{R}_k$, and m_k. If you specify these parameters, all of the kinematic hardening parameters must be specified directly. Test data input is not supported.

Defining the Isotropic Hardening Component by the Exponential Law

Specify the material parameters of the exponential law $\sigma|_0$, Q_∞, and b directly if they are already calibrated from test data. These parameters can be specified as functions of temperature and/or field variables.

Input File Usage: **CYCLIC HARDENING*, PARAMETERS

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Cyclic Hardening**: toggle on **Use parameters**.

Defining the Isotropic Hardening Component by Tabular Data

Isotropic hardening can be introduced by specifying the equivalent stress defining the size of the yield surface, σ^0, as a tabular function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$. The simplest way to obtain these data is to conduct a symmetric strain-controlled cyclic experiment with strain range $\Delta\varepsilon$. Since the material's elastic modulus is large compared to its hardening modulus, this experiment can be interpreted approximately as repeated cycles over the same plastic strain range $\Delta\varepsilon^{pl} \approx \Delta\varepsilon - 2\sigma_1^t/E$ (using the notation of *Figure 8*, where E is the Young's modulus of the material).

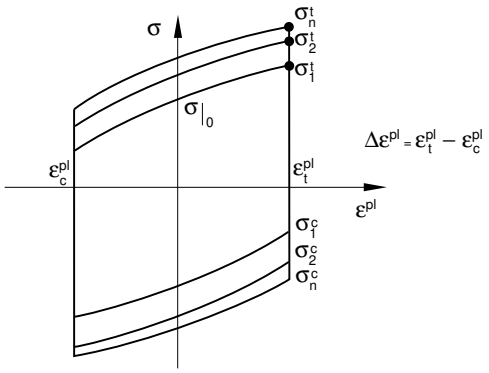


Figure 8: Symmetric strain cycle experiment.

The equivalent stress defining the size of the yield surface is $\sigma|_0$ at zero equivalent plastic strain; for the peak tensile stress points it is obtained by isolating the kinematic component from the yield stress (see *Figure 2*) as

$$\sigma_i^0 = \sigma_i^t - \alpha_i$$

for each cycle i, where $\alpha_i = \left(\sigma_i^t + \sigma_i^c\right)/2$. Since the model predicts approximately the same backstress value in each cycle at a particular strain level, $\alpha_i \approx \left(\sigma_1^t + \sigma_1^c\right)/2$. The equivalent plastic strain corresponding to σ_i^0 is

$$\bar{\varepsilon}_i^{pl} = \frac{1}{2}\left(4i - 3\right)\Delta\varepsilon^{pl}.$$

Data pairs (σ_i^0, $\bar{\varepsilon}_i^{pl}$), including the value $\sigma|_0$ at zero equivalent plastic strain, are specified in tabulated form. You should provide the tabulated values defining the size of the yield surface for the entire equivalent plastic strain range to which the material may be subjected. You can provide the data as functions of temperature and/or field variables. The yield stress at a given state is interpolated from this table of data, and it is assumed to remain constant outside the range of the independent variables other than equivalent plastic strain. Outside the range of equivalent plastic strains, you can choose if the yield stress remains constant (default) or is extrapolated linearly (see *Extrapolation of Material Data*).

To obtain accurate cyclic hardening data, such as would be needed for low-cycle fatigue calculations, the calibration experiment should be performed at a strain range, $\Delta\varepsilon$, that corresponds to the strain range anticipated in the analysis because the material model does not predict different isotropic hardening behavior at different strain ranges. This limitation also implies that, even though a component is made from the same material, it may

have to be divided into several regions with different hardening properties corresponding to different anticipated strain ranges. Field variables and field variable dependence of these properties can also be used for this purpose.

Abaqus allows the specification of strain rate effects in the isotropic component of the nonlinear isotropic/kinematic hardening model. The rate-dependent isotropic hardening data can be defined by specifying the equivalent stress defining the size of the yield surface, σ^0, as a tabular function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, at different values of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.

Input File Usage: Use the following option to define isotropic hardening with tabular data:

**CYCLIC HARDENING*

Use the following option to define rate-dependent isotropic hardening with tabular data:

**CYCLIC HARDENING*, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Use the following option to request that the yield stress remains constant outside the specified range of the equivalent plastic strains:

**CYCLIC HARDENING*, EXTRAPOLATION=CONSTANT (default)

Use the following option to request linear extrapolation of the yield stress outside the specified range of the equivalent plastic strains:

**CYCLIC HARDENING*, EXTRAPOLATION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Combined**: **Suboptions->Cyclic Hardening**

Requesting linear extrapolation outside the specified range of the equivalent plastic strains is not supported in Abaqus/CAE.

Defining the Isotropic Hardening Component in a User Subroutine in Abaqus/Standard

Specify σ^0 directly in user subroutine *UHARD*. σ^0 may be dependent on equivalent plastic strain and temperature. This method cannot be used if the kinematic hardening component is specified by using half-cycle test data.

Input File Usage: **CYCLIC HARDENING*, USER

Abaqus/CAE Usage: You cannot define the isotropic hardening component in user subroutine *UHARD* in Abaqus/CAE.

Defining the Kinematic Hardening Component by Specifying the Material Parameters Directly

The parameters C_k and γ_k can be specified directly as a function of temperature and/or field variables if they are already calibrated from test data. When γ_k depend on temperature and/or field variables, the response of the model under thermomechanical loading will generally depend on the *history* of temperature and/or field variables experienced at a material point. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, if this effect is not desired, constant values for γ_k should be specified to make the material response completely independent of the history of temperature and field variables. The algorithm

currently used to integrate the nonlinear isotropic/kinematic hardening model provides accurate solutions if the values of γ_k change moderately in an increment due to temperature and/or field variable dependence; however, this algorithm may not yield a solution with sufficient accuracy if the values of γ_k change abruptly in an increment.

The last term in the hardening law described in *Nonlinear Isotropic/Kinematic Hardening Model* above that is defined by the parameters ξ_k, $\overline{R}_k$, and m_k is called a static recovery term. This term allows you to capture the stress relaxation when modeling material deformation at elevated temperatures. Therefore, you should include this term if a model is subjected to loading at elevated temperatures that includes strain holds, such as commonly performed cycling loading tests with hold periods at tension and/or compression. If the static recovery term is defined, it is active only in quasi-static, fully coupled temperature-displacement, fully coupled thermal-electrical-structural, and implicit dynamic analyses.

Input File Usage: Use the following option to define hardening parameters if the static recovery term is not included:

**PLASTIC*, HARDENING=COMBINED, DATA TYPE=PARAMETERS, NUMBER BACKSTRESSES=*n*

Use the following option to define hardening parameters if the static recovery term is included:

**PLASTIC*, HARDENING=COMBINED, DATA TYPE=PARAMETERS, NUMBER BACKSTRESSES=*n*, STATIC RECOVERY

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Combined**, **Data type: Parameters**, **Number of backstresses**: *n*

Including the static recovery term is not supported in Abaqus/CAE.

Defining the Kinematic Hardening Component by Specifying Half-Cycle Test Data

If limited test data are available, C_k and γ_k can be based on the stress-strain data obtained from the first half cycle of a unidirectional tension or compression experiment. An example of such test data is shown in *Figure 9*. This approach is usually adequate when the simulation will involve only a few cycles of loading.

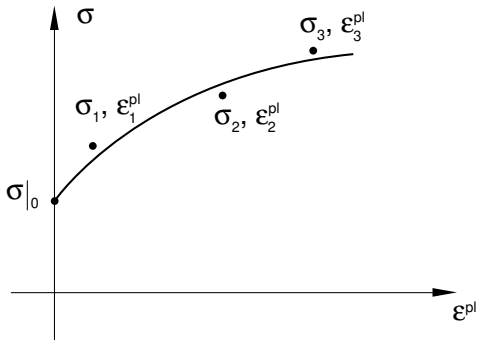


Figure 9: Half cycle of stress-strain data.

For each data point $(\sigma_i, \varepsilon_i^{pl})$ a value of α_i (α_i is the overall backstress obtained by summing all the backstresses at this data point) is obtained from the test data as

$$\alpha_i = \sigma_i - \sigma_i^0,$$

where σ_i^0 is the user-defined size of the yield surface at the corresponding plastic strain for the isotropic hardening component or the initial yield stress if the isotropic hardening component is not defined.

Integration of the backstress evolution laws over a half cycle yields the expressions

$$\alpha_k = \frac{C_k}{\gamma_k}\left(1 - e^{-\gamma_k \varepsilon^{pl}}\right),$$

which are used for calibrating C_k and γ_k.

When test data are given as functions of temperature and/or field variables, Abaqus determines several sets of material parameters (C_1, γ_1,..., C_N, γ_N), each corresponding to a given combination of temperature and/or field variables. Generally, this results in temperature-history (and/or field variable-history) dependent material behavior because the values of γ_k vary with changes in temperature and/or field variables. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, you can make the response of the material completely independent of the history of temperature and field variables by using constant values for the parameters γ_k. This can be achieved by running a data check analysis first; an appropriate constant values of γ_k can be determined from the information provided in the data file during the data check. The values for the parameters C_k and the constant parameters γ_k can then be entered directly as described above.

If the model with multiple backstresses is used, Abaqus obtains hardening parameters for different values of initial guesses and chooses the ones that give the best correlation with the experimental data provided. However, you should carefully examine the obtained parameters. In some cases it might be advantageous to obtain hardening parameters for different numbers of backstresses before choosing the set of parameters.

Static recovery is not available when you specify half-cycle test data.

Input File Usage: **PLASTIC*, HARDENING=COMBINED, DATA TYPE=HALF CYCLE, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Combined**, **Data type: Half Cycle**, **Number of backstresses**: *n*

Defining the Kinematic Hardening Component by Specifying Test Data from a Stabilized Cycle

Stress-strain data can be obtained from the stabilized cycle of a specimen that is subjected to symmetric strain cycles. A stabilized cycle is obtained by cycling the specimen over a fixed strain range $\Delta\varepsilon$ until a steady-state condition is reached; that is, until the stress-strain curve no longer changes shape from one cycle to the next.

Such a stabilized cycle is shown in *Figure 10*. Each data pair (σ_i, ε_i^{pl}) must be specified with the strain axis shifted to ε_p^0, so that

$$\varepsilon_i^{pl} = \varepsilon_i - \frac{\sigma_i}{E} - \varepsilon_p^0,$$

and, thus, $\varepsilon_1^{pl} = 0$. In the above equation, ε_i denotes the total strain for data point i, and $\varepsilon_p^0 = \varepsilon_1 - \sigma_1^0/E$.

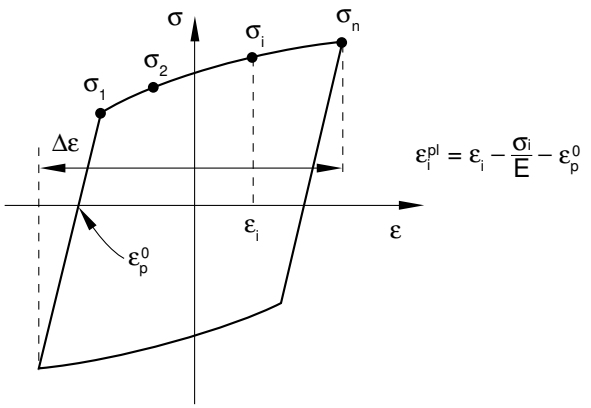


Figure 10: Stress-strain data for a stabilized cycle.

For each pair $(\sigma_i, \varepsilon_i^{pl})$ values of α_i (α_i is the overall backstress obtained by summing all the backstresses at this data point) are obtained from the test data as

$$\alpha_i = \sigma_i - \sigma^s,$$

where $\sigma^s = (\sigma_1 + \sigma_n)/2$ is the stabilized size of the yield surface.

Integration of the backstress evolution laws over this uniaxial strain cycle, with an exact match for the first data pair $(\sigma_1, 0)$, provides the expressions

$$\alpha_k = \frac{C_k}{\gamma_k}\left(1 - e^{-\gamma_k \varepsilon^{pl}}\right) + \alpha_{k,1} e^{-\gamma_k \varepsilon^{pl}},$$

where $\alpha_{k,1}$ denotes the k^{th} backstress at the first data point (initial value of the k^{th} backstress). The above equations enable calibration of the parameters C_k and γ_k.

If the shapes of the stress-strain curves are significantly different for different strain ranges, you may want to obtain several calibrated values of C_k and γ_k. The tabular data of the stress-strain curves obtained at different strain ranges can be entered directly in Abaqus. Calibrated values corresponding to each strain range are reported in the data file, together with an averaged set of parameters, if model definition data are requested (see *Controlling the Amount of analysis input file processor Information Written to the Data File*). Abaqus will use the averaged set in the analysis. These parameters may have to be adjusted to improve the match to the test data at the strain range anticipated in the analysis.

When test data are given as functions of temperature and/or field variables, Abaqus determines several sets of material parameters (C_1, γ_1,..., C_N, γ_N), each corresponding to a given combination of temperature and/or field variables. Generally, this results in temperature-history (and/or field variable-history) dependent material behavior because the values of γ_k vary with changes in temperature and/or field variables. This dependency on temperature-history is small and fades away with increasing plastic deformation. However, you can make the response of the material completely independent of the history of temperature and field variables by using constant values for the parameters γ_k. This can be achieved by running a data check analysis first; an appropriate constant values of γ_k can be determined from the information provided in the data file during the data check. The values for the parameters C_k and the constant parameters γ_k can then be entered directly as described above.

If the model with multiple backstresses is used, Abaqus obtains hardening parameters for different values of initial guesses and chooses the ones that give the best correlation with the experimental data provided. However, you should carefully examine the obtained parameters. In some cases it might be advantageous to obtain hardening parameters for different numbers of backstresses before choosing the set of parameters.

The isotropic hardening component should be defined by specifying the equivalent stress defining the size of the yield surface at zero plastic strain, as well as the evolution of the equivalent stress as a function of equivalent plastic strain. If this component is not defined, Abaqus will assume that no cyclic hardening occurs so that the equivalent stress defining the size of the yield surface is constant and equal to $(\sigma_1 + \sigma_n)/2$ (or the average of these quantities over several strain ranges when more than one strain range is provided). Since this size corresponds to the size of a saturated cycle, this is unlikely to provide accurate predictions of actual behavior, particularly in the initial cycles.

Static recovery is not available if you specify test data from a stabilized cycle.

Input File Usage: **PLASTIC*, HARDENING=COMBINED, DATA TYPE=STABILIZED, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Plastic**: **Hardening: Combined**, **Data type: Stabilized**, **Number of backstresses**: *n*

Multilinear Kinematic Hardening Model

The multilinear kinematic hardening model can be calibrated using the test data obtained from a unidirectional tension or compression experiment. Abaqus expects you to provide the stress-plastic strain data pairs to define this multilinear behavior. The hardening modulus between the *k*th data point and the $(k+1)$th data point on the stress versus plastic strain curve, H_k, is determined from the relation

$$H_k = \frac{\sigma_{k+1} - \sigma_k}{\varepsilon^{pl}_{k+1} - \varepsilon^{pl}_k}.$$

The data should be such that the hardening modulus does not increase with plastic strain. The number of data points provided by the user determines the number of subvolumes, N. The computational cost and memory requirements increase with N. Therefore, the number of data points used to approximate the stress-plastic strain curve with a piecewise linear response should always be minimized.

You can provide several sets of data pairs as a function of temperature and/or field variable. However, you must ensure that the total number of data points or subvolumes corresponding to different temperatures and/or field variables is the same.

Input File Usage: **PLASTIC*, HARDENING=MULTILINEAR KINEMATIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Plastic**: **Hardening: Multilinear-Kinematic**

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some hardening, Abaqus allows you to prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, and for the backstresses, $\boldsymbol{\alpha}_k$, in the case of the linear kinematic hardening model and the nonlinear isotropic/kinematic hardening model. When the nonlinear isotropic/kinematic hardening model is used, the initial conditions for each backstress, $\boldsymbol{\alpha}_k$, must satisfy the condition

$$\sqrt{\frac{3}{2}\boldsymbol{\alpha}_k^{dev} : \boldsymbol{\alpha}_k^{dev}} \leq C_k/\gamma_k$$

for the model to produce a kinematic hardening response. Abaqus allows the specification of initial backstresses that violate these conditions. However, in this case the response corresponding to the backstress for which the condition is violated produces kinematic softening response: the magnitude of the backstress decreases with plastic straining from its initial value to the saturation value. If the condition is violated for any of the backstresses, the overall response of the material is not guaranteed to produce kinematic hardening response. The initial condition for the backstress has no limitations when the linear kinematic hardening model is used.

You can specify the initial values of $\bar{\varepsilon}^{pl}$ and α_k directly as initial conditions (see *Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Number of backstresses**: *n*

User Subroutine Specification in Abaqus/Standard

For more complicated cases in Abaqus/Standard initial conditions can be defined through user subroutine *HARDINI*.

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING, USER, NUMBER BACKSTRESSES=*n*

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**, **Number of backstresses**: *n*

Elements

The linear kinematic hardening model and the nonlinear isotropic/kinematic hardening model can be used with elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom), except some beam elements in space. Beam elements in space that include shear stress caused by torsion (i.e., not thin-walled, open sections) and do not include hoop stress (i.e., not PIPE elements) cannot be used with the nonlinear kinematic hardening model. In Abaqus/Explicit, the linear kinematic hardening model and the nonlinear isotropic/kinematic hardening model can be used with any elements that include mechanical behavior, with the exception of one-dimensional elements (beams, pipes, and trusses) when the models are used with the Hill yield surface. The multilinear kinematic hardening model can be used only in Abaqus/Standard with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the kinematic hardening models:

ALPHA
Total kinematic hardening shift tensor components, α_{ij} $(i, j \leq 3)$.

ALPHA*k*
k^{th} kinematic hardening shift tensor components ($1 \leq k \leq 10$).

ALPHAN

All tensor components of all the kinematic hardening shift tensors, except the total shift tensor.

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \frac{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{pl}dt}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

PENER

Plastic work, defined as: $W^{pl} = \int_0^t \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl} dt$,. This quantity is not guaranteed to be monotonically increasing for kinematic hardening models. To get a quantity that is monotonically increasing, the plastic dissipation needs to be computed as:

$W^{pl} = \int_0^t (\boldsymbol{\sigma} - \boldsymbol{\alpha}) : \dot{\boldsymbol{\varepsilon}}^{pl} dt$. In Abaqus/Standard this quantity can be computed as a user-defined output variable in user subroutine *UVARM*.

YIELDS

Yield stress, σ^0.

Rate-Dependent Yield

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Classical Metal Plasticity*
- *Models for Metals Subjected to Cyclic Loading*
- *Johnson-Cook Plasticity*
- *Extended Drucker-Prager Models*
- *Crushable Foam Plasticity Models*
- *About the Material Library*
- *Inelastic Behavior*
- **RATE DEPENDENT*
- *Defining rate-dependent yield with yield stress ratios*

Overview

Rate-dependent yield:

- is needed to define a material's yield behavior accurately when the yield strength depends on the rate of straining and the anticipated strain rates are significant;
- is available only for the isotropic hardening metal plasticity models (Mises, Johnson-Cook, and Hill), the isotropic component of the nonlinear isotropic/kinematic plasticity models, the extended Drucker-Prager plasticity model, and the crushable foam plasticity model;
- can be conveniently defined on the basis of work hardening parameters and field variables by providing tabular data for the isotropic hardening metal plasticity models, the isotropic component of the nonlinear isotropic/kinematic plasticity models, and the extended Drucker-Prager plasticity model;
- can be defined through specification of user-defined overstress power law parameters, yield stress ratios, or Johnson-Cook rate dependence parameters (this last option is not available for the crushable foam plasticity model and is the only option available for the Johnson-Cook plasticity model);
- cannot be used with any of the Abaqus/Standard creep models (metal creep, time-dependent volumetric swelling, Drucker-Prager creep, or cap creep) since creep behavior is already a rate-dependent mechanism; and
- in dynamic analysis should be specified such that the yield stress increases with increasing strain rate.

Work Hardening Dependencies

Generally, a material's yield stress, $\overline{\sigma}$ (or $\overline{B}$ for the crushable foam model), is dependent on work hardening, which for isotropic hardening models is usually represented by a suitable measure of equivalent plastic strain, $\bar{\varepsilon}^{pl}$; the inelastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i:

$$\overline{\sigma} = \overline{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right).$$

Many materials show an increase in their yield strength as strain rates increase; this effect becomes important in many metals and polymers when the strain rates range between 0.1 and 1 per second, and it can be very

important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes.

Defining Hardening Dependencies for Various Material Models

Strain rate dependence can be defined by entering hardening curves at different strain rates directly or by defining yield stress ratios to specify the rate dependence independently.

Direct Entry of Test Data

Work hardening dependencies can be given quite generally as tabular data for the isotropic hardening Mises plasticity model, the isotropic component of the nonlinear isotropic/kinematic hardening model, and the extended Drucker-Prager plasticity model. The test data are entered as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates. The yield stress must be given as a function of the equivalent plastic strain and, if required, of temperature and of other predefined field variables. In defining this dependence at finite strains, "true" (Cauchy) stress and log strain values should be used. The hardening curve at each temperature must always start at zero plastic strain. For perfect plasticity only one yield stress, with zero plastic strain, should be defined at each temperature. It is possible to define the material to be strain softening as well as strain hardening. The work hardening data are repeated as often as needed to define stress-strain curves at different strain rates, starting with the static stress-strain curve and followed by curves corresponding to increasing values of strain rate. The yield stress at a given strain and strain rate is interpolated directly from these tables.

Input File Usage: Use one of the following options:

**PLASTIC*, HARDENING=ISOTROPIC, RATE=$\dot{\bar{\varepsilon}}^{pl}$

**CYCLIC HARDENING*, RATE=$\dot{\bar{\varepsilon}}^{pl}$

**DRUCKER PRAGER HARDENING*, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Use one of the following models:

Property module: material editor:
Mechanical->Plasticity->Plastic: **Hardening: Isotropic**,
Use strain-rate-dependent data
Mechanical->Plasticity->Drucker Prager: **Suboptions->Drucker Prager Hardening**: **Use strain-rate-dependent data**

Cyclic hardening is not supported in Abaqus/CAE.

Using Yield Stress Ratios

Alternatively, and as the only means of defining rate-dependent yield stress for the Johnson-Cook and the crushable foam plasticity models, the strain rate behavior can be assumed to be separable, so that the stress-strain dependence is similar at all strain rate levels:

$$\overline{\sigma} = \sigma^0\left(\bar{\varepsilon}^{pl}, \theta, f_i\right) R\left(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right),$$

where $\sigma^0\left(\bar{\varepsilon}^{pl}, \theta, f_i\right)$ (or $B\left(\bar{\varepsilon}^{pl}, \theta, f_i\right)$ in the foam model) is the static stress-strain behavior and $R\left(\dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)$ is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R\left(0, \theta, f_i\right) = 1.0$).

Three methods are offered to define *R* in Abaqus: specifying an overstress power law, defining *R* directly as a tabular function, or specifying an analytical Johnson-Cook form to define *R*.

Overstress Power Law

The Cowper-Symonds overstress power law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D(R-1)^n \quad \text{for} \quad \bar{\sigma} \geq \sigma^0 \quad \left(\text{or} \quad \overline{B} \geq B \quad \text{in the crushable foam model}\right),$$

where $D(\theta, f_i)$ and $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: **RATE DEPENDENT*, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Suboptions**->**Rate Dependent**: **Hardening: Power Law** *(available for valid plasticity models)*

Chaboche Rate Dependence

Chaboche rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left\langle \frac{\bar{\sigma}-\sigma^0}{K} \right\rangle^n,$$

where $\dot{\varepsilon}_0(\theta, f_i)$, $K(\theta, f_i)$, and $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

The above relation can be rewritten as

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left(\frac{\sigma^0}{K}\right)^n \left(R-1\right)^n.$$

If Chaboche rate dependence is used with the crushable foam model, $\bar{\sigma}$ and σ must be replaced with $\overline{B}$ and B, respectively, in the above relations. In the perfectly plastic case when σ^0 or B does not depend on plastic strain, this law and the overstress power law become identical if $D = \dot{\varepsilon}_0 \left(\frac{\sigma^0}{K}\right)^n$. However, in general, if hardening is defined these laws produce different results.

Input File Usage: **RATE DEPENDENT*, TYPE=CHABOCHE

Abaqus/CAE Usage: Chaboche rate dependence is not supported in Abaqus/CAE.

Tabular Function

Alternatively, R can be entered directly as a tabular function of the equivalent plastic strain rate (or the axial plastic strain rate in a uniaxial compression test for the crushable foam model), $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and field variables, f_i.

Input File Usage: **RATE DEPENDENT*, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Suboptions**->**Rate Dependent**: **Hardening: Yield Ratio** *(available for valid plasticity models)*

Johnson-Cook Rate Dependence

Johnson-Cook rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}(R-1)\right] \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where $\dot{\varepsilon}_0$ and C are material constants that do not depend on temperature and are assumed not to depend on predefined field variables. Johnson-Cook rate dependence can be used in conjunction with the Johnson-Cook plasticity model, the isotropic hardening metal plasticity models, and the extended Drucker-Prager plasticity model (it cannot be used in conjunction with the crushable foam plasticity model).

This is the only form of rate dependence available for the Johnson-Cook plasticity model. For more details, see *Johnson-Cook Plasticity*.

Input File Usage: **RATE DEPENDENT*, TYPE=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Suboptions**->**Rate Dependent**: **Hardening: Johnson-Cook** *(available for valid plasticity models)*

Elements

Rate-dependent yield can be used with all elements that include mechanical behavior (elements that have displacement degrees of freedom).

Rate-Dependent Plasticity: Creep and Swelling

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*
- **CREEP*
- **CREEP STRAIN RATE CONTROL*
- **POTENTIAL*
- **SWELLING*
- **RATIOS*
- *Defining a creep law*
- *Defining swelling*

Overview

The classical deviatoric metal creep behavior in Abaqus/Standard:

- can be defined using user subroutine *CREEP* or by providing parameters as input for some simple creep laws;
- can model either isotropic creep (using Mises stress potential) or anisotropic creep (using Hill's anisotropic stress potential);
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure;
- requires that the material's elasticity be defined as linear elastic behavior;
- can be modified to implement the auxiliary creep hardening rules specified in Nuclear Standard NEF 9-5T, “Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components”; these rules are exercised by means of a constitutive model developed by Oak Ridge National Laboratory (*ORNL – Oak Ridge National Laboratory Constitutive Model*);
- can be used in combination with creep strain rate control in analyses in which the creep strain rate must be kept within a certain range; and
- can potentially result in errors in calculated creep strains if anisotropic creep and plasticity occur simultaneously (discussed below).

Rate-dependent gasket behavior in Abaqus/Standard:

- uses unidirectional creep as part of the model of the gasket's thickness-direction behavior;
- can be defined using user subroutine *CREEP* or by providing parameters as input for some simple creep laws;
- is active only during steps using the quasi-static procedure; and
- requires that an elastic-plastic model be used to define the rate-independent part of the thickness-direction behavior of the gasket.

Volumetric swelling behavior in Abaqus/Standard:

- can be defined using user subroutine *CREEP* or by providing tabular input;

- can be either isotropic or anisotropic;
- is active only during steps using the coupled temperature-displacement procedure, the transient soils consolidation procedure, and the quasi-static procedure; and
- requires that the material's elasticity be defined as linear elastic behavior.

Creep Behavior

Creep behavior is specified by the equivalent uniaxial behavior—the creep "law." In practical cases, creep laws are typically of very complex form to fit experimental data; therefore, the laws are defined with user subroutine *CREEP*, as discussed below. Alternatively, five common creep laws are provided in Abaqus/Standard: the power law, the hyperbolic-sine law, the double power law, the Anand law, and the Darveaux law. These standard creep laws are used for modeling secondary or steady-state creep. Creep is defined by including creep behavior in the material model definition (*Material Data Definition*). Alternatively, creep can be defined in conjunction with gasket behavior to define the rate-dependent behavior of a gasket.

Input File Usage: Use the following options to include creep behavior in the material model definition:

**MATERIAL*
**CREEP*

Use the following options to define creep in conjunction with gasket behavior:

**GASKET BEHAVIOR*
**CREEP*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**

Choosing a Creep Model

The power law creep model is attractive for its simplicity. However, it is limited in its range of application. The time-hardening version of the power law creep model is typically recommended only in cases when the stress state remains essentially constant. The strain-hardening version of power law creep should be used when the stress state varies during an analysis. In the case where the stress is constant and there are no temperature or field dependencies, the time-hardening and strain-hardening versions of the power creep law are equivalent. For either version of the power law, the stresses should be relatively low.

In regions of high stress, such as around a crack tip, the creep strain rates frequently show an exponential dependence of stress. The hyperbolic-sine creep law shows exponential dependence on the stress, σ, at high stress levels ($\sigma/\sigma^0 \gg 1$, where σ^0 is the yield stress) and reduces to the power law at low stress levels (with no explicit time dependence).

The double power, Anand, and Darveaux models are particularly well suited for modeling the behavior of solder alloys used in electronic packaging and have been shown to produce accurate results for a wide range of temperatures and strain rates.

None of the above models is suitable for modeling creep under cyclic loading. The ORNL model (*ORNL – Oak Ridge National Laboratory Constitutive Model*) is an empirical model for stainless steel that gives approximate results for cyclic loading without having to perform the cyclic loading numerically. Generally, creep models for cyclic loading are complicated and must be added to a model with user subroutine *CREEP* or with user subroutine *UMAT*.

Modeling Simultaneous Creep and Plasticity

If creep and plasticity occur simultaneously and implicit creep integration is in effect, both behaviors might interact and a coupled system of constitutive equations needs to be solved. If creep and plasticity are isotropic, Abaqus/Standard properly takes into account such coupled behavior, even if the elasticity is anisotropic. However, if either creep or plasticity are anisotropic, Abaqus/Standard integrates the creep equations without taking plasticity into account, which might lead to substantial errors in the creep strains. This situation develops only if plasticity and creep are active at the same time, such as would occur during a long-term load increase; you would not expect to have a problem if there is a short-term preloading phase in which plasticity dominates, followed by a creeping phase in which no further yielding occurs. Integration of the creep laws and rate-dependent plasticity are discussed in *Rate-dependent metal plasticity (creep)*.

Time Power Law and Power Law Models

The time power law and power law models described below are equivalent to the "time hardening" and the "strain hardening" forms but avoid their drawbacks. The time power law and power law models rewrite the laws in such a way that the typical parameter values do not cause numerical difficulties. In addition, the units of all the parameters are physical, which makes unit conversion easier if it is required.

Time Power Law Model

The time power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left(\frac{\tilde{q}}{q_0} \right)^n \left(\dot{\varepsilon}_0 t \right)^m,$$

where $\tilde{q}$ is Mises equivalent stress or Hill's anisotropic equivalent deviatoric stress according to whether isotropic or anisotropic creep behavior is defined and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

The model is equivalent to the "time hardening" form. It is recommended that you use the time power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent time power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: **CREEP*, LAW=TIME POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Time Power**

Power Law Model

The power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left[\left(\frac{\tilde{q}}{q_0} \right)^n \left[(m+1)\, \bar{\varepsilon}^{cr} \right]^m \right]^{\frac{1}{m+1}},$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the uniaxial equivalent creep strain rate, $\sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}$,

$\bar{\varepsilon}^{cr}$

is the equivalent creep strain,

$\tilde{q}$

is the uniaxial equivalent deviatoric stress, and

$\dot{\varepsilon}_0, n, m,$ and q_0

are material parameters.

This model is equivalent to the "strain hardening" form. It is recommended that you use the power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: **CREEP*, LAW=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Power**

Time/Strain Hardening Models

Time hardening and strain hardening models to specify creep are available. However, to avoid the drawbacks of these models, it is recommended that you use the time power law and power law models (see *Time Power Law and Power Law Models*).

Time Hardening Form

The "time hardening" form is the simpler of the two forms and is defined as

$$\dot{\bar{\varepsilon}}^{cr} = A\tilde{q}^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the uniaxial equivalent creep strain rate, $\sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}$,

$\tilde{q}$

is the uniaxial equivalent deviatoric stress,

t

is the total or the creep time, and

A*, *n*, and *m

are defined by you as functions of temperature.

For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: **CREEP*, LAW=TIME

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Time-Hardening**

Strain Hardening Form

The "strain hardening" form is

$$\dot{\bar{\varepsilon}}^{cr} = \left(A\tilde{q}^n[(m+1)\,\bar{\varepsilon}^{cr}]^m\right)^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$ and $\bar{\varepsilon}^{cr}$ are defined above.

Input File Usage: **CREEP*, LAW=STRAIN

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Strain-Hardening**

Numerical Difficulties

Depending on the choice of units for either form, the value of A might be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations. Therefore, use another system of units or use the time power law or power law model (described below) to avoid such difficulties in the calculation of creep strain increments.

Time-Dependent Behavior

In the "time hardening" form and the time power law model, the total time or the creep time can be used. The total time is the accumulated time over all general analysis steps. The creep time is the sum of the times of the procedures with time-dependent material behavior. If the total time is used, it is recommended that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage: Use one of the following options:

**CREEP*, TIME=TOTAL (default)
**CREEP*, TIME=CREEP

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Time: Total** or **Creep**

Hyperbolic-Sine Law Model

The hyperbolic-sine law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A(\sinh B\tilde{q})^n \exp\left(-\frac{\triangle H}{R(\theta - \theta^Z)}\right),$$

where

$\dot{\bar{\varepsilon}}^{cr}$ and $\tilde{q}$

are defined above,

θ

is the temperature,

θ^Z

is the user-defined value of absolute zero on the temperature scale used,

$\triangle H$

is the activation energy,

R

is the universal gas constant, and

A, B, and n

are other material parameters.

This model includes temperature dependence, which is apparent in the above expression; however, the parameters A, B, n, $\triangle H$, and R cannot be defined as functions of temperature.

Input File Usage: Use both of the following options:

**CREEP*, LAW=HYPERB
**PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Define both of the following:

Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Hyperbolic-Sine**

Any module: **Model->Edit Attributes->*model_name***: **Absolute zero temperature**

Numerical Difficulties

As with the power law, A might be very small for typical creep strain rates. If A is very small (such as less than 10^{-27}), use another system of units to avoid numerical difficulties in the calculation of creep strain increments.

Anand Model

The Anand model is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A\left[\sinh\left(\xi\frac{\tilde{q}}{s}\right)\right]^{\frac{1}{m}} \exp\left(-\frac{Q}{R(\theta-\theta^Z)}\right),$$

where

$\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$, R, θ, and θ^Z

are defined above,

Q

is the activation energy,

s

is the deformation resistance, and

A, m, and ξ

are material parameters.

The evolution equation for the deformation resistance, s (initially $s = s_0$), is

$$\dot{s} = h_0\left|1-\frac{s}{s^*}\right|^a sign\left(1-\frac{s}{s^*}\right)\dot{\bar{\varepsilon}}^{cr},$$

with

$$s^* = \hat{s}\left[\frac{1}{A}\dot{\bar{\varepsilon}}^{cr}\exp\left(\frac{Q}{R(\theta-\theta^Z)}\right)\right]^n,$$

where

$$h_0 = A_0 + A_1\left(\theta-\theta^Z\right) + A_2\left(\theta-\theta^Z\right)^2 + A_3\dot{\bar{\varepsilon}}^{cr} + A_4\left(\dot{\bar{\varepsilon}}^{cr}\right)^2,$$

and a, n, $\hat{s}$, A_0, A_1, A_2, A_3, and A_4 are material parameters.

In addition, the initial deformation resistance is a function of temperature of the form

$$s_0 = S_1 + S_2\left(\theta-\theta^Z\right) + S_3\left(\theta-\theta^Z\right)^2,$$

where S_1, S_2, and S_3 are material parameters.

Input File Usage: Use both of the following options:

**CREEP*, LAW=ANAND
**PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Define both of the following:

Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Anand**

Any module: **Model->Edit Attributes->*model_name***: **Absolute zero temperature**

Darveaux Model

The Darveaux model involves both primary and secondary creep. The secondary creep (steady-state) component is described by a standard hyperbolic sine law

$$\dot{\bar{\varepsilon}}_s^{cr} = C_{ss}[\sinh(\alpha\tilde{q})]^n \exp\left(-\frac{Q}{R(\theta-\theta^Z)}\right).$$

The steady-state law is modified to include the primary creep effects through

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\bar{\varepsilon}}_s^{cr}\left[1+\epsilon_T B\exp\left(-B\dot{\bar{\varepsilon}}_s^{cr}t\right)\right],$$

where

$\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$, R, Q, θ, and θ^Z

are defined above,

C_{ss}

is the steady-state creep prefactor,

α

is the steady-state creep power law breakdown, and

n, ϵ_T, and B

are other material parameters.

Input File Usage: Use both of the following options:

**CREEP*, LAW=DARVEAUX
**PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Define both of the following:

Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Darveaux**

Any module: **Model->Edit Attributes->*model_name***: **Absolute zero temperature**

Double Power Model

The double power law is available in the form

$$\dot{\bar{\varepsilon}}^{cr} = A_1\exp\left(-\frac{B_1}{(\theta-\theta^Z)}\right)\left(\frac{\tilde{q}}{\sigma_0}\right)^{C_1} + A_2\exp\left(-\frac{B_2}{(\theta-\theta^Z)}\right)\left(\frac{\tilde{q}}{\sigma_0}\right)^{C_2},$$

where

$\dot{\bar{\varepsilon}}^{cr}$, $\tilde{q}$, θ, and θ^Z

are defined above,

σ_0

is the normalized stress, and

A_1, A_2, B_1, B_2, C_1, and C_2

are other material parameters.

Input File Usage: Use both of the following options:

**CREEP*, LAW=DOUBLE POWER
**PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Define both of the following:

Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Double Power**

Any module: **Model->Edit Attributes->*model_name***: **Absolute zero temperature**

Anisotropic Creep

Anisotropic creep can be defined to specify the stress ratios that appear in Hill's function. You must define the ratios R_{ij} in each direction that will be used to scale the stress value when the creep strain rate is calculated. The ratios can be defined as constant or dependent on temperature and other predefined field variables. The ratios are defined with respect to the user-defined local material directions or the default directions (see *Orientations*). Further details are provided in *Hill Anisotropic Yield/Creep*. Anisotropic creep is not available when creep is used to define a rate-dependent gasket behavior since only the gasket thickness-direction behavior can have rate-dependent behavior.

Input File Usage: **POTENTIAL*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Suboptions->Potential**

Volumetric Swelling Behavior

As with the creep laws, volumetric swelling laws are usually complex and are most conveniently specified in user subroutine *CREEP* as discussed below. However, a means of tabular input is also provided for the form

$$\dot{\tilde{\varepsilon}}^{sw} = f(\theta, f_1, f_2, \ldots),$$

where $\dot{\tilde{\varepsilon}}^{sw}$ is the volumetric strain rate caused by swelling and f_1, f_2, ... are predefined fields such as irradiation fluxes in cases involving nuclear radiation effects. Up to six predefined fields can be specified.

Volumetric swelling cannot be used to define a rate-dependent gasket behavior.

Input File Usage: **SWELLING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Swelling**

Anisotropic Swelling

Anisotropy can easily be included in the swelling behavior. If anisotropic swelling behavior is defined, the anisotropic swelling strain rate is expressed as

$$\dot{\bar{\varepsilon}}_A^{sw} = \dot{\bar{\varepsilon}}_{11}^{sw} + \dot{\bar{\varepsilon}}_{22}^{sw} + \dot{\bar{\varepsilon}}_{33}^{sw} = (r_{11} + r_{22} + r_{33}) \frac{1}{3} \dot{\bar{\varepsilon}}^{sw},$$

where $\dot{\bar{\varepsilon}}^{sw}$ is the volumetric swelling strain rate that you define either directly (discussed above) or in user subroutine *CREEP*. The ratios r_{11}, r_{22}, and r_{33} are also user-defined. The directions of the components of the swelling strain rate are defined by the local material directions, which can be either user-defined or the default directions (see *Orientations*).

Input File Usage: Use both of the following options:

**SWELLING*
**RATIOS*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Swelling**: **Suboptions->Ratios**

User Subroutine CREEP

User subroutine *CREEP* provides a very general capability for implementing viscoplastic models such as creep and swelling models in which the strain rate potential can be written as a function of equivalent pressure stress, p; the Mises or Hill's equivalent deviatoric stress, $\tilde{q}$; and any number of solution-dependent state variables. Solution-dependent state variables are used in conjunction with the constitutive definition; their values evolve with the solution and can be defined in this subroutine. Examples are hardening variables associated with the model.

The user subroutine can also be used to define very general rate- and time-dependent thickness-direction gasket behavior. When an even more general form is required for the strain rate potential, user subroutine *UMAT* (*User-Defined Mechanical Material Behavior*) can be used.

Input File Usage: Use one or both of the following options. Only the first option can be used to define gasket behavior.

**CREEP*, LAW=USER
**SWELLING*, LAW=USER

Abaqus/CAE Usage: Use one or both of the following models. Only the first model can be used to define gasket behavior.

Property module: material editor:
Mechanical->Plasticity->Creep: **Law: User defined**
Mechanical->Plasticity->Swelling: **Law: User subroutine CREEP**

Removing Creep Effects in an Analysis Step

You can specify that no creep (or viscoelastic) response can occur during certain analysis steps, even if creep (or viscoelastic) material properties have been defined.

Input File Usage: Use one of the following options:

**COUPLED TEMPERATURE-DISPLACEMENT*, CREEP=NONE
**SOILS*, CONSOLIDATION, CREEP=NONE

Abaqus/CAE Usage: Use one of the following options:

Step module: **Create Step**:
Coupled temp-displacement: toggle off **Include creep/swelling/viscoelastic behavior**

Soils: **Pore fluid response: Transient consolidation**: toggle off **Include creep/swelling/viscoelastic behavior**

Integration

Explicit integration, implicit integration, or both integration schemes can be used in a creep analysis, depending on the procedure used, the parameters specified for the procedure, the presence of plasticity, and whether or not geometric nonlinearity is requested.

Application of Explicit and Implicit Schemes

Nonlinear creep problems are often solved efficiently by forward-difference integration of the inelastic strains (the "initial strain" method). This explicit method is computationally efficient because, unlike implicit methods, iteration is not required. Although this method is only conditionally stable, the numerical stability limit of the explicit operator is usually sufficiently large to allow the solution to be developed in a small number of time increments.

Abaqus/Standard uses either an explicit or an implicit integration scheme or switches from explicit to implicit in the same step. These schemes are outlined first, followed by a description of which procedures use these integration schemes.

1. Integration scheme 1: Starts with explicit integration and switches to implicit integration based on either stability or if plasticity is active. The stability limit used in explicit integration is discussed in the next section.
2. Integration scheme 2: Starts with explicit integration and switches to implicit integration when plasticity is active. The stability criterion does not play a role here.
3. Integration scheme 3: Always uses implicit integration.

The use of the above integration schemes is determined by the procedure type, your choice of the integration type to be used, as well as whether or not geometric nonlinearity is requested. For quasi-static and coupled temperature-displacement procedures, if you do not choose an integration type, integration scheme 1 is used for a geometrically linear analysis and integration scheme 3 is used for a geometrically nonlinear analysis. You can force Abaqus/Standard to use explicit integration for creep and swelling effects in coupled temperature-displacement or quasi-static procedures, when plasticity is not active throughout the step (integration scheme 2). Explicit integration can be used regardless of whether or not geometric nonlinearity has been requested (see *General and Perturbation Procedures*).

For a transient soils consolidation procedure, the implicit integration scheme (integration scheme 3) is always used, irrespective of whether a geometrically linear or nonlinear analysis is performed.

Input File Usage: Use one of the following options to restrict Abaqus/Standard to using explicit integration:

*VISCO, CREEP=EXPLICIT
*COUPLED TEMPERATURE-DISPLACEMENT, CREEP=EXPLICIT

Abaqus/CAE Usage: Use one of the following options to restrict Abaqus/Standard to using explicit integration:

Step module: **Create Step**:
Visco: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Coupled temp-displacement: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: **Creep/swelling/viscoelastic integration: Explicit**

Automatic Monitoring of Stability Limit during Explicit Integration

Abaqus/Standard monitors the stability limit automatically during explicit integration. If, at any point in the model, the creep strain increment $\left(\dot{\bar{\varepsilon}}^{cr}\big|_t \Delta t\right)$ is larger than the total elastic strain, the problem will become unstable. Therefore, a stable time step, Δt_s, is calculated every increment by

$$\Delta t_s = 0.5 \frac{\bar{\varepsilon}^{el}\big|_t}{\dot{\bar{\varepsilon}}^{cr}\big|_t},$$

where $\bar{\varepsilon}^{el}\big|_t$ is the equivalent total elastic strain at time t, the beginning of the increment, and $\dot{\bar{\varepsilon}}^{cr}\big|_t$ is the equivalent creep strain rate at time t. Furthermore,

$$\bar{\varepsilon}^{el}\big|_t = \frac{\tilde{q}|_t}{\widetilde{E}},$$

where $\tilde{q}|_t$ is the Mises stress at time t, and

$$\widetilde{E} = 2(1+\nu)\left(\mathbf{n} : \mathbf{D}^{el} : \mathbf{n}\right) \approx 2.5\overline{E},$$

where

$\mathbf{n} = \partial \tilde{q}|_t / \partial \boldsymbol{\sigma}$

is the gradient of the deviatoric stress potential,

$\mathbf{D}^{el}$

is the elasticity matrix, and

$\overline{E}$

is an effective elastic modulus—for isotropic elasticity $\overline{E}$ can be approximated by Young's modulus.

At every increment for which explicit integration is performed, the stable time increment, Δt_s, is compared to the critical time increment, Δt_c, which is calculated as follows:

$$\Delta t_c = \frac{errtol}{\dot{\bar{\varepsilon}}^{cr}\big|_{t+\Delta t} - \dot{\bar{\varepsilon}}^{cr}\big|_t}.$$

The quantity *errtol* is an error tolerance that you define as discussed below. If Δt_s is less than Δt_c, Δt_s is used as the time increment, which would mean that the stability criterion was limiting the size of the time step further than required by accuracy considerations. Abaqus/Standard will automatically switch to the backward difference operator (the implicit method, which is unconditionally stable) if Δt_s is less than Δt_c for nine consecutive increments, you have not restricted Abaqus/Standard to explicit integration as discussed above, and there is sufficient time left in the analysis (time left $\geq 50\Delta t$). The stiffness matrix will be reformed at every iteration if the implicit algorithm is used.

Specifying the Tolerance for Automatic Incrementation

The integration tolerance must be chosen so that increments in stress, $\Delta\boldsymbol{\sigma}$, are calculated accurately. Consider a one-dimensional example. The stress increment, $\Delta\sigma$, is

$$\Delta\sigma = E\Delta\varepsilon^{el} = E\left(\Delta\varepsilon - \Delta\varepsilon^{cr}\right),$$

where $\Delta\varepsilon^{el}$, $\Delta\varepsilon$, and $\Delta\varepsilon^{cr}$ are the uniaxial elastic, total, and creep strain increments, respectively, and E is the elastic modulus. For $\Delta\sigma$ to be calculated accurately, the error in the creep strain increment, $\Delta\varepsilon^{cr}_{err}$, must be small compared to $\Delta\varepsilon^{el}$; that is,

$$\Delta\varepsilon^{cr}_{err} \ll \Delta\varepsilon^{el}.$$

Measuring the error in $\Delta\varepsilon^{cr}$ as

$$\Delta\varepsilon^{cr}_{err} = \left(\dot{\bar{\varepsilon}}^{cr}\big|_{t+\Delta t} - \dot{\bar{\varepsilon}}^{cr}\big|_t\right)\Delta t$$

leads to

$$\left(\dot{\bar{\varepsilon}}^{cr}\big|_{t+\Delta t} - \dot{\bar{\varepsilon}}^{cr}\big|_t\right)\Delta t \ll \Delta\varepsilon^{el} = \frac{\Delta\sigma}{E}, \text{ or}$$

$$errtol \ll \frac{\Delta\sigma}{E}.$$

You define *errtol* for the applicable procedure by choosing an acceptable stress error tolerance and dividing this by a typical elastic modulus; therefore, it should be a small fraction of the ratio of the typical stress and the effective elastic modulus in a problem. It is important to recognize that this approach for selecting a value for *errtol* is often very conservative, and acceptable solutions can usually be obtained with higher values.

Input File Usage: Use one of the following options:

**VISCO*, CETOL=*errtol*
**COUPLED TEMPERATURE-DISPLACEMENT*, CETOL=*errtol*
**SOILS*, CONSOLIDATION, CETOL=*errtol*

Abaqus/CAE Usage: Use one of the following options:

Step module: **Create Step**:
Visco: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Coupled temp-displacement: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Soils: **Pore fluid response: Transient consolidation**: toggle on **Include creep/swelling/viscoelastic behavior**: **Incrementation**: toggle on **Creep/swelling/viscoelastic strain error tolerance**, and enter a value

Loading Control Using Creep Strain Rate

In superplastic forming a controllable pressure is applied to deform a body. Superplastic materials can deform to very large strains, provided that the strain rates of the deformation are maintained within very tight tolerances. The objective of the superplastic analysis is to predict how the pressure must be controlled to form the component as fast as possible without exceeding a superplastic strain rate anywhere in the material.

To achieve this using Abaqus/Standard, the controlling algorithm is as follows. During an increment Abaqus/Standard calculates $r_{\max}$, the maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate for any integration point in a specified element set. If $r_{\max}$ is less than 0.2 or greater than 3.0 in a given increment, the increment is abandoned and restarted with the following load modifications:

$$r_{\max} < 0.2 \quad p = 2.0\, p_{\text{old}}, \text{ or}$$
$$r_{\max} > 3.0 \quad p = 0.5\, p_{\text{old}},$$

where p is the new load magnitude and p_{old} is the old load magnitude. If $0.2 \le r_{\max} \le 3.0$, the increment is accepted; and at the beginning of the following time increment, the load magnitudes are modified as follows:

$$0.2 \le r_{\max} < 0.5 \quad p = 1.5\, p_{\text{old}};$$
$$0.5 \le r_{\max} < 0.8 \quad p = 1.2\, p_{\text{old}};$$
$$0.8 \le r_{\max} < 1.5 \quad p = p_{\text{old}}; \text{ or}$$
$$1.5 \le r_{\max} \le 3.0 \quad p = p_{\text{old}}/1.2.$$

When you activate the above algorithm, the loading in a creep and/or swelling problem can be controlled on the basis of the maximum equivalent creep strain rate found in a defined element set. As a minimum requirement, this method is used to define a target equivalent creep strain rate; however, if required, it can also be used to define the target creep strain rate as a function of equivalent creep strain (measured as log strain), temperature, and other predefined field variables. The creep strain dependency curve at each temperature must always start at zero equivalent creep strain.

A solution-dependent amplitude is used to define the minimum and maximum limits of the loading (see *Defining a Solution-Dependent Amplitude for Superplastic Forming Analysis*). Any number or combination of loads can be used. The current value of $r_{\max}$ is available for output as discussed below.

Input File Usage: Use all of the following options:

**AMPLITUDE*, NAME=*name*, DEFINITION=SOLUTION DEPENDENT
**CLOAD*, **DLOAD*, **DSLOAD*, and/or **BOUNDARY* with AMPLITUDE=*name*
**CREEP STRAIN RATE CONTROL*, AMPLITUDE=*name*, ELSET=*elset*

The **AMPLITUDE* option must appear in the model definition portion of an input file, while the loading options (**CLOAD*, **DLOAD*, **DSLOAD*, and **BOUNDARY*) and the **CREEP STRAIN RATE CONTROL* option should appear in each relevant step definition.

Abaqus/CAE Usage: Creep strain rate control is not supported in Abaqus/CAE.

Elements

Rate-dependent plasticity (creep and swelling behavior) can be used with any continuum, shell, membrane, gasket, and beam element in Abaqus/Standard that has displacement degrees of freedom. Creep (but not swelling) can also be defined in the thickness direction of any gasket element in conjunction with the gasket behavior definition.

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables relate directly to creep and swelling models:

CEEQ
Equivalent creep strain, $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{cr} : \dot{\boldsymbol{\varepsilon}}^{cr}}\, dt$.

CESW
Magnitude of swelling strain.

SDEFRES
Deformation resistance, s. This output is relevant only for the Anand model.

The following output, which is relevant only for an analysis with creep strain rate loading control as discussed above, is printed at the beginning of an increment and is written automatically to the results file and output database file when any output to these files is requested:

RATIO
Maximum value of the ratio of the equivalent creep strain rate to the target creep strain rate, r_{max}.

AMPCU
Current value of the solution-dependent amplitude.

Annealing or Melting

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- **ANNEAL TEMPERATURE*
- *Specifying the annealing temperature of an elastic-plastic material*

Overview

This capability:

- is intended to model the effects of melting and resolidification in metals subjected to high-temperature processes or the effects of annealing at a material point when its temperature rises above a certain level;
- is available for only the Mises, Johnson-Cook, and Hill plasticity models;
- is intended to be used in conjunction with appropriate temperature-dependent material properties (in particular, the model assumes perfectly plastic behavior at or above the annealing or melting temperature); and
- can be modeled simply by defining an annealing or melting temperature.

Effects of Annealing or Melting

When the temperature of a material point exceeds a user-specified value called the annealing temperature, Abaqus assumes that the material point loses its hardening memory. The effect of prior work hardening is removed by setting the equivalent plastic strain to zero. For kinematic and combined hardening models the backstress tensor is also reset to zero. If the temperature of the material point falls below the annealing temperature at a subsequent point in time, the material point can work harden again. Depending on the temperature history a material point may lose and accumulate memory several times, which in the context of modeling melting would correspond to repeated melting and resolidification. Any accumulated material damage is not healed when the annealing temperature is reached. Damage will continue to accumulate after annealing according to any damage model in effect (see *About Damage and Failure for Ductile Metals*).

In Abaqus/Explicit an annealing step can be defined to simulate the annealing process for the entire model, independent of temperature; see *Annealing* for details.

Material Properties

The annealing temperature is a material property that can optionally be defined as a function of field variables. This material property must be used in conjunction with an appropriate definition of material properties as functions of temperature for the Mises plasticity model. In particular, the hardening behavior must be defined as a function of temperature and zero hardening must be specified at or above the annealing temperature. In general, hardening receives contributions from two sources. The first source of hardening can be classified broadly as static, and its effect is measured by the rate of change of the yield stress with respect to the plastic strain at a fixed strain rate. The second source of hardening can be classified broadly as rate dependent, and its effect is measured by the rate of change of the yield stress with respect to the strain rate at a fixed plastic strain.

For the Mises plasticity model, if the material data that describe hardening (both static and rate-dependent contributions) are completely specified through tabular input of yield stress versus plastic strain at different

values of the strain rate (see *Rate-Dependent Yield*), the (temperature-dependent) static part of the hardening at each strain rate is specified by defining several yield stress versus plastic strain curves (each at a different temperature). For metals the yield stress at a fixed strain rate typically decreases with increasing temperature. Abaqus expects the hardening at each strain rate to vanish at or above the annealing temperature and issues an error message if you specify otherwise in the material definition. Zero (static) hardening can be specified by simply specifying a single data point (at zero plastic strain) in the yield stress versus plastic strain curve at or above the annealing temperature. In addition, you must also ensure that at or above the annealing temperature, the yield stress does not vary with the strain rate. This can be accomplished by specifying the same value of yield stress at all values of strain rate in the single data point approach discussed above.

Alternatively, the static part of the hardening can be defined at zero strain rate, and the rate-dependent part can be defined utilizing the overstress power law (see *Rate-Dependent Yield*). In that case, zero static hardening at or above the annealing temperature can be specified by specifying a single data point (at zero plastic strain) in the yield stress versus plastic strain curve at or above the annealing temperature. The overstress power law parameters can also be appropriately selected to ensure that at or above the annealing temperature the yield stress does not vary with strain rate. This can be accomplished by selecting a large value for the parameter D (relative to the static yield stress) and setting the parameter $n = 1$.

For hardening defined in Abaqus/Standard with user subroutine *UHARD*, Abaqus/Standard checks the hardening slope at or above the annealing temperature during the actual computations and issues an error message if appropriate.

The Johnson-Cook plasticity model in Abaqus/Explicit requires a separate melting temperature to define the hardening behavior. If the annealing temperature is defined to be less than the melting temperature specified for the metal plasticity model, the hardening memory is removed at the annealing temperature and the melting temperature is used strictly to define the hardening function. Otherwise, the hardening memory is removed automatically at the melting temperature.

Input File Usage:

**ANNEAL TEMPERATURE*

Abaqus/CAE Usage:

Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Anneal Temperature**

Example: Annealing or Melting

The following input is an example of a typical usage of the annealing or melting capability. It is assumed that you have defined the static stress versus plastic strain behavior (see *Figure 1*) for the isotropic hardening model at three different temperatures, including the annealing temperature. It is also assumed that the plastic behavior is rate independent.

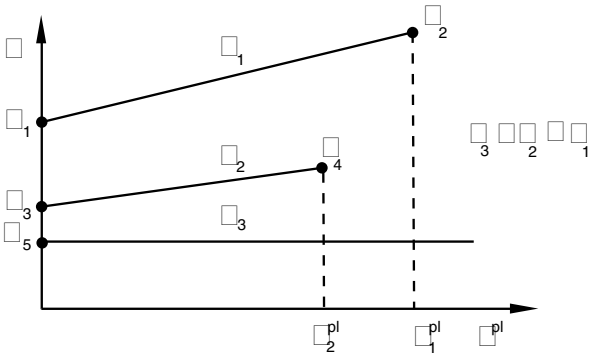


Figure 1: Stress versus plastic strain behavior.

The plastic response corresponds to linear hardening below the annealing temperature and perfect plasticity at the annealing temperature. The elastic properties, which may also be temperature dependent, are not shown.

Plasticity Data, Isotropic Hardening:		
Yield Stress	**Plastic Strain**	**Temperature**
σ_1	0	θ_1
σ_2	ϵ_1^{pl}	θ_1
σ_3	0	θ_2
σ_4	ϵ_2^{pl}	θ_2
σ_5	0	θ_3
Anneal Temperature: θ_3		

Elements

This capability can be used with all elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

Only the equivalent plastic strain (output variable PEEQ) and the backstress (output variable ALPHA) are reset to zero at the melting temperature. The plastic strain tensor (output variable PE) is not reset to zero and provides a measure of the total plastic deformation during the analysis. In Abaqus/Standard the plastic strain tensor also provides a measure of the plastic strain magnitude (output variable PEMAG).

Hill Anisotropic Yield/Creep

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Classical Metal Plasticity*
- *Critical State (Clay) Plasticity Model*
- *Crushable Foam Plasticity Models*
- *Extended Drucker-Prager Models*
- *Models for Metals Subjected to Cyclic Loading*
- *Rate-Dependent Plasticity: Creep and Swelling*
- **POTENTIAL*
- *Defining anisotropic yield and creep*

Overview

Hill anisotropic yield and/or creep:

- can be used for materials that exhibit different yield and/or creep behavior in different directions;
- is introduced through user-defined stress ratios that are applied in the quadratic Hill's potential function;
- can be used only with the metal plasticity and, in Abaqus/Standard, the metal creep material models;
- can be used with the extended Drucker-Prager, critical state (clay) plasticity, and crushable foam plasticity models;
- is available for the nonlinear isotropic/kinematic hardening model in Abaqus/Explicit (*Models for Metals Subjected to Cyclic Loading*); and
- can be used with the models of progressive damage and failure (*About Damage and Failure for Ductile Metals*) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh.

Yield and Creep Stress Ratios

Anisotropic yield or creep behavior using quadratic Hill's potential is modeled through the use of yield or creep stress ratios, R_{ij}. In the case of anisotropic yield the yield ratios are defined with respect to a reference yield stress, σ^0 (given for the metal plasticity definition), such that if σ_{ij} is applied as the only nonzero stress, the corresponding yield stress is $R_{ij}\sigma^0$. The plastic flow rule is defined below.

In the case of anisotropic creep, the R_{ij} are creep ratios used to scale the stress value when the creep strain rate is calculated. Thus, if σ_{11} is the only nonzero stress, the equivalent stress, $\tilde{q}$, used in the user-defined creep law is $\tilde{q} = \frac{|\sigma_{11}|}{R_{11}}$.

Yield and creep stress ratios can be defined as constants or as tabular functions of temperature and predefined field variables. A local orientation must be used to define the direction of anisotropy (see *Orientations*).

Input File Usage: Use the following option to define anisotropic yield or creep:

**POTENTIAL*, TYPE=HILL (default)

This option must appear immediately after the **PLASTIC* or the **CREEP* material option data to which it applies. Thus, if anisotropic metal plasticity and anisotropic creep behavior are both required, the **POTENTIAL* option must appear twice in the material definition, once after the metal plasticity data and again after the creep data.

Abaqus/CAE Usage: Use one of the following models:

Property module: material editor:
Mechanical->Plasticity->Plastic: **Suboptions->Potential**
Mechanical->Plasticity->Creep: **Suboptions->Potential**

Anisotropic Yield

Hill's potential function is a simple extension of the Mises function, which can be expressed in terms of rectangular Cartesian stress components as

$$f(\boldsymbol{\sigma}) = \sqrt{F(\sigma_{22}-\sigma_{33})^2 + G(\sigma_{33}-\sigma_{11})^2 + H(\sigma_{11}-\sigma_{22})^2 + 2L{\sigma_{23}}^2 + 2M{\sigma_{31}}^2 + 2N{\sigma_{12}}^2},$$

where $F, G, H, L, M,$ and N are constants obtained by tests of the material in different orientations. They are defined as

$$F = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{22}^2}+\frac{1}{\bar{\sigma}_{33}^2}-\frac{1}{\bar{\sigma}_{11}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{22}^2}+\frac{1}{R_{33}^2}-\frac{1}{R_{11}^2}\right),$$

$$G = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{33}^2}+\frac{1}{\bar{\sigma}_{11}^2}-\frac{1}{\bar{\sigma}_{22}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{33}^2}+\frac{1}{R_{11}^2}-\frac{1}{R_{22}^2}\right),$$

$$H = \frac{(\sigma^0)^2}{2}\left(\frac{1}{\bar{\sigma}_{11}^2}+\frac{1}{\bar{\sigma}_{22}^2}-\frac{1}{\bar{\sigma}_{33}^2}\right) = \frac{1}{2}\left(\frac{1}{R_{11}^2}+\frac{1}{R_{22}^2}-\frac{1}{R_{33}^2}\right),$$

$$L = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{23}}\right)^2 = \frac{3}{2R_{23}^2},$$

$$M = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{13}}\right)^2 = \frac{3}{2R_{13}^2},$$

$$N = \frac{3}{2}\left(\frac{\tau^0}{\bar{\sigma}_{12}}\right)^2 = \frac{3}{2R_{12}^2},$$

where each $\bar{\sigma}_{ij}$ is the measured yield stress value when σ_{ij} is applied as the only nonzero stress component; σ^0 is the user-defined reference yield stress specified for the metal plasticity definition; R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, and R_{23} are anisotropic yield stress ratios; and $\tau^0 = \sigma^0/\sqrt{3}$. Therefore, the six yield stress ratios are defined as follows (in the order in which you must provide them):

$$\frac{\bar{\sigma}_{11}}{\sigma^0},\quad \frac{\bar{\sigma}_{22}}{\sigma^0},\quad \frac{\bar{\sigma}_{33}}{\sigma^0},\quad \frac{\bar{\sigma}_{12}}{\tau^0},\quad \frac{\bar{\sigma}_{13}}{\tau^0},\quad \frac{\bar{\sigma}_{23}}{\tau^0}.$$

Because of the form of the yield function, all of these ratios must be positive. If the constants *F*, *G*, and *H* are positive, the yield function is always well defined. However, if one or more of these constants is negative, the yield function might be undefined for some stress states because the quantity under the square root is negative.

The flow rule is

$$d\boldsymbol{\varepsilon}^{pl} = d\lambda \frac{\partial f}{\partial \boldsymbol{\sigma}} = \frac{d\lambda}{f}\mathbf{b},$$

where, from the definition of f above,

$$\mathbf{b} = \begin{bmatrix} -G(\sigma_{33} - \sigma_{11}) + H(\sigma_{11} - \sigma_{22}) \\ F(\sigma_{22} - \sigma_{33}) - H(\sigma_{11} - \sigma_{22}) \\ -F(\sigma_{22} - \sigma_{33}) + G(\sigma_{33} - \sigma_{11}) \\ 2N\sigma_{12} \\ 2M\sigma_{31} \\ 2L\sigma_{23} \end{bmatrix}.$$

Input File Usage: Use both of the following options:

**PLASTIC*
**POTENTIAL*, TYPE=HILL (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Plastic**: **Suboptions**->**Potential**

Anisotropic Creep

For anisotropic creep in Abaqus/Standard Hill's function can be expressed as

$$\tilde{q}(\boldsymbol{\sigma}) = \sqrt{F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L{\sigma_{23}}^2 + 2M{\sigma_{31}}^2 + 2N{\sigma_{12}}^2},$$

where $\tilde{q}(\boldsymbol{\sigma})$ is the equivalent stress and F, G, H, L, M, and N are constants obtained by tests of the material in different orientations. The constants are defined with the same general relations as those used for anisotropic yield (above). However, the reference yield stress, $\boldsymbol{\sigma}^0$, is replaced by the uniaxial equivalent deviatoric stress, $\tilde{q}$ (found in the creep law), and R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, and R_{23} are referred to as "anisotropic creep stress ratios." The six creep stress ratios are, therefore, defined as follows (in the order in which they must be provided):

$$\frac{\sigma_{11}}{\tilde{q}}, \quad \frac{\sigma_{22}}{\tilde{q}}, \quad \frac{\sigma_{33}}{\tilde{q}}, \quad \frac{\sigma_{12}}{\tilde{q}/\sqrt{3}}, \quad \frac{\sigma_{13}}{\tilde{q}/\sqrt{3}}, \quad \frac{\sigma_{23}}{\tilde{q}/\sqrt{3}}.$$

You must define the ratios R_{ij} in each direction that will be used to scale the stress value when the creep strain rate is calculated. If all six R_{ij} values are set to unity, isotropic creep is obtained.

Input File Usage: Use both of the following options:

**CREEP*
**POTENTIAL*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Creep**: **Suboptions**->**Potential**

Defining Anisotropic Yield Behavior on the Basis of Strain Ratios (Lankford's *r*-Values)

As discussed above, Hill's anisotropic plasticity potential is defined in Abaqus from user input consisting of ratios of yield stress in different directions with respect to a reference stress. However, in some cases, such as sheet metal forming applications, it is common to find the anisotropic material data given in terms of ratios of width strain to thickness strain. Mathematical relationships are then necessary to convert the strain ratios to stress ratios that can be input into Abaqus.

In sheet metal forming applications, we are generally concerned with plane stress conditions. Consider x, y to be the "rolling" and "cross" directions in the plane of the sheet; z is the thickness direction. From a design viewpoint, the type of anisotropy usually required is that in which the sheet is isotropic in the plane and has an increased strength in the thickness direction, which is normally referred to as transverse anisotropy. Planar anisotropy is another type of anisotropy, characterized by different strengths in different directions in the plane of the sheet.

In a simple tension test performed in the x-direction in the plane of the sheet, the flow rule for this potential (given above) defines the incremental strain ratios (assuming small elastic strains) as

$$d\varepsilon_{11} : d\varepsilon_{22} : d\varepsilon_{33} = G + H : -H : -G.$$

Therefore, the ratio of width to thickness strain, often referred to as Lankford's r-value, is

$$r_x = \frac{d\varepsilon_{22}}{d\varepsilon_{33}} = \frac{H}{G}.$$

Similarly, for a simple tension test performed in the y-direction in the plane of the sheet, the incremental strain ratios are

$$d\varepsilon_{11} : d\varepsilon_{22} : d\varepsilon_{33} = -H : F + H : -F,$$

and

$$r_y = \frac{d\varepsilon_{11}}{d\varepsilon_{33}} = \frac{H}{F}.$$

Transverse Anisotropy

A transversely anisotropic material is one where $r_x = r_y$. If we define σ^0 in the metal plasticity model to be equal to $\overline{\sigma}_{11}$,

$$R_{11} = R_{22} = 1$$

and, using the relationships above,

$$R_{33} = \sqrt{\frac{r_x+1}{2}}.$$

If $r_x = 1$ (isotropic material), $R_{33} = 1$ and the Mises isotropic plasticity model is recovered.

Planar Anisotropy

In the case of planar anisotropy r_x and r_y are different and R_{11}, R_{22}, R_{33} will all be different. If we define σ^0 in the metal plasticity model to be equal to $\overline{\sigma}_{11}$,

$$R_{11} = 1$$

and, using the relationships above, we obtain

$$R_{22} = \sqrt{\frac{r_y(r_x+1)}{r_x(r_y+1)}}, \qquad R_{33} = \sqrt{\frac{r_y(r_x+1)}{(r_x+r_y)}}.$$

Again, if $r_x = r_y = 1$, $R_{22} = R_{33} = 1$ and the Mises isotropic plasticity model is recovered.

General Anisotropy

Thus far, we have only considered loading applied along the axes of anisotropy. To derive a more general anisotropic model in plane stress, the sheet must be loaded in one other direction in its plane. Suppose we perform a simple tension test at an angle α to the *x*-direction; then, from equilibrium considerations we can write the nonzero stress components as

$$\sigma_{11} = \sigma \cos^2 \alpha, \quad \sigma_{22} = \sigma \sin^2 \alpha, \quad \sigma_{12} = \sigma \sin \alpha \cos \alpha,$$

where σ is the applied tensile stress. Substituting these values in the flow equations and assuming small elastic strains yields

$$d\varepsilon_{11} = \left[(G+H) \cos^2 \alpha - H \sin^2 \alpha\right] \frac{\sigma}{f} d\lambda,$$

$$d\varepsilon_{22} = \left[(F+H) \sin^2 \alpha - H \cos^2 \alpha\right] \frac{\sigma}{f} d\lambda,$$

$$d\varepsilon_{33} = -\left[F \sin^2 \alpha + G \cos^2 \alpha\right] \frac{\sigma}{f} d\lambda, \text{ and}$$

$$d\gamma_{12} = [2N \sin \alpha \cos \alpha] \frac{\sigma}{f} d\lambda.$$

Assuming small geometric changes, the width strain increment (the increment of strain at right angles to the direction of loading, α) is written as

$$d\varepsilon_{\alpha+\frac{\pi}{2}} = d\varepsilon_{11} \sin^2 \alpha + d\varepsilon_{22} \cos^2 \alpha - d\gamma_{12} \sin \alpha \cos \alpha,$$

and Lankford's *r*-value for loading at an angle α is

$$r_\alpha = \frac{d\varepsilon_{\alpha+\frac{\pi}{2}}}{d\varepsilon_{33}} = \frac{H+(2N-F-G-4H) \sin^2 \alpha \cos^2 \alpha}{F \sin^2 \alpha + G \cos^2 \alpha}.$$

One of the more commonly performed tests is that in which the loading direction is at 45°. In this case

$$r_{45} = \frac{2N-(F+G)}{2(F+G)} \quad \text{or} \quad \frac{N}{G} = \left(r_{45} + \frac{1}{2}\right)\left(1 + \frac{r_x}{r_y}\right).$$

If σ^0 is equal to $\overline{\sigma}_{11}$ in the metal plasticity model, $R_{11} = 1$. R_{22}, R_{33} are as defined before for transverse or planar anisotropy and, using the relationships above,

$$R_{12} = \sqrt{\frac{3(r_x+1)r_y}{(2r_{45}+1)(r_x+r_y)}}.$$

Progressive Damage and Failure

Hill anisotropic yield can be used with the models of progressive damage and failure discussed in *About Damage and Failure for Ductile Metals*. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations.

Input File Usage: Use the following options:

**PLASTIC*
**DAMAGE INITIATION*
**DAMAGE EVOLUTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*damage initiation type***: specify the damage initiation criterion: **Suboptions->Damage Evolution**: specify the damage evolution parameters

Initial Conditions

When you want to study the behavior of a material that has already been subjected to some work hardening, you can prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, by specifying the conditions directly (*Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

User Subroutine Specification in Abaqus/Standard

For more complicated cases, you can define initial conditions in Abaqus/Standard through the user subroutine *HARDINI*.

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

You can define Hill anisotropic yield for any element that can be used with the following models:

- Classical metal plasticity models in Abaqus (*Classical Metal Plasticity*)
- Critical state (clay) plasticity model (*Critical State (Clay) Plasticity Model*)
- Extended Drucker-Prager material models (*Extended Drucker-Prager Models*)
- Crushable foam plasticity models (*Crushable Foam Plasticity Models*)

You cannot define Hill anisotropic yield for one-dimensional elements in Abaqus/Explicit (beams and trusses). In Abaqus/Standard it can also be defined for any element that can be used with the linear kinematic hardening plasticity model (*Models for Metals Subjected to Cyclic Loading*) but not with the nonlinear isotropic/kinematic hardening model. Likewise, anisotropic creep with Hill's function can be defined for any element that can be used with the classical metal creep model in Abaqus/Standard (*Rate-Dependent Plasticity: Creep and Swelling*).

Output

The standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*) and all output variables associated with the following models are available when you define anisotropic yield and creep:

- Creep model (*Rate-Dependent Plasticity: Creep and Swelling*)
- Classical metal plasticity models (*Classical Metal Plasticity*)
- Critical state (clay) plasticity model (*Critical State (Clay) Plasticity Model*)
- Extended Drucker-Prager material models (*Extended Drucker-Prager Models*)
- Crushable foam plasticity models (*Crushable Foam Plasticity Models*)
- Linear kinematic hardening plasticity model (*Models for Metals Subjected to Cyclic Loading*)

The following variables have special meaning if anisotropic yield and creep are defined:

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \dot{\bar{\varepsilon}}^{pl} dt = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \frac{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{pl} dt}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

CEEQ

Equivalent creep strain, $\bar{\varepsilon}^{cr} = \int_0^t \dot{\bar{\varepsilon}}^{cr} dt = \int_0^t \frac{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{cr} dt}{\sigma^0}$.

YIELDS

Yield stress, σ^0.

YIELDPOT

Yield potential, $f(\boldsymbol{\sigma})$. This output variable is available only when Hill anisotropic yield is used with the metal plasticity model.

Nonquadratic Yield

Products: Abaqus/Standard Abaqus/Explicit

References:

- *Inelastic Behavior*
- *Classical Metal Plasticity*
- **POTENTIAL*
- *Barlat anisotropic plasticity*

Overview

Nonquadratic yield surfaces:

- are available in the form of Tresca or Hosford yield surfaces with associated plastic flow to model isotropic yield or in the form of Barlat yield surfaces with associated plastic flow to model anisotropic yield;
- are introduced through user-defined exponents and coefficients of nonquadratic stress potential functions;
- can be used with perfect plasticity or isotropic hardening behavior; and
- can be used with progressive damage and failure models (*About Damage and Failure for Ductile Metals*) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh.

Nonquadratic Yield Surfaces

Nonquadratic yield surfaces with associated plastic flow for classical metal plasticity are available. You can specify the Tresca or Hosford yield surface for isotropic yield or the Barlat yield surface for anisotropic yield.

Tresca Yield Surface

The Tresca yield potential for isotropic yielding takes the form

$$f(\boldsymbol{\sigma}) = \max(|\sigma_1 - \sigma_2|, |\sigma_1 - \sigma_3|, |\sigma_2 - \sigma_3|) \leq \sigma^0,$$

where σ^0 is the uniaxial yield stress and σ_1, σ_2, and σ_3 are the principal stresses.

Input File Usage: Use both of the following options:

**PLASTIC (to specify the uniaxial yield stress, σ^0)*
**POTENTIAL*, TYPE=TRESCA

Hosford Yield Surface

The Hosford yield surface for isotropic yielding is a generalization of the Mises yield surface (*Mises Yield Surface*). The Hosford stress potential function is

$$f(\boldsymbol{\sigma}) = \left(\tfrac{1}{2}(|\sigma_1 - \sigma_2|^\alpha + |\sigma_1 - \sigma_3|^\alpha + |\sigma_2 - \sigma_3|^\alpha)\right)^{\frac{1}{\alpha}} \leq \sigma^0,$$

where σ^0 is the uniaxial yield stress; σ_1, σ_2, and σ_3 are the principal stresses; and α is the exponent. When $\alpha = 1$ or α goes to infinity, the Hosford yield function reduces to the Tresca yield function. When $\alpha = 2$ or $\alpha = 4$, the Hosford yield function reduces to the Mises yield function.

Input File Usage: Use both of the following options:

**PLASTIC (to specify the uniaxial yield stress, σ^0)*
**POTENTIAL*, TYPE=HOSFORD, POWER=α

Barlat Yield Surface

The Barlat yield surface allows you to model complex anisotropic yielding behavior. The plastic flow rule is defined below. Anisotropic yield with Barlat's potential is modeled through the use of anisotropy coefficients that are defined with respect to a reference yield stress, σ^0.

Both three-dimensional and plane-stress formulations are available. The three-dimensional formulations include the Yld2004-18p potential function that contains 18 coefficients and the Yld91 potential function that contains 6 coefficients. The Yld2000-2d potential function uses a plane-stress formulation with 8 coefficients. Although designed for plane-stress conditions, you can extend the Yld2000-2d potential function to full three-dimensional stress states through simplified assumptions.

The anisotropy coefficients can be defined as constants or as tabular functions of temperature and predefined field variables. You must use a local orientation to define the direction of anisotropy (see *Orientations*).

Input File Usage: Use both of the following options to define the Barlat Yld2004-18p yield surface:

**PLASTIC (to specify the reference yield stress, σ^0)*
**POTENTIAL*, TYPE=BARLAT, POWER=α
c'_{1122}, c'_{1133}, c'_{2211}, c'_{2233}, c'_{3311}, c'_{3322}, c'_{1212}, c'_{1313}
c'_{2323}, c''_{1122}, c''_{1133}, c''_{2211}, c''_{2233}, c''_{3311}, c''_{3322}, c''_{1212}
c''_{1313}, c''_{2323}

Use both of the following options to define the Barlat Yld91 yield surface:

**PLASTIC (to specify the reference yield stress, σ^0)*
**POTENTIAL*, TYPE=BARLAT91, POWER=α
$\overline{a}$, $\overline{b}$, $\overline{c}$, $\overline{f}$, $\overline{g}$, $\overline{h}$

Use both of the following options to define the Barlat Yld2000-2d yield surface:

**PLASTIC (to specify the reference yield stress, σ^0)*
**POTENTIAL*, TYPE=BARLAT2000, POWER=α
α_1, α_2, α_3, α_4, α_5, α_6, α_7, α_8

Barlat Plasticity

Barlat's anisotropic yield potential is based on linear transformations of stress. The stress potential function (Yld2004-18p) proposed by *Barlat et al. (2005)* is

$$f(\sigma) = \left(\frac{1}{4}\left(\left|\tilde{S}'_1 - \tilde{S}''_1\right|^\alpha + \left|\tilde{S}'_1 - \tilde{S}''_2\right|^\alpha + \left|\tilde{S}'_1 - \tilde{S}''_3\right|^\alpha + \left|\tilde{S}'_2 - \tilde{S}''_1\right|^\alpha + \left|\tilde{S}'_2 - \tilde{S}''_2\right|^\alpha + \left|\tilde{S}'_2 - \tilde{S}''_3\right|^\alpha + \left|\tilde{S}'_3 - \tilde{S}''_1\right|^\alpha + \left|\tilde{S}'_3 - \tilde{S}''_2\right|^\alpha + \left|\tilde{S}'_3 - \tilde{S}''_3\right|^\alpha\right)\right)^{\frac{1}{\alpha}},$$

where α is the exponent and $\tilde{S}'_i$, $\tilde{S}''_i$ (i=1, 2, 3) are principal values of stress tensors $\tilde{\mathbf{S}}'$ and $\tilde{\mathbf{S}}''$, respectively.

The principal values of a stress tensor $\tilde{\mathbf{S}}$ ($\tilde{\mathbf{S}}'$ or $\tilde{\mathbf{S}}''$) are the roots of the characteristic equation

$$-\tilde{S}_k^3 + \tilde{I}_1\tilde{S}_k^2 - \tilde{I}_2\tilde{S}_k + \tilde{I}_3 = 0,$$

where the first, second, and third stress invariants are

$$\tilde{I}_1 = \tilde{S}_{11} + \tilde{S}_{22} + \tilde{S}_{33},$$
$$\tilde{I}_2 = \tilde{S}_{11}\tilde{S}_{22} + \tilde{S}_{11}\tilde{S}_{33} + \tilde{S}_{22}\tilde{S}_{33} - \tilde{S}_{12}^2 - \tilde{S}_{13}^2 - \tilde{S}_{23}^2,$$
$$\tilde{I}_3 = \tilde{S}_{11}\tilde{S}_{22}\tilde{S}_{33} + 2\tilde{S}_{12}\tilde{S}_{13}\tilde{S}_{23} - \tilde{S}_{12}^2\tilde{S}_{33} - \tilde{S}_{13}^2\tilde{S}_{22} - \tilde{S}_{23}^2\tilde{S}_{11}.$$

The principal values are

$$\tilde{S}_1 = \frac{2}{3}\tilde{q}\cos\left(\tilde{\Theta}\right) - \tilde{p},$$
$$\tilde{S}_2 = \frac{2}{3}\tilde{q}\cos\left(\tilde{\Theta} - \frac{2\pi}{3}\right) - \tilde{p},$$
$$\tilde{S}_3 = \frac{2}{3}\tilde{q}\cos\left(\tilde{\Theta} - \frac{4\pi}{3}\right) - \tilde{p},$$

where the deviatoric polar angle is

$$\tilde{\Theta} = \frac{1}{3}\arccos\left(\left(\frac{\tilde{r}}{\tilde{q}}\right)^3\right),$$

and the stress invariants are

$$\tilde{p} = -\frac{1}{3}\tilde{I}_1,$$
$$\tilde{q} = \sqrt{\tilde{I}_1^2 - 3\tilde{I}_2},$$
$$\tilde{r} = 3\left(\frac{1}{27}\tilde{I}_1^3 - \frac{1}{6}\tilde{I}_1\tilde{I}_2 + \frac{1}{2}\tilde{I}_3\right)^{\frac{1}{3}}.$$

The tensors $\tilde{\mathbf{S}}'$ and $\tilde{\mathbf{S}}''$ are defined by two linear transformations of the deviatoric stress $\mathbf{S}$:

$$\tilde{\mathbf{s}}' = \mathbf{C}'\mathbf{S} = \mathbf{C}'\mathbf{T}\boldsymbol{\sigma} \equiv \mathbf{L}'\boldsymbol{\sigma},$$
$$\tilde{\mathbf{s}}'' = \mathbf{C}''\mathbf{S} = \mathbf{C}''\mathbf{T}\boldsymbol{\sigma} \equiv \mathbf{L}''\boldsymbol{\sigma},$$

where $\mathbf{C}'$ and $\mathbf{C}''$ are transformation tensors that contain anisotropy coefficients and the tensor $\mathbf{T}$ transforms the Cauchy stress, $\boldsymbol{\sigma}$, to the deviatoric stress. The transformations can be expressed in matrix form:

$$\begin{Bmatrix} \widetilde{S}'_{11} \\ \widetilde{S}'_{22} \\ \widetilde{S}'_{33} \\ \widetilde{S}'_{12} \\ \widetilde{S}'_{13} \\ \widetilde{S}'_{23} \end{Bmatrix} = \begin{bmatrix} 0 & -c'_{1122} & -c'_{1133} & 0 & 0 & 0 \\ -c'_{2211} & 0 & -c'_{2233} & 0 & 0 & 0 \\ -c'_{3311} & -c'_{3322} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2c'_{1212} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2c'_{1313} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2c'_{2323} \end{bmatrix} \begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{12} \\ S_{13} \\ S_{23} \end{Bmatrix} =$$

$$\frac{1}{3}\begin{bmatrix} c'_{1122}+c'_{1133} & -2c'_{1122}+c'_{1133} & c'_{1122}-2c'_{1133} & 0 & 0 & 0 \\ -2c'_{2211}+c'_{2233} & c'_{2211}+c'_{2233} & c'_{2211}-2c'_{2233} & 0 & 0 & 0 \\ -2c'_{3311}+c'_{3322} & c'_{3311}-2c'_{3322} & c'_{3311}+c'_{3322} & 0 & 0 & 0 \\ 0 & 0 & 0 & 6c'_{1212} & 0 & 0 \\ 0 & 0 & 0 & 0 & 6c'_{1313} & 0 \\ 0 & 0 & 0 & 0 & 0 & 6c'_{2323} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix},$$

and

$$\begin{Bmatrix} \widetilde{S}''_{11} \\ \widetilde{S}''_{22} \\ \widetilde{S}''_{33} \\ \widetilde{S}''_{12} \\ \widetilde{S}''_{13} \\ \widetilde{S}''_{23} \end{Bmatrix} = \begin{bmatrix} 0 & -c''_{1122} & -c''_{1133} & 0 & 0 & 0 \\ -c''_{2211} & 0 & -c''_{2233} & 0 & 0 & 0 \\ -c''_{3311} & -c''_{3322} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2c''_{1212} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2c''_{1313} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2c''_{2323} \end{bmatrix} \begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{12} \\ S_{13} \\ S_{23} \end{Bmatrix} =$$

$$\frac{1}{3}\begin{bmatrix} c''_{1122}+c''_{1133} & -2c''_{1122}+c''_{1133} & c''_{1122}-2c''_{1133} & 0 & 0 & 0 \\ -2c''_{2211}+c''_{2233} & c''_{2211}+c''_{2233} & c''_{2211}-2c''_{2233} & 0 & 0 & 0 \\ -2c''_{3311}+c''_{3322} & c''_{3311}-2c''_{3322} & c''_{3311}+c''_{3322} & 0 & 0 & 0 \\ 0 & 0 & 0 & 6c''_{1212} & 0 & 0 \\ 0 & 0 & 0 & 0 & 6c''_{1313} & 0 \\ 0 & 0 & 0 & 0 & 0 & 6c''_{2323} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}$$

The 18 anisotropy coefficients c'_{ijkl}, c''_{ijkl} can be calibrated from experiments.

If $c'_{iijj} = c''_{iijj} = 1$ and $c'_{ijij} = c''_{ijij} = 0.5$, the yield function reduces to the Hosford isotropic yield function. In addition, when $\alpha = 2$ or $\alpha = 4$, the Mises isotropic plasticity model is recovered.

The flow rule is

$$d\boldsymbol{\varepsilon}^{pl} = d\lambda \frac{\partial f}{\partial \boldsymbol{\sigma}},$$

where, from the definition of f above,

$$\frac{\partial f}{\partial \sigma_{ij}} = \frac{\partial f}{\partial \widetilde{S}'_p}\frac{\partial \widetilde{S}'_p}{\partial \tilde{I}'_q}\frac{\partial \tilde{I}'_q}{\partial \widetilde{S}'_{rs}}L'_{rsij} + \frac{\partial f}{\partial \widetilde{S}''_p}\frac{\partial \widetilde{S}''_p}{\partial \tilde{I}''_q}\frac{\partial \tilde{I}''_q}{\partial \widetilde{S}''_{rs}}L''_{rsij}.$$

Barlat Plasticity (Yld91)

When the Barlat plasticity considers only one linear transformation (that is, $\tilde{\mathbf{S}}' = \tilde{\mathbf{S}}'' = \tilde{\mathbf{S}}$), the yield surface reduces to the Barlat Yld91 yield surface (*Barlat et al., 1991*) as

$$f(\boldsymbol{\sigma}) = \left(\tfrac{1}{2}\left(\left|\tilde{S}_1 - \tilde{S}_2\right|^{\alpha} + \left|\tilde{S}_1 - \tilde{S}_3\right|^{\alpha} + \left|\tilde{S}_2 - \tilde{S}_3\right|^{\alpha}\right)\right)^{\frac{1}{\alpha}},$$

where α is the exponent; and $\tilde{S}_1$, $\tilde{S}_2$, and $\tilde{S}_3$ are principal values of the stress tensor $\tilde{\mathbf{S}}$.

The linear transformation is defined as

$$\tilde{\mathbf{S}} = \mathbf{L}\boldsymbol{\sigma},$$

or expressed in matrix form,

$$\begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{12} \\ S_{13} \\ S_{23} \end{Bmatrix} = \tfrac{1}{3}\begin{bmatrix} \bar{b}+\bar{c} & -\bar{c} & -\bar{b} & 0 & 0 & 0 \\ -\bar{c} & \bar{c}+\bar{a} & -\bar{a} & 0 & 0 & 0 \\ -\bar{b} & -\bar{a} & \bar{a}+\bar{b} & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\bar{h} & 0 & 0 \\ 0 & 0 & 0 & 0 & 3\bar{g} & 0 \\ 0 & 0 & 0 & 0 & 0 & 3\bar{f} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

The six anisotropy coefficients $\bar{a}$, $\bar{b}$, $\bar{c}$, $\bar{f}$, $\bar{g}$, and $\bar{h}$ can be calibrated from experiments. If all six coefficients are set to unity, the yield function reduces to the Hosford isotropic yield function. In addition, when $\alpha = 2$ or $\alpha = 4$, the Mises isotropic plasticity model is recovered.

Barlat Plasticity (Yld2000-2d)

The Barlat Yld2000-2d yield criterion (*Barlat et. al, 2000*) is widely used to model the plastic anisotropy of sheet metals under plane-stress conditions. The yield surface is defined as

$$f(\boldsymbol{\sigma}) = \left[\tfrac{1}{2}\left(\left|S_1' - S_2'\right|^{\alpha} + \left|2S_1'' + S_2''\right|^{\alpha} + \left|S_1'' + 2S_2''\right|^{\alpha}\right)\right]^{\frac{1}{\alpha}},$$

where α is the exponent; and S_1' and S_2' are principal values of the stress tensor $\mathbf{S}'$ and $\mathbf{S}''$, respectively:

$$S_1' = \tfrac{1}{2}\left(S_{11}' + S_{22}' + \sqrt{\left(S_{11}' - S_{22}'\right)^2 + 4{S_{12}'}^2}\right),$$

$$S_2' = \tfrac{1}{2}\left(S_{11}' + S_{22}' - \sqrt{\left(S_{11}' - S_{22}'\right)^2 + 4{S_{12}'}^2}\right),$$

$$S_1'' = \tfrac{1}{2}\left(S_{11}'' + S_{22}'' + \sqrt{\left(S_{11}'' - S_{22}''\right)^2 + 4{S_{12}''}^2}\right),$$

$$S_2'' = \tfrac{1}{2}\left(S_{11}'' + S_{22}'' - \sqrt{\left(S_{11}'' - S_{22}''\right)^2 + 4{S_{12}''}^2}\right).$$

The tensors $\mathbf{S}'$ and $\mathbf{S}''$ are defined by two linear transformations of the stress $\boldsymbol{\sigma}$:

$$\mathbf{S}' = \mathbf{L}'\boldsymbol{\sigma}$$

and

$$\mathbf{S}'' = \mathbf{L}''\boldsymbol{\sigma}.$$

The transformations can be expressed in matrix form:

$$\begin{Bmatrix} S'_{11} \\ S'_{22} \\ S'_{12} \end{Bmatrix} = \frac{1}{3} \begin{bmatrix} 2\alpha_1 & -\alpha_1 & 0 \\ -\alpha_2 & 2\alpha_2 & 0 \\ 0 & 0 & 3\alpha_7 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix},$$

and

$$\begin{Bmatrix} S''_{11} \\ S''_{22} \\ S''_{12} \end{Bmatrix} = \frac{1}{9} \begin{bmatrix} -2\alpha_3 + 2\alpha_4 + 8\alpha_5 - 2\alpha_6 & \alpha_3 - 4\alpha_4 - 4\alpha_5 + 4\alpha_6 & 0 \\ 4\alpha_3 - 4\alpha_4 - 4\alpha_5 + \alpha_6 & -2\alpha_3 + 8\alpha_4 + 2\alpha_5 - 2\alpha_6 & 0 \\ 0 & 0 & 9\alpha_8 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix}.$$

The eight anisotropy coefficients α_i ($i = 1, 8$) can be calibrated from experiments. If all eight coefficients are set to unity, and $\alpha = 2$, the Mises isotropic plasticity model is recovered.

The Barlat Yld2000-2d yield criterion can be extended to three-dimensional stress states with simplified assumptions (*Dunand et. al, 2012*). In the extended formulation, the principal values of the stress tensor $\mathbf{S}'$ and $\mathbf{S}''$ are:

$$S'_1 = \frac{1}{2}\left(S'_{11} + S'_{22} + \sqrt{\left(S'_{11} - S'_{22}\right)^2 + 4{S'_{12}}^2 + 4{S'_{13}}^2 + 4{S'_{23}}^2}\right),$$

$$S'_2 = \frac{1}{2}\left(S'_{11} + S'_{22} - \sqrt{\left(S'_{11} - S'_{22}\right)^2 + 4{S'_{12}}^2 + 4{S'_{13}}^2 + 4{S'_{23}}^2}\right),$$

$$S''_1 = \frac{1}{2}\left(S''_{11} + S''_{22} + \sqrt{\left(S''_{11} - S''_{22}\right)^2 + 4{S''_{12}}^2 + 4{S''_{13}}^2 + 4{S''_{23}}^2}\right),$$

$$S''_2 = \frac{1}{2}\left(S''_{11} + S''_{22} - \sqrt{\left(S''_{11} - S''_{22}\right)^2 + 4{S''_{12}}^2 + 4{S''_{13}}^2 + 4{S''_{23}}^2}\right).$$

The transformations can be expressed in the following matrix form:

$$\begin{Bmatrix} S'_{11} \\ S'_{22} \\ S'_{12} \\ S'_{13} \\ S'_{23} \end{Bmatrix} = \frac{1}{3} \begin{bmatrix} 2\alpha_1 & -\alpha_1 & -\alpha_1 & 0 & 0 & 0 \\ -\alpha_2 & 2\alpha_2 & -\alpha_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\alpha_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix},$$

and

$$\begin{Bmatrix} S''_{11} \\ S''_{22} \\ S''_{12} \\ S''_{13} \\ S''_{23} \end{Bmatrix} = \frac{1}{9} \begin{bmatrix} -2\alpha_3 + 2\alpha_4 + 8\alpha_5 - 2\alpha_6 & \alpha_3 - 4\alpha_4 - 4\alpha_5 + 4\alpha_6 & \alpha_3 + 2\alpha_4 - 4\alpha_5 - 2\alpha_6 & 0 & 0 & 0 \\ 4\alpha_3 - 4\alpha_4 - 4\alpha_5 + \alpha_6 & -2\alpha_3 + 8\alpha_4 + 2\alpha_5 - 2\alpha_6 & -2\alpha_3 - 4\alpha_4 + 2\alpha_5 + \alpha_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9\alpha_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix}.$$

Progressive Damage and Failure

Nonquadratic yield can be used with the progressive damage and failure models, as discussed in *About Damage and Failure for Ductile Metals*. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure

choices, including the removal of elements from the mesh because of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations.

Input File Usage: Use the following options:

**PLASTIC*
**DAMAGE INITIATION*
**DAMAGE EVOLUTION*

Elements

Nonquadratic yield is available with three-dimensional solid elements, plane-strain elements, plane-stress elements, shell elements, and membrane elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Explicit Output Variable Identifiers* and *Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning for the classical metal plasticity models:

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \dot{\bar{\varepsilon}}^{pl}\, dt = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}\, dt}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

YIELDS

Yield stress, σ^0.

YIELDPOT

Yield potential, $f(\boldsymbol{\sigma})$.

References

- Barlat, F., H. Aretz, J. W. Yoon, M. E. Karabin, J. C. Brem, and R. E. Dick, "Linear Transformation-based Anisotropic Yield Functions," *International Journal of Plasticity*, vol. 21, pp. 1009–1039, 2005.
- Barlat, F., D. J. Lege, and J. C. Brem, "A Six-Component Yield Function for Anisotropic Materials," *International Journal of Plasticity*, vol. 7, pp. 693–712, 1991.
- Barlat, F., J. C. Brem, J. W. Yoon, K. Chung, R. E. Dick, D. J. Lege, F. Pourboghrat, S. H. Choi, and E. Chu, "Plane Stress Yield Function for Aluminum Alloy Sheets—Part 1: Theory," *International Journal of Plasticity*, vol. 19, pp. 1297–1319, 2003.
- Dunand, M., A. P. Maertens, M. Luo, and D. Mohr, "Experiments and Modeling of Anisotropic Aluminum Extrusions under Multi-axial Loading – Part I: Plasticity," *International Journal of Plasticity*, vol. 36, pp. 34–49, 2012.

Johnson-Cook Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Classical Metal Plasticity*
- *Rate-Dependent Yield*
- *Equation of State*
- *Progressive Damage and Failure*
- *Dynamic Failure Models*
- *Annealing or Melting*
- *About the Material Library*
- *Inelastic Behavior*
- *Progressive Damage and Failure*
- **ANNEAL TEMPERATURE*
- **PLASTIC*
- **RATE DEPENDENT*
- **SHEAR FAILURE*
- **TENSILE FAILURE*
- **DAMAGE INITIATION*
- **DAMAGE EVOLUTION*
- *Using the Johnson-Cook hardening model to define classical metal plasticity*

Overview

The Johnson-Cook plasticity model:

- is suitable for high-strain-rate deformation of many materials, including most metals;
- is typically used in adiabatic transient dynamic simulations;
- can be used in conjunction with the Johnson-Cook dynamic failure model in Abaqus/Explicit;
- can be used to specify the plastic response for models defined using the parallel rheological framework (*Parallel Rheological Framework*) in Abaqus/Standard;
- can be used in conjunction with the tensile failure model to model tensile spall or a pressure cutoff in Abaqus/Explicit;
- can be used in conjunction with the progressive damage and failure models (*Progressive Damage and Failure*) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh; and
- must be used in conjunction with either the linear elastic material model (*Linear Elastic Behavior*), the equation of state material model (*Equation of State*), or the hyperelastic material model (*Hyperelastic Behavior of Rubberlike Materials*) in Abaqus/Standard.

Yield Surface and Flow Rule

A Mises yield surface with associated flow is used in the Johnson-Cook plasticity model.

Johnson-Cook Hardening

Johnson-Cook hardening is a particular type of isotropic hardening where the static yield stress, σ^0, is assumed to be of the form

$$\sigma^0 = \left[A + B\left(\bar{\varepsilon}^{pl}\right)^n\right]\left(1 - \hat{\theta}^m\right),$$

where $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain and A, B, n, and m are material parameters. $\hat{\theta}$ is the nondimensional temperature defined as

$$\hat{\theta} \equiv \begin{cases} 0 & \text{for} \quad \theta < \theta_{\text{transition}} \\ \left(\theta - \theta_{\text{transition}}\right) / \left(\theta_{\text{melt}} - \theta_{\text{transition}}\right) & \text{for} \quad \theta_{\text{transition}} \le \theta \le \theta_{\text{melt}} \\ 1 & \text{for} \quad \theta > \theta_{\text{melt}} \end{cases},$$

where θ is the current temperature, θ_{melt} is the melting temperature, and $\theta_{\text{transition}}$ is the transition temperature defined as the one at or below which there is no temperature dependence of the yield stress. The material parameters A, B, and n must be measured at or below the transition temperature. The material parameter m should be determined based on measurements above the transition temperature. Temperature dependency of σ^0 is ignored if you specify a zero value or if you do not specify a value for m.

When $\theta \ge \theta_{\text{melt}}$, the material will be melted and will behave like a fluid; there will be no shear resistance since $\sigma^0 = 0$. The hardening memory will be removed by setting the equivalent plastic strain to zero. If backstresses are specified for the model, these will also be set to zero.

If you include annealing behavior in the material definition and the annealing temperature is defined to be less than the melting temperature specified for the metal plasticity model, the hardening memory will be removed at the annealing temperature and the melting temperature will be used strictly to define the hardening function. Otherwise, the hardening memory will be removed automatically at the melting temperature. If the temperature of the material point falls below the annealing temperature at a subsequent point in time, the material point can work harden again. For more details, see *Annealing or Melting*.

You provide the values of A, B, n, m, θ_{melt}, and $\theta_{\text{transition}}$ as part of the metal plasticity material definition.

Input File Usage: **PLASTIC*, HARDENING=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Plastic**: **Hardening: Johnson-Cook**

Johnson-Cook Strain Rate Dependence

Johnson-Cook strain rate dependence assumes that

$$\bar{\sigma} = \sigma^0\left(\bar{\varepsilon}^{pl}, \theta\right) R\left(\dot{\bar{\varepsilon}}^{pl}\right)$$

and

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}\left(R - 1\right)\right] \quad \text{for} \quad \bar{\sigma} \ge \sigma^0,$$

where

$\overline{\sigma}$

is the yield stress at nonzero strain rate;

$\dot{\overline{\varepsilon}}^{pl}$

is the equivalent plastic strain rate;

$\dot{\varepsilon}_0$ and C

are material parameters measured at or below the transition temperature, $\theta_{\text{transition}}$;

$\sigma^0\left(\overline{\varepsilon}^{pl}, \theta\right)$

is the static yield stress; and

$R\left(\dot{\overline{\varepsilon}}^{pl}\right)$

is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R\left(\dot{\varepsilon}_0\right) = 1.0$).

The yield stress is, therefore, expressed as

$$\overline{\sigma} = \left[A + B\left(\overline{\varepsilon}^{pl}\right)^n\right]\left[1 + C\ \ln\left(\frac{\dot{\overline{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right]\left(1 - \hat{\theta}^m\right).$$

You provide the values of C and $\dot{\varepsilon}_0$ when you define Johnson-Cook rate dependence.

The use of Johnson-Cook hardening does not necessarily require the use of Johnson-Cook strain rate dependence.

Input File Usage: Use both of the following options:

**PLASTIC*, HARDENING=JOHNSON COOK
**RATE DEPENDENT*, TYPE=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Plastic**: **Hardening: Johnson-Cook**: **Suboptions**->**Rate Dependent**: **Hardening: Johnson-Cook**

Johnson-Cook Dynamic Failure

Abaqus/Explicit provides a dynamic failure model specifically for the Johnson-Cook plasticity model, which is suitable only for high-strain-rate deformation of metals. This model is referred to as the "Johnson-Cook dynamic failure model." Abaqus/Explicit also offers a more general implementation of the Johnson-Cook failure model as part of the family of damage initiation criteria, which is the recommended technique for modeling progressive damage and failure of materials (see *About Damage and Failure for Ductile Metals*). The Johnson-Cook dynamic failure model is based on the value of the equivalent plastic strain at element integration points; failure is assumed to occur when the damage parameter exceeds 1. The damage parameter, ω, is defined as

$$\omega = \sum\left(\frac{\Delta\overline{\varepsilon}^{pl}}{\overline{\varepsilon}_f^{pl}}\right),$$

where $\Delta\bar{\varepsilon}^{pl}$ is an increment of the equivalent plastic strain, $\bar{\varepsilon}_f^{pl}$ is the strain at failure, and the summation is performed over all increments in the analysis. The strain at failure, $\bar{\varepsilon}_f^{pl}$, is assumed to be dependent on a nondimensional plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}/\dot{\varepsilon}_0$; a dimensionless pressure-deviatoric stress ratio, p/q (where p is the pressure stress and q is the Mises stress); and the nondimensional temperature, $\hat{\theta}$, defined earlier in the Johnson-Cook hardening model. The dependencies are assumed to be separable and are of the form

$$\bar{\varepsilon}_f^{pl} = \left[d_1 + d_2 \exp\left(d_3 \frac{p}{q}\right)\right]\left[1 + d_4 \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right]\left(1 + d_5 \hat{\theta}\right),$$

where d_1–d_5 are failure parameters measured at or below the transition temperature, $\theta_{\text{transition}}$, and $\dot{\varepsilon}_0$ is the reference strain rate. You provide the values of d_1–d_5 when you define the Johnson-Cook dynamic failure model.

This expression for $\bar{\varepsilon}_f^{pl}$ differs from the original formula published by *Johnson and Cook (1985)* in the sign of the parameter d_3. This difference is motivated by the fact that most materials experience an increase in $\bar{\varepsilon}_f^{pl}$ with increasing pressure-deviatoric stress ratio; therefore, d_3 in the above expression will usually take positive values.

When this failure criterion is met, the deviatoric stress components are set to zero and remain zero for the rest of the analysis. Depending on your choice, the pressure stress may also be set to zero for the rest of calculation (if this is the case, you must specify element deletion and the element will be deleted) or it may be required to remain compressive for the rest of the calculation (if this is the case, you must choose not to use element deletion). By default, the elements that meet the failure criterion are deleted.

The Johnson-Cook dynamic failure model is suitable for high-strain-rate deformation of metals; therefore, it is most applicable to truly dynamic situations. For quasi-static problems that require element removal, the progressive damage and failure models (*Progressive Damage and Failure*) or the Gurson metal plasticity model (*Porous Metal Plasticity*) are recommended.

The use of the Johnson-Cook dynamic failure model requires the use of Johnson-Cook hardening but does not necessarily require the use of Johnson-Cook strain rate dependence. However, the rate-dependent term in the Johnson-Cook dynamic failure criterion will be included only if Johnson-Cook strain rate dependence is defined. The Johnson-Cook damage initiation criterion described in *Damage Initiation for Ductile Metals* does not have these limitations.

Input File Usage: Use both of the following options:

**PLASTIC*, HARDENING=JOHNSON COOK
**SHEAR FAILURE*, TYPE=JOHNSON COOK,
ELEMENT DELETION=YES or NO

Abaqus/CAE Usage: Johnson-Cook dynamic failure is not supported in Abaqus/CAE.

Progressive Damage and Failure

The Johnson-Cook plasticity model can be used in conjunction with the progressive damage and failure models discussed in *About Damage and Failure for Ductile Metals*. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and, in Abaqus/Explicit, Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The models offer two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage

models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations. This is a great advantage over the dynamic failure models discussed above.

Input File Usage: Use the following options:

**PLASTIC*, HARDENING=JOHNSON COOK
**DAMAGE INITIATION*
**DAMAGE EVOLUTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*damage initiation type***: specify the damage initiation criterion: **Suboptions->Damage Evolution**: specify the damage evolution parameters

Tensile Failure

In Abaqus/Explicit the tensile failure model can be used in conjunction with the Johnson-Cook plasticity model to define tensile failure of the material. The tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff and offers a number of failure choices including element removal. Similar to the Johnson-Cook dynamic failure model, the Abaqus/Explicit tensile failure model is suitable for high-strain-rate deformation of metals and is most applicable to truly dynamic problems. For more details, see *Dynamic Failure Models*.

Input File Usage: Use both of the following options:

**PLASTIC*, HARDENING=JOHNSON COOK
**TENSILE FAILURE*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Hardening: Johnson-Cook**: **Suboptions->Tensile Failure**

Heat Generation by Plastic Work

Abaqus allows for an adiabatic thermal-stress analysis (*Adiabatic Analysis*), a fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*), or a fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*) to be performed in which heat generated by plastic straining of a material is calculated. This method is typically used in the simulation of bulk metal forming or high-speed manufacturing processes involving large amounts of inelastic strain, where the heating of the material caused by its deformation is an important effect because of temperature dependence of the material properties. Since the Johnson-Cook plasticity model is motivated by high-strain-rate transient dynamic applications, temperature change in this model is generally computed by assuming adiabatic conditions (no heat transfer between elements). Heat is generated in an element by plastic work, and the resulting temperature rise is computed using the specific heat of the material.

This effect is introduced by defining the fraction of the rate of inelastic dissipation that appears as a heat flux per volume.

Input File Usage: Use all of the following options in the same material data block:

**PLASTIC*, HARDENING=JOHNSON COOK
**SPECIFIC HEAT*
**DENSITY*
**INELASTIC HEAT FRACTION*

Abaqus/CAE Usage: Use all of the following options in the same material definition:

Property module: material editor:
Mechanical->Plasticity->Plastic: **Hardening: Johnson-Cook**
Thermal->Specific Heat
General->Density
Thermal->Inelastic Heat Fraction

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state (see *Initial Conditions*). An initial backstress, $\boldsymbol{\alpha}_0$, can also be specified. The backstress $\boldsymbol{\alpha}_0$ represents a constant kinematic shift of the yield surface, which can be useful for modeling the effects of residual stresses without considering them in the equilibrium solution.

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Elements

The Johnson-Cook plasticity model can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the Johnson-Cook plasticity model:

PEEQ
Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

STATUS
Status of element. The status of an element is 1.0 if the element is active and 0.0 if the element is not.

STATUSMP
Status of each material point in the element (1.0 if a material point is active, 0.0 if it is not). Abaqus/Explicit only.

YIELDS
Yield stress, $\bar{\sigma}$.

References

- Johnson, G. R., and W. H. Cook, "Fracture Characteristics of Three Metals Subjected to Various Strains, Strain rates, Temperatures and Pressures," *Engineering Fracture Mechanics*, vol. 21, no. 1, pp. 31–48, 1985.

Dynamic Failure Models

Products: Abaqus/Explicit Abaqus/CAE

References:

- *Equation of State*
- *Classical Metal Plasticity*
- *Rate-Dependent Yield*
- *Johnson-Cook Plasticity*
- *About the Material Library*
- *Inelastic Behavior*
- *Progressive Damage and Failure*
- **SHEAR FAILURE*
- **TENSILE FAILURE*
- *Defining tensile failure*
- *Defining tensile failure for an equation of state*

Overview

The progressive damage and failure models described in *About Damage and Failure for Ductile Metals* are the recommended method for modeling material damage and failure in Abaqus; these models are suitable for both quasi-static and dynamic situations. Abaqus/Explicit offers two additional element failure models suitable only for high-strain-rate dynamic problems. The shear failure model is driven by plastic yielding. The tensile failure model is driven by tensile loading. These failure models can be used to limit subsequent load-carrying capacity of an element (up to the point of removing the element) once a stress limit is reached. Both models can be used for the same material.

The shear failure model:

- is designed for high-strain-rate deformation of many materials, including most metals;
- uses the equivalent plastic strain as a failure measure;
- offers two choices for what occurs upon failure, including the removal of elements from the mesh;
- can be used in conjunction with either the Mises or the Johnson-Cook plasticity models; and
- can be used in conjunction with the tensile failure model.

The tensile failure model:

- is designed for high-strain-rate deformation of many materials, including most metals;
- uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff;
- offers a number of choices for what occurs upon failure, including the removal of elements from the mesh;
- can be used in conjunction with either the Mises or the Johnson-Cook plasticity models or the equation of state material model; and
- can be used in conjunction with the shear failure model.

Shear Failure Model

The shear failure model can be used in conjunction with the Mises or the Johnson-Cook plasticity models in Abaqus/Explicit to define shear failure of the material.

Shear Failure Criterion

The shear failure model is based on the value of the equivalent plastic strain at element integration points; failure is assumed to occur when the damage parameter exceeds 1. The damage parameter, ω, is defined as

$$\omega = \frac{\bar{\varepsilon}_0^{pl} + \sum \Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_f^{pl}},$$

where $\bar{\varepsilon}_0^{pl}$ is any initial value of the equivalent plastic strain, $\Delta\bar{\varepsilon}^{pl}$ is an increment of the equivalent plastic strain, $\bar{\varepsilon}_f^{pl}$ is the strain at failure, and the summation is performed over all increments in the analysis.

The strain at failure, $\bar{\varepsilon}_f^{pl}$, is assumed to depend on the plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; a dimensionless pressure-deviatoric stress ratio, p/q (where p is the pressure stress and q is the Mises stress); temperature; and predefined field variables. There are two ways to define the strain at failure, $\bar{\varepsilon}_f^{pl}$. One is to use direct tabular data, where the dependencies are given in a tabular form. Alternatively, the analytical form proposed by Johnson and Cook can be invoked (see *Johnson-Cook Plasticity* for more details).

When direct tabular data are used to define the shear failure model, the strain at failure, $\bar{\varepsilon}_f^{pl}$, must be given as a tabular function of the equivalent plastic strain rate, the pressure-deviatoric stress ratio, temperature, and predefined field variables. This method requires the use of the Mises plasticity model.

For the Johnson-Cook shear failure model, you must specify the failure parameters, d_1–d_5 (see *Johnson-Cook Plasticity* for more details on these parameters). The shear failure data must be calibrated at or below the transition temperature, $\theta_{\text{transition}}$, defined in *Johnson-Cook Plasticity*. This method requires the use of the Johnson-Cook plasticity model.

Input File Usage: Use both of the following options for the Mises plasticity model:

**PLASTIC*, HARDENING=ISOTROPIC
**SHEAR FAILURE*, TYPE=TABULAR

Use both of the following options for the Johnson-Cook plasticity model:

**PLASTIC*, HARDENING=JOHNSON COOK
**SHEAR FAILURE*, TYPE=JOHNSON COOK

Abaqus/CAE Usage: The shear failure model is not supported in Abaqus/CAE.

Element Removal

When the shear failure criterion is met at an integration point, all the stress components are set to zero and that material point fails. An element is deleted (or removed) from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a

material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively. Element deletion is the default failure choice.

An alternative failure choice, where the element is not deleted, is to specify that when the shear failure criterion is met at a material point, the deviatoric stress components are set to zero for that point and remain zero for the rest of the calculation. The pressure stress is then required to remain compressive; that is, if a negative pressure stress is computed in a failed material point in an increment, it is reset to zero. This failure choice is not allowed when using plane stress, shell, membrane, beam, pipe, and truss elements because the structural constraints may be violated.

Input File Usage: Use the following option to allow element deletion when the failure criterion is met (the default):

**SHEAR FAILURE*, ELEMENT DELETION=YES

Use the following option to allow the element to take hydrostatic compressive stress only when the failure criterion is met:

**SHEAR FAILURE*, ELEMENT DELETION=NO

Abaqus/CAE Usage: The shear failure model is not supported in Abaqus/CAE.

Determining When to Use the Shear Failure Model

The shear failure model in Abaqus/Explicit is suitable for high-strain-rate dynamic problems where inertia is important. Improper use of the shear failure model may result in an incorrect simulation.

For quasi-static problems that may require element removal, the progressive damage and failure models (*Progressive Damage and Failure*) or the Gurson porous metal plasticity model (*Porous Metal Plasticity*) are recommended.

Tensile Failure Model

The tensile failure model can be used in conjunction with either the Mises or the Johnson-Cook plasticity models or the equation of state material model in Abaqus/Explicit to define tensile failure of the material.

Tensile Failure Criterion

The Abaqus/Explicit tensile failure model uses the hydrostatic pressure stress as a failure measure to model dynamic spall or a pressure cutoff. The tensile failure criterion assumes that failure occurs when the pressure stress, p, becomes more tensile than the user-specified hydrostatic cutoff stress, σ_{cutoff}. The hydrostatic cutoff stress may be a function of temperature and predefined field variables. There is no default value for this stress.

The tensile failure model can be used with either the Mises or the Johnson-Cook plasticity models or the equation of state material model.

Input File Usage: Use both of the following options for the Mises or Johnson-Cook plasticity models:

**PLASTIC*
**TENSILE FAILURE*

Use both of the following options for the equation of state material model:

**EOS*
**TENSILE FAILURE*

Abaqus/CAE Usage: Use the following options for the Mises or Johnson-Cook plasticity models:

Property module: material editor: **Mechanical->Plasticity->Plastic: Hardening: Isotopic** or **Johnson-Cook**: **Suboptions->Tensile Failure**

Use the following options for the equation of state material model:

Property module: material editor: **Mechanical->Eos**: **Type**: **JWL**, **Us-Up**, or **Tabular**: **Suboptions->Tensile Failure**

Failure Choices

When the tensile failure criterion is met at an element integration point, the material point fails. Five failure choices are offered for the failed material points: the default choice, which includes element removal, and four different spall models. These failure choices are described below.

Element Removal

When the tensile failure criterion is met at an integration point, all the stress components are set to zero and that material point fails. By default, an element is deleted (or removed) from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Input File Usage: **TENSILE FAILURE*, ELEMENT DELETION=YES (default)

Abaqus/CAE Usage: Property module: material editor: **Suboptions->Tensile Failure**: **Allow element deletion when the failure criterion is met**

Spall Models

An alternative failure choice that is based on spall (the crumbling of a material), rather than element removal, is also available. Four failure combinations are available in this category. When the tensile failure criterion is met at a material point, the deviatoric stress components may be unaffected or may be required to be zero, and the pressure stress may be limited by the hydrostatic cutoff stress or may be required to be compressive. Therefore, there are four possible failure combinations (see *Figure 1*, where "O" is the stress that would exist if the tensile failure model were not used).

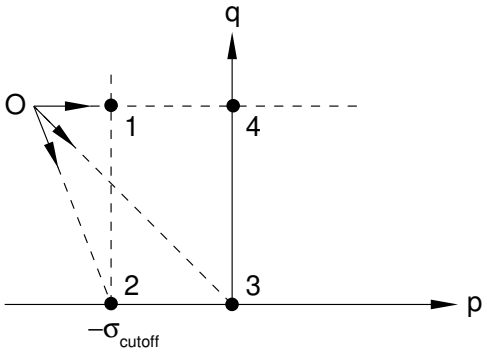


Figure 1: Tensile failure choices.

These failure combinations are as follows:

- Ductile shear and ductile pressure: this choice corresponds to point 1 in *Figure 1* and models the case in which the deviatoric stress components are unaffected and the pressure stress is limited by the hydrostatic cutoff stress; that is, $p = max\left(-\sigma_{\text{cutoff}}, p\right)$.
- Brittle shear and ductile pressure: this choice corresponds to point 2 in *Figure 1* and models the case in which the deviatoric stress components are set to zero and remain zero for the rest of the calculation, and the pressure stress is limited by the hydrostatic cutoff stress; that is, $p = max\left(-\sigma_{\text{cutoff}}, p\right)$.
- Brittle shear and brittle pressure: this choice corresponds to point 3 in *Figure 1* and models the case in which the deviatoric stress components are set to zero and remain zero for the rest of the calculation, and the pressure stress is required to be compressive; that is, $p = max\left(0, p\right)$.
- Ductile shear and brittle pressure: this choice corresponds to point 4 in *Figure 1* and models the case in which the deviatoric stress components are unaffected and the pressure stress is required to be compressive; that is, $p = max\left(0, p\right)$.

There is no default failure combination for the spall models. If you choose not to use the element deletion model, you must specify the failure combination explicitly. If the material's deviatoric behavior is not defined (for example, the equation of state model without deviatoric behavior is used), the deviatoric part of the combination is meaningless and will be ignored. The spall models are not allowed when using plane stress, shell, membrane, beam, pipe, and truss elements.

Input File Usage: Use the following options to model ductile shear and ductile pressure:

**TENSILE FAILURE*, ELEMENT DELETION=NO,
SHEAR=DUCTILE, PRESSURE=DUCTILE

Use the following options to model brittle shear and ductile pressure:

**TENSILE FAILURE*, ELEMENT DELETION=NO,
SHEAR=BRITTLE, PRESSURE=DUCTILE

Use the following options to model brittle shear and brittle pressure:

**TENSILE FAILURE*, ELEMENT DELETION=NO,
SHEAR=BRITTLE, PRESSURE=BRITTLE

Use the following options to model ductile shear and brittle pressure:

**TENSILE FAILURE*, ELEMENT DELETION=NO,
SHEAR=DUCTILE, PRESSURE=BRITTLE

Abaqus/CAE Usage: Use the following options to model the failure combinations:

Property module: material editor: **Suboptions->Tensile Failure**: **Pressure**: **BRITTLE** or **DUCTILE**, **Shear**: **BRITTLE** or **DUCTILE**

Determining When to Use the Tensile Failure Model

The tensile failure model in Abaqus/Explicit is suitable for high-strain-rate dynamic problems in which inertia effects are important. Improper use of the tensile failure model may result in an incorrect simulation.

Using the Failure Models with Rebar

It is possible to use the shear failure and/or the tensile failure models in elements for which rebars are also defined. When such elements fail according to the failure criterion, the base material contribution to the element stress-carrying capacity is removed or adjusted depending on the type of failure chosen, but the rebar contribution to the element stress-carrying capacity is not removed. However, if you also include failure in the rebar material definition, the rebar contribution to the element stress-carrying capacity will also be removed or adjusted if the failure criterion specified for the rebar is met.

Elements

The shear and tensile failure models with element deletion can be used with any elements in Abaqus/Explicit that include mechanical behavior (elements that have displacement degrees of freedom). The shear and tensile failure models without element deletion can be used only with plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus/Explicit.

Output

In addition to the standard output identifiers available in Abaqus/Explicit (*Abaqus/Explicit Output Variable Identifiers*), the following variable has special meaning for the shear and tensile failure models:

STATUS	Status of element (1.0 if the element is active, 0.0 if it is not).
STATUSMP	Status of each material point in the element (1.0 if a material point is active, 0.0 if it is not).

Porous Metal Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **POROUS METAL PLASTICITY*
- **POROUS FAILURE CRITERIA*
- **VOID NUCLEATION*
- *Defining porous metal plasticity*

Overview

The porous metal plasticity model:

- is used to model materials with a dilute concentration of voids in which the relative density is greater than 0.9;
- is based on Gurson's porous metal plasticity theory (Gurson, 1977) with void nucleation and, in Abaqus/Explicit, a failure definition; and
- defines the inelastic flow of the porous metal on the basis of a potential function that characterizes the porosity in terms of a single state variable, the relative density.

Elastic and Plastic Behavior

You specify the elastic part of the response separately; only linear isotropic elasticity can be specified (see *Linear Elastic Behavior*). The porous metal plasticity model cannot be used in conjunction with porous elasticity (*Elastic Behavior of Porous Materials*).

You specify the hardening behavior of the fully dense matrix material by defining a metal plasticity model (see *Classical Metal Plasticity*). Only isotropic hardening can be specified. The hardening curve must describe the yield stress of the matrix material as a function of plastic strain in the matrix material. In defining this dependence at finite strains, “true” (Cauchy) stress and log strain values should be given. Rate dependency effects for the matrix material can be modeled (see *Rate-Dependent Yield*).

Yield Condition

The relative density of a material, r, is defined as the ratio of the volume of solid material to the total volume of the material. The relationships defining the model are expressed in terms of the void volume fraction, f, which is defined as the ratio of the volume of voids to the total volume of the material. It follows that $f = 1 - r$. For a metal containing a dilute concentration of voids, Gurson (1977) proposed a yield condition as a function of the void volume fraction. This yield condition was later modified by Tvergaard (1981) to the form

$$\Phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f \cosh\left(-q_2 \frac{3p}{2\sigma_y}\right) - \left(1 + q_3 f^2\right) = 0,$$

where

$\mathbf{S} = p\mathbf{I} + \sigma$

is the deviatoric part of the Cauchy stress tensor $\boldsymbol{\sigma}$;

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$

is the effective Mises stress;

$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$

is the hydrostatic pressure;

$\sigma_y\left(\bar{\varepsilon}_m^{pl}\right)$

is the yield stress of the fully dense matrix material as a function of $\bar{\varepsilon}_m^{pl}$, the equivalent plastic strain in the matrix; and

q_1, q_2, q_3

are material parameters.

The Cauchy stress is defined as the force per "current unit area," comprised of voids and the solid (matrix) material.

$f = 0$ ($r = 1$) implies that the material is fully dense, and the Gurson yield condition reduces to the Mises yield condition. $f = 1$ ($r = 0$) implies that the material is completely voided and has no stress carrying capacity. The model generally gives physically reasonable results only for $f < 0.1$ ($r > 0.9$).

The model is described in detail in *Porous metal plasticity*, along with a discussion of its numerical implementation.

If the porous metal plasticity model is used during a pore pressure analysis (see *Coupled Pore Fluid Diffusion and Stress Analysis*), the relative density, r, is tracked independently of the void ratio.

Specifying q_1, q_2, and q_3

You specify the parameters q_1, q_2, and q_3 directly for the porous metal plasticity model. For typical metals the ranges of the parameters reported in the literature are q_1 = 1.0 to 1.5, q_2 = 1.0, and $q_3 = q_1^2$ = 1.0 to 2.25 (see *Necking of a round tensile bar*). The original Gurson model is recovered when $q_1 = q_2 = q_3 = 1.0$. You can define these parameters as tabular functions of temperature and/or field variables.

Input File Usage: **POROUS METAL PLASTICITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Porous Metal Plasticity**

Failure Criteria in Abaqus/Explicit

The porous metal plasticity model in Abaqus/Explicit allows for failure. In this case the yield condition is written as

$$\Phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f^* \cosh\left(-q_2 \frac{3p}{2\sigma_y}\right) - \left(1 + q_3 f^{*2}\right) = 0,$$

where the function $f^*(f)$ models the rapid loss of stress carrying capacity that accompanies void coalescence. This function is defined in terms of the void volume fraction:

$$f^* = \begin{cases} f & \text{if } f \le f_c, \\ f_c + \dfrac{\bar{f}_F - f_c}{f_F - f_c}(f - f_c) & \text{if } f_c < f < f_F, \\ \bar{f}_F & \text{if } f \ge f_F, \end{cases}$$

where

$$\bar{f}_F = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3}.$$

In the above relationship f_c is a critical value of the void volume fraction, and f_F is the value of void volume fraction at which there is a complete loss of stress carrying capacity in the material. The user-specified parameters f_c and f_F model the material failure when $f_c < f < f_F$, due to mechanisms such as micro fracture and void coalescence. When $f \ge f_F$, total failure at the material point occurs. An element is deleted (or removed) from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Input File Usage: Use the following option in conjunction with the **POROUS METAL PLASTICITY* option:

**POROUS FAILURE CRITERIA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Porous Metal Plasticity**: **Suboptions->Porous Failure Criteria**

Specifying the Initial Relative Density

You can specify the initial relative density of the porous material, r_0, at material points or at nodes. If you do not specify the initial relative density, Abaqus will assign it a value of 1.0.

At Material Points

You can specify the initial relative density as part of the porous metal plasticity material definition.

Input File Usage: **POROUS METAL PLASTICITY*, RELATIVE DENSITY=r_0

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Porous Metal Plasticity**: **Relative density: r_0**

At Nodes

Alternatively, you can specify the initial relative density at nodes as initial conditions (*Initial Conditions*); these values are interpolated to the material points. The initial conditions are applied only if the relative density is not specified as part of the porous metal plasticity material definition. When a discontinuity of the initial relative

density field occurs at the element boundaries, separate nodes must be used to define the elements at these boundaries, with multi-point constraints applied to make the nodal displacements and rotations equivalent.

Input File Usage: **INITIAL CONDITIONS*, TYPE=RELATIVE DENSITY

Abaqus/CAE Usage: Initial relative density is not supported in Abaqus/CAE.

Flow Rule and Hardening

The presence of pressure in the yield condition results in nondeviatoric plastic strains. Plastic flow is assumed to be normal to the yield surface:

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\lambda}\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}.$$

The hardening of the fully dense matrix material is described through $\sigma_y = \sigma_y\left(\bar{\varepsilon}_m^{pl}\right)$. The evolution of the equivalent plastic strain in the matrix material is obtained from the following equivalent plastic work expression:

$$(1-f)\,\sigma_y\dot{\bar{\varepsilon}}_m^{pl} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}.$$

The model is illustrated in *Figure 1*, where the yield surfaces for different levels of void volume fraction are shown in the p–q plane.

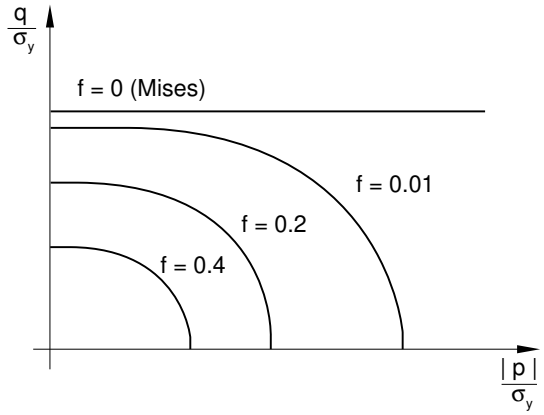


Figure 1: Schematic of the yield surface in the p–q plane.

Figure 2 compares the behavior of a porous material (whose initial yield stress is σ_{y_0}) in tension and compression against the behavior of the perfectly plastic matrix material. In compression the porous material "hardens" due to closing of the voids, and in tension it "softens" due to growth and nucleation of the voids.

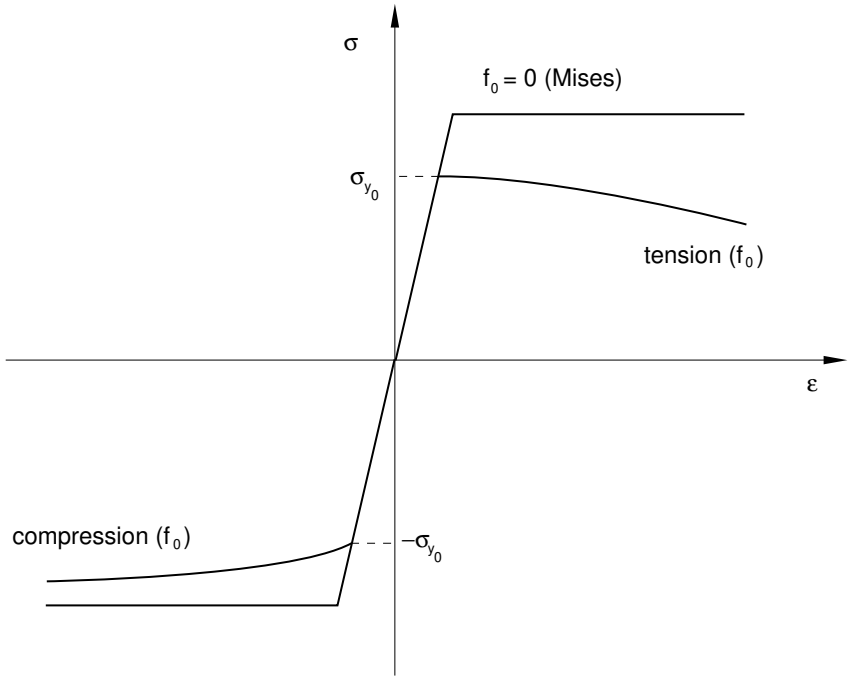


Figure 2: Schematic of uniaxial behavior of a porous metal (perfectly plastic matrix material with initial volume fraction of voids = f_0).

Void Growth and Nucleation

The total change in void volume fraction is given as

$$\dot{f} = \dot{f}_{\text{gr}} + \dot{f}_{\text{nucl}},$$

where $\dot{f}_{\text{gr}}$ is change due to growth of existing voids and $\dot{f}_{\text{nucl}}$ is change due to nucleation of new voids. Growth of the existing voids is based on the law of conservation of mass and is expressed in terms of the void volume fraction:

$$\dot{f}_{\text{gr}} = (1 - f)\, \dot{\boldsymbol{\varepsilon}}^{pl} : \mathbf{I}.$$

The nucleation of voids is given by a strain-controlled relationship:

$$\dot{f}_{\text{nucl}} = A \dot{\bar{\varepsilon}}^{pl}_{m},$$

where

$$A = \frac{f_N}{s_N \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}^{pl}_{m} - \varepsilon_N}{s_N}\right)^2\right].$$

The normal distribution of the nucleation strain has a mean value ε_N and standard deviation s_N. f_N is the volume fraction of the nucleated voids, and voids are nucleated only in hydrostatic tension.

The nucleation function A/f_N is assumed to have a normal distribution, as shown in *Figure 3* for different values of the standard deviation s_N.

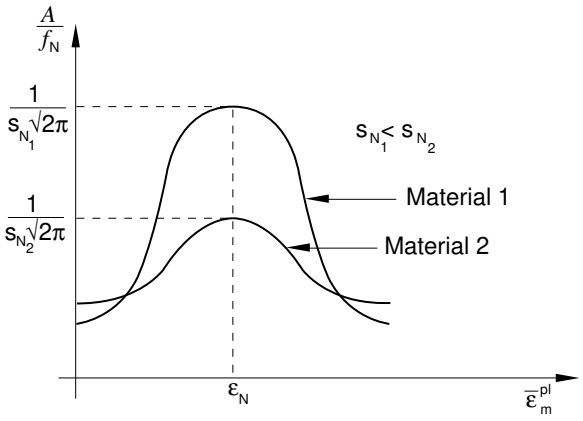


Figure 3: Nucleation function A/f_N.

Figure 4 shows the extent of softening in a uniaxial tension test of a porous material for different values of f_N.

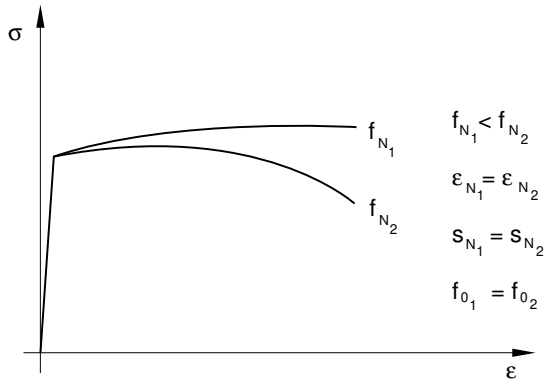


Figure 4: Softening (in uniaxial tension) as a function of f_N.

The following ranges of values are reported in the literature for typical metals: ε_N = 0.1 to 0.3, s_N = 0.05 to 0.1, and f_N = 0.04 (see *Necking of a round tensile bar*). You specify these parameters, which can be defined as tabular functions of temperature and predefined field variables. Abaqus will include void nucleation in a tensile field only when you include it in the material definition.

In Abaqus/Standard the accuracy of the implicit integration of the void nucleation and growth equation is controlled by prescribing the maximum allowable time increment in the automatic time incrementation scheme.

Input File Usage: **VOID NUCLEATION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Porous Metal Plasticity**: **Suboptions->Void Nucleation**

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, Abaqus allows you to prescribe initial conditions directly for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ (*Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Defining Initial Hardening Conditions in a User Subroutine

For more complicated cases, initial conditions can be defined in Abaqus/Standard through user subroutine *HARDINI*.

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING, USER

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The porous metal plasticity model can be used with any stress/displacement elements other than one-dimensional elements (beam, pipe, and truss elements) or elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning in the porous metal plasticity model:

PEEQ
Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{pl}}{(1-f)\sigma_y}$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*).

RD
Relative density, r.

VVF
Void volume fraction, f.

VVFG
Void volume fraction due to void growth.

VVFN
Void volume fraction due to void nucleation.

References

- Gurson, A. L., "Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Materials," *Journal of Engineering Materials and Technology*, vol. 99, pp. 2–15, 1977.
- Tvergaard, V., "Influence of Voids on Shear Band Instabilities under Plane Strain Condition," *International Journal of Fracture Mechanics*, vol. 17, pp. 389–407, 1981.

Cast Iron Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Combining Material Behaviors*
- *Inelastic Behavior*
- **CAST IRON COMPRESSION HARDENING*
- **CAST IRON PLASTICITY*
- **CAST IRON TENSION HARDENING*
- *Defining cast iron plasticity*

Overview

The cast iron plasticity model:

- is intended for the constitutive modeling of gray cast iron;
- provides elastic-plastic behavior with different yield strengths, flow, and hardening in tension and compression;
- is based on a yield function that depends on the maximum principal stress under tensile loading conditions and pressure-independent (von Mises type) behavior under compressive loading conditions;
- allows for simultaneous inelastic dilatation and inelastic shearing under tensile loading conditions;
- allows only inelastic shearing under compressive loading conditions;
- is intended for the simulation of material response only under essentially monotonic loading conditions; and
- cannot be used to model rate dependence.

Elastic and Plastic Behavior

The cast iron plasticity model describes the mechanical behavior of gray cast iron, a material with a microstructure consisting of a distribution of graphite flakes in a steel matrix. In tension the graphite flakes act as stress concentrators, resulting in yielding as a function of the maximum principal stress, followed by brittle behavior. In compression the graphite flakes do not have an appreciable effect on the macroscopic response, resulting in a ductile behavior similar to that of many steels.

You specify the elastic part of the response separately; only linear isotropic elasticity can be used (see *Linear Elastic Behavior*). The elastic stiffness is assumed to be the same under tension and compression.

The cast iron plasticity model is used to provide the value of the plastic “Poisson's ratio,” which is the absolute value of the ratio of the transverse to the longitudinal plastic strain under uniaxial tension. The plastic Poisson's ratio can vary with the plastic deformation. However, the model in Abaqus assumes that it is constant with respect to plastic deformation. It can depend on temperature and field variables. If no value is specified for the plastic Poisson's ratio, a default value of 0.04 is assumed. This default value is based on experimental results for permanent volumetric strain under uniaxial tension (see *Cast iron plasticity* for details).

Independent hardening (see *Figure 1*) of the material under tension and compression can be specified as described below. The tension hardening data provide the uniaxial tension yield stress as a function of plastic strain,

temperature, and field variables under uniaxial tension. The compression hardening data provide the uniaxial compression yield stress as a function of plastic strain, temperature, and field variables under uniaxial compression.

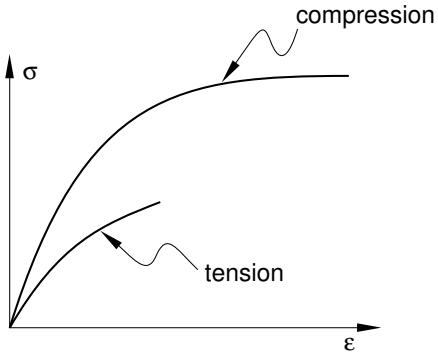


Figure 1: Typical stress-strain response of gray cast iron under uniaxial tension and uniaxial compression.

Input File Usage: **CAST IRON PLASTICITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cast Iron Plasticity**

Yield Condition

Abaqus makes use of a composite yield surface to describe the different behavior in tension and compression. In tension yielding is assumed to be governed by the maximum principal stress, while in compression yielding is assumed to be pressure independent and governed by the deviatoric stresses alone (Mises yield condition).

The model is described in detail in *Cast iron plasticity*.

Flow Rule

For the purposes of discussing the flow and hardening behavior, it is useful to divide the meridional plane into the two regions shown in *Figure 2*.

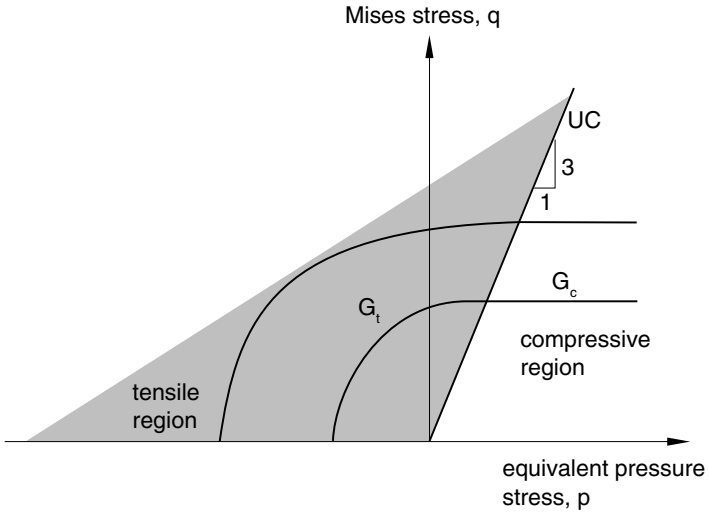


Figure 2: Schematic of the flow potentials in the *p–q* plane.

The region to the left of the uniaxial compression line (labeled UC) is referred to as the "tensile region," while the region to the right of the uniaxial compression line is referred to as the "compressive region." The flow potential consists of the Mises cylinder in the compressive region and an ellipsoidal "cap" in the tensile region. The transition between the two surfaces is smooth. The projection of the flow potential on the meridional plane (see *Figure 2*) consists of a straight line in the compressive region and an ellipse in the tensile region. The corresponding projection on the deviatoric plane is a circle. A consequence of the above choice is that plastic flow results in inelastic volume expansion in the tensile region and no inelastic volume change in the compressive region (see *Cast iron plasticity* for details).

Nonassociated Flow

Since the flow potential is different from the yield surface ("nonassociated" flow), the material Jacobian matrix is unsymmetric. Hence, to improve convergence, use the unsymmetric matrix storage and solution scheme (see *Defining an Analysis*).

Hardening

Since the hardening of gray cast iron is different in uniaxial tension and uniaxial compression, you need to provide two sets of hardening data in tabular form: one based on a uniaxial tension experiment that defines $\sigma_t = \sigma_t\left(\bar{\varepsilon}_t^{pl}, \theta, f^{\alpha}\right)$ and the other based on a uniaxial compression experiment that defines $\sigma_c = \sigma_c\left(\bar{\varepsilon}_c^{pl}, \theta, f^{\alpha}\right)$. Here, $\bar{\varepsilon}_t^{pl}$ and $\bar{\varepsilon}_c^{pl}$ are the equivalent plastic strains in uniaxial tension and uniaxial compression, respectively.

Input File Usage: Use both of the following options in conjunction with the **CAST IRON PLASTICITY* option:

**CAST IRON COMPRESSION HARDENING*
**CAST IRON TENSION HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cast Iron Plasticity**: **Compression Hardening** and **Tension Hardening**

Restrictions on Material Data

The plastic Poisson's ratio, ν_{pl}, is expected to be less than 0.5 since experimental results suggest that there is a permanent increase in the volume of gray cast iron when it is loaded in uniaxial tension beyond yield. For the potential to be well-defined, ν_{pl} must be greater than -1.0. Thus, the plastic Poisson's ratio must satisfy $-1.0 < \nu_{pl} \leq 0.5$.

The cast iron plasticity material model is intended for modeling cast iron and other materials like cast iron for which the behavior in uniaxial tension and uniaxial compression matches the behavior shown in *Figure 1*. In particular, the model expects the initial yield stress in uniaxial tension to be less than the initial yield stress in uniaxial compression. Even if the overall stress-strain response and hardening behavior in uniaxial stress states of some material other than cast iron is consistent with that of cast iron, you must also ensure that the flow potential (which has been constructed specifically for modeling cast iron) for the model is meaningful for other materials. Abaqus issues a warning message only if the initial yield stress in uniaxial tension is equal to or greater than that in uniaxial compression. No other checks are carried out in this regard.

If the yield stress in uniaxial tension is higher than that in uniaxial compression, a material point in uniaxial tension may actually yield at the initial yield stress specified for uniaxial compression. This apparent anomalous behavior is due to the fact that (as a result of unrealistic user-specified material properties) a uniaxial tension stressing path in stress space meets the compressive (Mises) part of the yield surface first.

Elements

The cast iron plasticity model can be used with any stress/displacement element in Abaqus other than elements for which the assumed stress state is plane stress (plane stress continuum, shell, and membrane elements). It can be used with one-dimensional elements (trusses and beams in a plane) and, in Abaqus/Standard, with beams in space.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the cast iron plasticity material model:

PEEQ

Equivalent plastic strain in uniaxial compression, $\bar{\varepsilon}_c^{pl} = \int_0^t \dot{\bar{\varepsilon}}_c^{pl} dt$.

PEEQT

Equivalent plastic strain in uniaxial tension, $\bar{\varepsilon}_t^{pl} = \int_0^t \dot{\bar{\varepsilon}}_t^{pl} dt$.

Two-Layer Viscoplasticity

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Combining Material Behaviors*
- *Inelastic Behavior*
- **ELASTIC*
- **PLASTIC*
- **VISCOUS*
- *Defining the viscous component of a two-layer viscoplasticity model*

Overview

The two-layer viscoplastic model:

- is intended for modeling materials in which significant time-dependent behavior as well as plasticity is observed, which for metals typically occurs at elevated temperatures;
- consists of an elastic-plastic network that is in parallel with an elastic-viscous network (in contrast to the coupled creep and plasticity capabilities in which the plastic and the viscous networks are in series);
- is based on a Mises or Hill yield condition in the elastic-plastic network and any of the available creep models in Abaqus/Standard (except the hyperbolic creep law) in the elastic-viscous network;
- assumes a deviatoric inelastic response (hence, the pressure-dependent plasticity or creep models cannot be used to define the behavior of the two networks);
- is intended for modeling material response under fluctuating loads over a wide range of temperatures; and
- has been shown to provide good results for thermomechanical loading.

Material Behavior

The material behavior is broken down into three parts: elastic, plastic, and viscous. *Figure 1* shows a one-dimensional idealization of this material model, with the elastic-plastic and the elastic-viscous networks in parallel. The following subsections describe the elastic and the inelastic (plastic and viscous) behavior in detail.

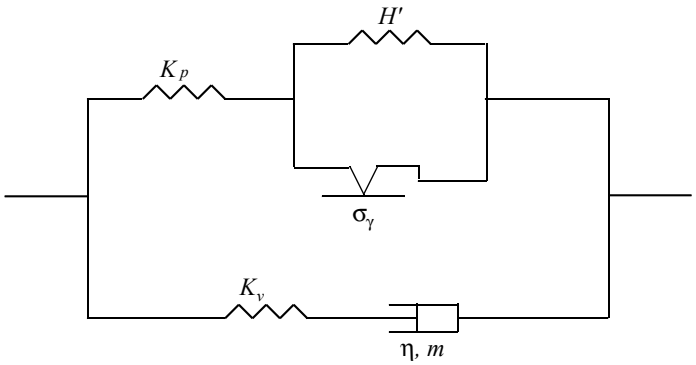


Figure 1: One-dimensional idealization of the two-layer viscoplasticity model.

Elastic Behavior

The elastic part of the response for both networks is specified using a linear isotropic elasticity definition. Any one of the available elasticity models in Abaqus/Standard can be used to define the elastic behavior of the networks. Referring to the one-dimensional idealization (*Figure 1*), the ratio of the elastic modulus of the elastic-viscous network (K_V) to the total (instantaneous) modulus ($K_P + K_V$) is given by

$$f = \frac{K_V}{(K_P+K_V)}.$$

The user-specified ratio *f*, given as part of the viscous behavior definition as discussed later, apportions the total moduli specified for the elastic behavior among the elastic-viscous and the elastic-plastic networks. As a result, if isotropic elastic properties are defined, the Poisson's ratios are the same in both networks. However, if anisotropic elasticity is defined, the same type of anisotropy holds for both networks. The properties specified for the elastic behavior are assumed to be the instantaneous properties ($K_P + K_V$).

Input File Usage: **ELASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**

Plastic Behavior

A plasticity definition can be used to provide the static hardening data for the material model. All available metal plasticity models, including Hill's plasticity model to define anisotropic yield (*Hill Anisotropic Yield/Creep*), can be used.

The elastic-plastic network does not take into account rate-dependent yield. Hence, any specification of strain rate dependence for the plasticity model is not allowed.

Input File Usage: Use the following options:

**PLASTIC*
**POTENTIAL*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Potential**

Viscous Behavior

The viscous behavior of the material can be governed by any of the available creep laws in Abaqus/Standard (*Rate-Dependent Plasticity: Creep and Swelling*), except the hyperbolic creep law. When you define the viscous behavior, you specify the viscosity parameters and choose the specific type of viscous behavior. If you choose to input the creep law through user subroutine *CREEP*, only deviatoric creep should be defined—more specifically, volumetric swelling behavior should not be defined within user subroutine *CREEP*. In addition, you also specify the fraction, *f*, that defines the ratio of the elastic modulus of the elastic-viscous network to the total (instantaneous) modulus. Viscous stress ratios can be specified under the viscous behavior definition to define anisotropic viscosity (see *Hill Anisotropic Yield/Creep*).

All material properties can be specified as functions of temperature and predefined field variables.

Input File Usage: Use the following options:

*VISCOUS, LAW=*creep law*
POTENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Viscous**: **Law: Time**, **Strain**, **User**, **Anand**, **Darveaux**, **Double Power**, **Power**, or **Time Power**

Property module: material editor: **Mechanical->Plasticity->Viscous**: **Suboptions->Potential**

Time-Dependent Behavior

In the "time hardening" power law model the total time or the creep time can be used. The total time is the accumulated time over all general analysis steps. The creep time is the sum of the times of the procedures with time-dependent material behavior. If the total time is used, it is recommended that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage: Use one of the following options:

*VISCOUS, TIME=TOTAL (default)
*VISCOUS, TIME=CREEP

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Viscous**: **Time: Total** or **Creep**

Thermal Expansion

Thermal expansion can be modeled by providing the thermal expansion coefficient of the material (*Thermal Expansion*). Anisotropic expansion can be defined in the usual manner. In the one-dimensional idealization the expansion element is assumed to be in series with the rest of the network.

Calibration of Material Parameters

The calibration procedure is best explained in the context of the one-dimensional idealization of the material model. In the following discussion the viscous behavior is assumed to be governed by the Norton-Hoff rate law, which is given by

$$\dot{\varepsilon}_V^v = A\sigma_V^n.$$

In the expression above the subscript V denotes quantities in the elastic-viscous network alone. This form of the rate law may be chosen, for example, by choosing a time-hardening power law for the viscous behavior and setting $m = 0$. For this basic case there are six material parameters that need to be calibrated (*Figure 1*). These are the elastic properties of the two networks, K_P and K_V; the initial yield stress σ_y; the hardening H'; and the Norton-Hoff rate parameters, A and n.

The experiment that needs to be performed is uniaxial tension under different constant strain rates. A static (effectively zero strain rate) uniaxial tension test determines the long-term modulus, K_P; the initial yield stress, σ_y; and the hardening, H'. The hardening is assumed to be linear for illustration purposes. The material model is not limited to linear hardening, and any general hardening behavior can be defined for the plasticity model.

The instantaneous elastic modulus, $K = K_P + K_V$, can be measured by measuring the initial elastic response of the material under nonzero, relatively high, strain rates. Several such measurements at different applied strain

rates can be compared until the instantaneous moduli does not change with a change in the applied strain rate. The difference between K and K_P determines K_V.

To calibrate the parameters A and n, it is useful to recognize that the long-term (steady-state) behavior of the elastic-viscous network under a constant applied strain rate, $\dot{\varepsilon}_o$, is a constant stress of magnitude $\sigma_V = A^{-\frac{1}{n}} \dot{\varepsilon}_o{}^{\frac{1}{n}}$. Under the assumption that the hardening modulus is negligible compared to the elastic modulus ($K_P \gg H'$), the steady-state response of the overall material is given by

$$\sigma = A^{-\frac{1}{n}} \dot{\varepsilon}_o{}^{\frac{1}{n}} + \sigma_y + H' \varepsilon,$$

where σ is the total stress for a given total strain ε. To determine whether steady state has been reached, one can plot the quantity $\overline{\sigma} = \sigma - \sigma_y - H' \varepsilon$ as a function of ε and note when it becomes a constant. The constant value of $\overline{\sigma}$ is equal to $A^{-\frac{1}{n}} \dot{\varepsilon}_o{}^{\frac{1}{n}}$. By performing several tests at different values of the constant applied strain rate $\dot{\varepsilon}_o$, it is possible to determine the constants A and n.

Material Response in Different Analysis Steps

The material is active during all stress/displacement procedure types. In a static analysis step where the long-term response is requested (see *Static Stress Analysis*), only the elastic-plastic network will be active; the elastic-viscous network will not contribute in any manner. In particular, the stress in the viscous network will be zero during a long-term static response. If the creep effects are removed in a coupled temperature-displacement procedure or a soils consolidation procedure, the response of the elastic-viscous network will be assumed to be elastic only. During a linear perturbation step, only the elastic response of the networks is considered.

Some stress/displacement procedure types (coupled temperature-displacement, soils consolidation, and quasi-static) allow user control of the time integration accuracy of the viscous constitutive equations through a user-specified error tolerance. In other procedure types where no such direct control is currently available (static, dynamic), you must choose appropriate time increments. These time increments must be small compared to the typical relaxation time of the material.

Elements

The two-layer viscoplastic model is not available for one-dimensional elements (beams and trusses). It can be used with any other element in Abaqus/Standard that includes mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning for the two-layer viscoplastic material model:

EE
The elastic strain is defined as: $\varepsilon^{el} = f\varepsilon_V^{el} + (1-f)\,\varepsilon_P^{el}$.

IE
The all inelastic strain is defined as: $\varepsilon^{inel} = f\varepsilon_V^{v} + (1-f)\,\varepsilon_P^{pl}$.

PE
Plastic strain, ε_P^{pl}, in the elastic-plastic network.

VE Viscous strain, $\boldsymbol{\varepsilon}_V^v$, in the elastic-viscous network.

PS Stress, $\boldsymbol{\sigma}_P$, in the elastic-plastic network.

VS Stress, $\boldsymbol{\sigma}_V$, in the elastic-viscous network.

PEEQ The equivalent plastic strain, defined as $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_P^{pl} : \dot{\boldsymbol{\varepsilon}}_P^{pl}}\, dt$.

VEEQ The equivalent viscous strain, defined as $\int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_V^{v} : \dot{\boldsymbol{\varepsilon}}_V^{v}}\, dt$.

SENER The elastic strain energy density per unit volume, defined as $\frac{1}{2}\boldsymbol{\sigma}_P : \boldsymbol{\varepsilon}_P^{el} + \frac{1}{2}\boldsymbol{\sigma}_V : \boldsymbol{\varepsilon}_V^{el}$.

PENER The plastic dissipated energy per unit volume, defined as $\int_0^t \boldsymbol{\sigma}_P : \dot{\boldsymbol{\varepsilon}}_P^{pl}\, dt$.

VENER The viscous dissipated energy per unit volume, defined as $\int_0^t \boldsymbol{\sigma}_V : \dot{\boldsymbol{\varepsilon}}_V^{v}\, dt$.

SDEFRES Deformation resistance, s. This output is relevant only for the Anand model.

The above definitions of the strain tensors imply that the total strain is related to the elastic, plastic, and viscous strains through the following relation:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{el} + (1-f)\,\boldsymbol{\varepsilon}^{pl} + f\boldsymbol{\varepsilon}^{v},$$

where according to the definitions given above $\boldsymbol{\varepsilon}^{pl} = \boldsymbol{\varepsilon}_P^{pl}$ and $\boldsymbol{\varepsilon}^{v} = \boldsymbol{\varepsilon}_V^{v}$. The above definitions of the output variables apply to all procedure types. In particular, when the long-term response is requested for a static procedure, the elastic-viscous network does not carry any stress and the definition of the elastic strain reduces to $\boldsymbol{\varepsilon}^{el} = (1-f)\,\boldsymbol{\varepsilon}_P^{el}$, which implies that the total stress is related to the elastic strain through the instantaneous elastic moduli.

References

- Kichenin, J., “Comportement Thermomécanique du Polyéthylène—Application aux Structures Gazières,” *Thèse de Doctorat de l'Ecole Polytechnique, Spécialité: Mécanique et Matériaux*, 1992.

ORNL – Oak Ridge National Laboratory Constitutive Model

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Classical Metal Plasticity*
- *Rate-Dependent Plasticity: Creep and Swelling*
- **ORNL*
- **PLASTIC*
- **CREEP*
- *Using the Oak Ridge National Laboratory (ORNL) constitutive model in plasticity and creep calculations*
- *Specifying cycled yield stress data for the ORNL model*

Overview

The Oak Ridge National Laboratory (ORNL) constitutive model:

- allows for use of the rules defined in the Nuclear Standard NEF 9–5T, “Guidelines and Procedures for Design of Class 1 Elevated Temperature Nuclear System Components,” in plasticity and creep calculations;
- is intended for use in modeling types 304 and 316 stainless steel at relatively high temperatures;
- can be used only with the metal plasticity models (linear kinematic hardening only) and/or the strain hardening form of the metal creep law; and
- is described in detail in *ORNL constitutive theory*.

Usage with Plasticity

The ORNL constitutive model in Abaqus/Standard is based on the March 1981 issue of the Nuclear Standard NEF 9–5T and on the October 1986 issue, which revises the constitutive model extensively. This model adds isotropic hardening of the plastic yield surface from a virgin material state to a fully cycled state. Initially the material is assumed to harden kinematically according to a bilinear representation of the virgin stress-strain curve. If a strain reversal takes place or if the creep strain reaches 0.2%, the yield surface expands isotropically to the user-defined tenth-cycle stress-strain curve. Further hardening occurs kinematically according to a bilinear representation of the tenth-cycle stress-strain curve.

You must specify the virgin yield stress and the hardening through a plasticity model definition and the elastic part of the response through a linear elasticity model definition. You specify the tenth-cycle yield stress and hardening values separately. The yield stress at each temperature should be defined by giving its value at zero plastic strain and at one additional nonzero plastic strain point, thus giving a constant hardening rate (linear work hardening).

Input File Usage: Use all of the following options in the same material data block:

**PLASTIC*
**ORNL*
**CYCLED PLASTIC*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**: **Suboptions->Ornl** and **Suboptions->Cycled Plastic**

Specifying the Optional Kinematic Shift Reset Procedure

Abaqus/Standard also allows you to invoke the optional kinematic shift ($\boldsymbol{\alpha}$) reset procedure that is described in Section 4.3.5 of the Nuclear Standard. If you do not specify the $\boldsymbol{\alpha}$ reset procedure explicitly, it is not used.

Input File Usage: **ORNL*, RESET

Abaqus/CAE Usage: Property module: material editor: **Suboptions->Ornl**: **Invoke reset procedure**

Usage with Creep

The ORNL constitutive model assumes that creep uses the strain hardening formulation. It introduces auxiliary hardening rules when strain reversals occur. An algorithm providing details is presented in *ORNL constitutive theory*. It can be used only when the creep behavior is defined by a strain-hardening power law.

Input File Usage: Use both of the following options in the same material data block:

**CREEP*, LAW=STRAIN
**ORNL*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Creep**: **Law: Strain-Hardening**: **Suboptions->Ornl**

Translation of the Yield Surface during Creep

The ORNL formulation can also cause the center of the yield surface to translate during creep for use in subsequent plastic increments; this behavior is defined through two optional user-defined parameters.

Specifying Saturation Rates for Kinematic Shift

You can specify A, the saturation rates for kinematic shift caused by creep strain as defined by Equation (15) of Section 4.3.3–3 of the Nuclear Standard. The default value is 0.3. Set A=0.0 to use the 1986 revision of the standard.

Input File Usage: **ORNL*, A=*A*

Abaqus/CAE Usage: Property module: material editor: **Suboptions->Ornl**: **Saturation rates for kinematic shift:** *A*

Specifying the Rate of Kinematic Shift

You can specify H, the rate of kinematic shift with respect to creep strain (Equation (7) of Section 4.3.2–1 of the Nuclear Standard). Set H=0.0 to use the 1986 revision of the standard. If you do not specify a value for H, it is determined according to Section 4.3.3–3 of the 1981 revision of the standard.

Input File Usage: **ORNL*, H=*H*

Abaqus/CAE Usage: Property module: material editor: **Suboptions->Ornl**: **Rate of kinematic shift wrt creep strain:** *H*

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, initial equivalent plastic strain values can be provided to specify the yield stress corresponding to the work hardened state. See *Inelastic Behavior* for additional details. Initial values can also be provided for the backstress tensor, $\boldsymbol{\alpha}$, to include strain-induced anisotropy. See *Initial Conditions* for more information. For more complicated cases initial conditions can be defined through user subroutine *HARDINI*.

Input File Usage: Use the following option to specify the initial equivalent plastic strain directly:

**INITIAL CONDITIONS*, TYPE=HARDENING

Use the following option in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine *HARDINI*:

**INITIAL CONDITIONS*, TYPE=HARDENING, USER

Abaqus/CAE Usage: Use the following options to specify the initial equivalent plastic strain directly:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use the following options in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine *HARDINI*:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The ORNL constitutive model can be used with any elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), variables associated with creep (*Rate-Dependent Plasticity: Creep and Swelling*) and the kinematic hardening plasticity models (*Models for Metals Subjected to Cyclic Loading*) are available for the ORNL constitutive model.

Deformation Plasticity

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **DEFORMATION PLASTICITY*
- *Defining deformation plasticity*

Overview

The deformation theory Ramberg-Osgood plasticity model:

- is primarily intended for use in developing fully plastic solutions for fracture mechanics applications in ductile metals; and
- cannot appear with any other mechanical response material models since it completely describes the mechanical response of the material.

One-Dimensional Model

In one dimension the model is

$$E\varepsilon = \sigma + \alpha\left(\frac{|\sigma|}{\sigma^0}\right)^{n-1}\sigma,$$

where

σ

is the stress;

ε

is the strain;

E

is Young's modulus (defined as the slope of the stress-strain curve at zero stress);

α

is the “yield” offset;

σ^0

is the yield stress, in the sense that, when $\sigma = \sigma^0$, $\varepsilon = (1+\alpha)\,\sigma^0/E$; and

n

is the hardening exponent for the "plastic" (nonlinear) term: $n > 1$.

The material behavior described by this model is nonlinear at all stress levels, but for commonly used values of the hardening exponent ($n \sim 5$ or more) the nonlinearity becomes significant only at stress magnitudes approaching or exceeding σ^0.

Generalization to Multiaxial Stress States

The one-dimensional model is generalized to multiaxial stress states using Hooke's law for the linear term and the Mises stress potential and associated flow law for the nonlinear term:

$$E\boldsymbol{\varepsilon} = (1+\nu)\,\mathbf{S} - (1-2\nu)\,p\mathbf{I} + \frac{3}{2}\alpha\left(\frac{q}{\sigma^0}\right)^{n-1}\mathbf{S},$$

where

$\boldsymbol{\varepsilon}$

is the strain tensor,

$\boldsymbol{\sigma}$

is the stress tensor,

$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I}$

is the equivalent hydrostatic stress,

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$

is the Mises equivalent stress,

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$

is the stress deviator, and

ν

is the Poisson's ratio.

The linear part of the behavior can be compressible or incompressible, depending on the value of the Poisson's ratio, but the nonlinear part of the behavior is incompressible (because the flow is normal to the Mises stress potential). The model is described in detail in *Deformation plasticity*.

You specify the parameters E, ν, σ^0, n, and α directly. They can be defined as a tabular function of temperature.

Input File Usage: **DEFORMATION PLASTICITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Deformation Plasticity**

Typical Applications

The deformation plasticity model is most commonly applied in static loading with small-displacement analysis, where the fully plastic solution must be developed in a part of the model. Generally, the load is ramped on until all points in the region being monitored satisfy the condition that the "plastic strain" dominates and, hence, exhibit fully plastic behavior, which is defined as

$$\frac{\alpha}{E}\left(\frac{q}{\sigma^0}\right)^{n-1} q > 10\frac{q}{E},$$

or

$$q > \left(\frac{10}{\alpha}\right)^{1/(n-1)} \sigma^0.$$

You can specify the name of a particular element set to be monitored in a static analysis step for fully plastic behavior. The step will end when the solutions at all constitutive calculation points in the element set are fully plastic, when the maximum number of increments specified for the step is reached, or when the time period specified for the static step is exceeded, whichever comes first.

Input File Usage: **STATIC*, FULLY PLASTIC=*ElsetName*

Abaqus/CAE Usage: Step module: **Create Step**: **General**: **Static, General**: **Other**: **Stop when region *region* is fully plastic.**

Elements

Deformation plasticity can be used with any stress/displacement element in Abaqus/Standard. Since it will generally be used for cases when the deformation is dominated by plastic flow, the use of "hybrid" (mixed formulation) or reduced-integration elements is recommended with this material model.

Other Plasticity Models

In this section:

- *Extended Drucker-Prager Models*
- *Modified Drucker-Prager/Cap Model*
- *Soft Rock Plasticity Model*
- *Mohr-Coulomb Plasticity*
- *Critical State (Clay) Plasticity Model*
- *Crushable Foam Plasticity Models*

Extended Drucker-Prager Models

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Rate-Dependent Yield*
- *Rate-Dependent Plasticity: Creep and Swelling*
- *Progressive Damage and Failure*
- **DRUCKER PRAGER*
- **DRUCKER PRAGER HARDENING*
- **RATE DEPENDENT*
- **DRUCKER PRAGER CREEP*
- **TRIAXIAL TEST DATA*
- *Defining Drucker-Prager plasticity*

Overview

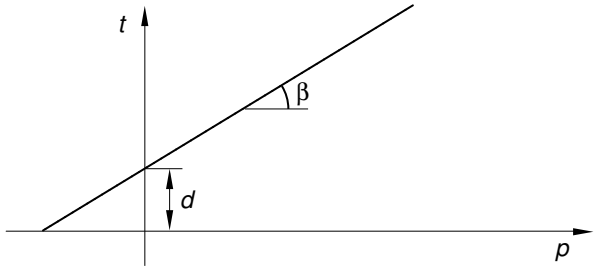
The extended Drucker-Prager models:

- are used to model frictional materials, which are typically granular-like soils and rock, and exhibit pressure-dependent yield (the material becomes stronger as the pressure increases);
- are used to model materials in which the compressive yield strength is greater than the tensile yield strength, such as those commonly found in composite and polymeric materials;
- allow a material to harden and/or soften isotropically;
- generally allow for volume change with inelastic behavior: the flow rule, defining the inelastic straining, allows simultaneous inelastic dilation (volume increase) and inelastic shearing;
- can include creep in Abaqus/Standard if the material exhibits long-term inelastic deformations;
- can be defined to be sensitive to the rate of straining, as is often the case in polymeric materials;
- can be used in conjunction with either the elastic material model (*Linear Elastic Behavior*) or, in Abaqus/Standard if creep is not defined, the porous elastic material model (*Elastic Behavior of Porous Materials*);
- can be used in conjunction with an equation of state model (*Equation of State*) to describe the hydrodynamic response of the material in Abaqus/Explicit;
- can be used in conjunction with the models of progressive damage and failure (*About Damage and Failure for Ductile Metals*) to specify different damage initiation criteria and damage evolution laws that allow for the progressive degradation of the material stiffness and the removal of elements from the mesh; and
- are intended to simulate material response under essentially monotonic loading.

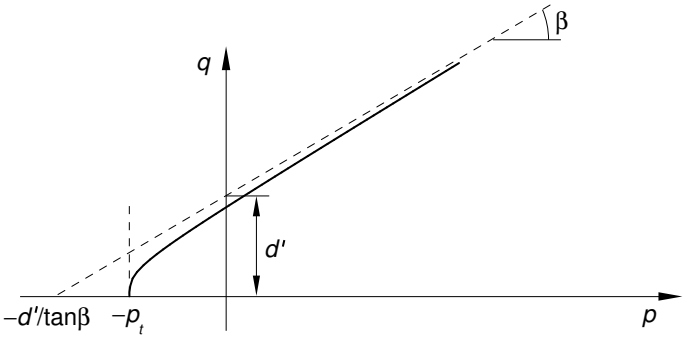
Yield Criteria

The yield criteria for this class of models are based on the shape of the yield surface in the meridional plane. The yield surface can have a linear form, a hyperbolic form, or a general exponent form. These surfaces are

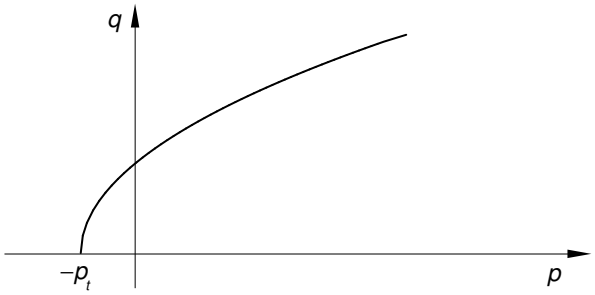
illustrated in *Figure 1*. The stress invariants and other terms in each of the three related yield criteria are defined later in this section.



a) Linear Drucker-Prager: $F = t - p \tan \beta - d = 0$



b) Hyperbolic: $F = \sqrt{(d'|_0 - p_t|_0 \tan \beta)^2 + q^2} - p \tan \beta - d' = 0$



c) Exponent form: $F = aq^b - p - p_t = 0$

Figure 1: Yield surfaces in the meridional plane.

The linear model (*Figure 1*a) provides for a possibly noncircular yield surface in the deviatoric plane ($\boldsymbol{\Pi}$-plane) to match different yield values in triaxial tension and compression, associated inelastic flow in the deviatoric plane, and separate dilation and friction angles. Input data parameters define the shape of the yield and flow surfaces in the meridional and deviatoric planes as well as other characteristics of inelastic behavior such that a range of simple theories is provided—the original Drucker-Prager model is available within this model. However, this model cannot provide a close match to Mohr-Coulomb behavior, as described later in this section.

The hyperbolic and general exponent models use a von Mises (circular) section in the deviatoric stress plane. In the meridional plane a hyperbolic flow potential is used for both models, which, in general, means nonassociated flow.

The choice of model to be used depends largely on the analysis type, the kind of material, the experimental data available for calibration of the model parameters, and the range of pressure stress values that the material is likely to experience. It is common to have either triaxial test data at different levels of confining pressure or test data that are already calibrated in terms of a cohesion and a friction angle and, sometimes, a triaxial tensile strength value. If triaxial test data are available, the material parameters must be calibrated first. The accuracy with which the linear model can match these test data is limited by the fact that it assumes linear dependence of deviatoric stress on pressure stress. Although the hyperbolic model makes a similar assumption at high confining pressures, it provides a nonlinear relationship between deviatoric and pressure stress at low confining pressures, which may provide a better match of the triaxial experimental data. The hyperbolic model is useful for brittle materials for which both triaxial compression and triaxial tension data are available, which is a common situation for materials such as rocks. The most general of the three yield criteria is the exponent form. This criterion provides the most flexibility in matching triaxial test data. Abaqus determines the material parameters required for this model directly from the triaxial test data. A least-squares fit that minimizes the relative error in stress is used for this purpose.

For cases where the experimental data are already calibrated in terms of a cohesion and a friction angle, the linear model can be used. If these parameters are provided for a Mohr-Coulomb model, it is necessary to convert them to Drucker-Prager parameters. The linear model is intended primarily for applications where the stresses are mostly compressive. If tensile stresses are significant, hydrostatic tension data should be available (along with the cohesion and friction angle) and the hyperbolic model should be used.

Calibration of these models is discussed later in this section.

Hardening and Rate Dependence

For granular materials these models are often used as a failure surface, in the sense that the material can exhibit unlimited flow when the stress reaches yield. This behavior is called perfect plasticity. The models are also provided with isotropic hardening. In this case plastic flow causes the yield surface to change size uniformly with respect to all stress directions. This hardening model is useful for cases involving gross plastic straining or in which the straining at each point is essentially in the same direction in strain space throughout the analysis. Although the model is referred to as an isotropic “hardening” model, strain softening, or hardening followed by softening, can be defined.

As strain rates increase, many materials show an increase in their yield strength. This effect becomes important in many polymers when the strain rates range between 0.1 and 1 per second; it can be very important for strain rates ranging between 10 and 100 per second, which are characteristic of high-energy dynamic events or manufacturing processes. The effect is generally not as important in granular materials. The evolution of the yield surface with plastic deformation is described in terms of the equivalent stress $\overline{\sigma}$, which can be chosen as either the uniaxial compression yield stress, the uniaxial tension yield stress, or the shear (cohesion) yield stress:

$$\begin{aligned}
\overline{\sigma} = \ & \sigma_c\left(\overline{\varepsilon}^{pl}, \dot{\overline{\varepsilon}}^{pl}, \theta, f_i\right) && \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c; \\
= \ & \sigma_t\left(\overline{\varepsilon}^{pl}, \dot{\overline{\varepsilon}}^{pl}, \theta, f_i\right) && \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t; \text{ or} \\
= \ & d\left(\overline{\varepsilon}^{pl}, \dot{\overline{\varepsilon}}^{pl}, \theta, f_i\right) && \text{if hardening is defined by the cohesion, } d,
\end{aligned}$$

where

$\dot{\overline{\varepsilon}}^{pl}$

is the equivalent plastic strain rate, defined for the linear Drucker-Prager model as

$\dot{\bar{\varepsilon}}^{pl}$

$=\left|\dot{\varepsilon}_{11}^{pl}\right|$ if hardening is defined in uniaxial compression;

$=\dot{\varepsilon}_{11}^{pl}$ if hardening is defined in uniaxial tension;

$=\dot{\gamma}^{pl}/\sqrt{3}$ if hardening is defined in pure shear,

and defined for the hyperbolic and exponential Drucker-Prager models as

$$\dot{\bar{\varepsilon}}^{pl} = \frac{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}^{pl}}{\bar{\sigma}};$$

$\bar{\varepsilon}^{pl} = \int_0^t \dot{\bar{\varepsilon}}^{pl} dt$

is the equivalent plastic strain;

θ

is temperature; and

$f_i, i = 1, 2, \ldots$

are other predefined field variables.

The functional dependence $\bar{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)$ includes hardening as well as rate-dependent effects. The material data can be input either directly in a tabular format or by correlating it to static relations based on yield stress ratios.

Rate dependence as described here is most suitable for moderate- to high-speed events in Abaqus/Standard. Time-dependent inelastic deformation at low deformation rates can be better represented by creep models. Such inelastic deformation, which can coexist with rate-independent plastic deformation, is described later in this section. However, the existence of creep in an Abaqus/Standard material definition precludes the use of rate dependence as described here.

When using the Drucker-Prager material model, Abaqus allows you to prescribe initial hardening by defining initial equivalent plastic strain values, as discussed below along with other details regarding the use of initial conditions.

Direct Tabular Data

Test data are entered as tables of yield stress values versus equivalent plastic strain at different equivalent plastic strain rates; one table per strain rate. Compression data are more commonly available for geological materials, whereas tension data are usually available for polymeric materials. The guidelines on how to enter these data are provided in *Rate-Dependent Yield*.

Input File Usage: **DRUCKER PRAGER HARDENING*, RATE=$\dot{\bar{\varepsilon}}^{pl}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Suboptions->Drucker Prager Hardening**: toggle on **Use strain-rate-dependent data**

Yield Stress Ratios

Alternatively, the strain rate behavior can be assumed to be separable, so that the stress-strain dependence is similar at all strain rates:

$$\overline{\sigma} = \sigma^0 \left(\overline{\varepsilon}^{pl}, \theta, f_i\right) R\left(\dot{\overline{\varepsilon}}^{pl}, \theta, f_i\right),$$

where $\sigma^0 \left(\overline{\varepsilon}^{pl}, \theta, f_i\right)$ is the static stress-strain behavior and $R\left(\dot{\overline{\varepsilon}}^{pl}, \theta, f_i\right)$ is the ratio of the yield stress at nonzero strain rate to the static yield stress (so that $R\left(0, \theta, f_i\right) = 1.0$).

Two methods are offered to define *R* in Abaqus: specifying an overstress power law or defining the variable *R* directly as a tabular function of $\dot{\overline{\varepsilon}}^{pl}$.

Overstress Power Law

The Cowper-Symonds overstress power law has the form

$$\dot{\overline{\varepsilon}}^{pl} = D(R-1)^n \quad \text{for} \quad \overline{\sigma} \geq \sigma^0,$$

where $D\left(\theta, f_i\right)$ and $n\left(\theta, f_i\right)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: Use both of the following options:

**DRUCKER PRAGER HARDENING*
**RATE DEPENDENT*, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Suboptions->Drucker Prager Hardening**; **Suboptions->Rate Dependent**: **Hardening: Power Law**

Chaboche Rate Dependence

Chaboche rate dependence has the form

$$\dot{\overline{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left\langle \frac{\overline{\sigma} - \sigma^0}{K} \right\rangle^n,$$

where $\dot{\varepsilon}_0\left(\theta, f_i\right)$, $K\left(\theta, f_i\right)$, and $n\left(\theta, f_i\right)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

The above relation can be rewritten as

$$\dot{\overline{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left(\frac{\sigma^0}{K}\right)^n \left(R-1\right)^n.$$

Input File Usage: Use both of the following options:

**DRUCKER PRAGER HARDENING*
**RATE DEPENDENT*, TYPE=CHABOCHE

Abaqus/CAE Usage: Chaboche rate dependence is not supported in Abaqus/CAE.

Tabular Function

When R is entered directly, it is entered as a tabular function of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i.

Input File Usage: Use both of the following options:

**DRUCKER PRAGER HARDENING*
**RATE DEPENDENT*, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Suboptions->Drucker Prager Hardening**; **Suboptions->Rate Dependent**: **Hardening: Yield Ratio**

Johnson-Cook Rate Dependence

Johnson-Cook rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \exp\left[\frac{1}{C}(R-1)\right] \quad \text{for} \quad \bar{\sigma} \geq \sigma^0,$$

where $\dot{\varepsilon}_0$ and C are material constants that do not depend on temperature and are assumed not to depend on predefined field variables.

Input File Usage: Use both of the following options:

**DRUCKER PRAGER HARDENING*
**RATE DEPENDENT*, TYPE=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Suboptions->Drucker Prager Hardening**; **Suboptions->Rate Dependent**: **Hardening: Johnson-Cook**

Stress Invariants

The yield stress surface makes use of two invariants, defined as the equivalent pressure stress,

$$p = -\frac{1}{3}\,\text{trace}\ (\boldsymbol{\sigma}),$$

and the Mises equivalent stress,

$$q = \sqrt{\frac{3}{2}(\mathbf{S}:\mathbf{S})},$$

where $\mathbf{S}$ is the stress deviator, defined as

$$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}.$$

In addition, the linear model also uses the third invariant of deviatoric stress,

$$r = \left(\frac{9}{2}\mathbf{S}\cdot\mathbf{S}:\mathbf{S}\right)^{\frac{1}{3}}.$$

Linear Drucker-Prager Model

The linear model is written in terms of all three stress invariants. It provides for a possibly noncircular yield surface in the deviatoric plane to match different yield values in triaxial tension and compression, associated inelastic flow in the deviatoric plane, and separate dilation and friction angles.

Yield Criterion

The linear Drucker-Prager criterion (see *Figure 1*a) is written as

$$F = t - p\tan\beta - d = 0,$$

where

$$t = \frac{1}{2}q\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right].$$

$\beta(\theta, f_i)$

is the slope of the linear yield surface in the p–t stress plane and is commonly referred to as the friction angle of the material;

d

is the cohesion of the material; and

$K(\theta, f_i)$

is the ratio of the yield stress in triaxial tension to the yield stress in triaxial compression and, thus, controls the dependence of the yield surface on the value of the intermediate principal stress (see *Figure 2*).

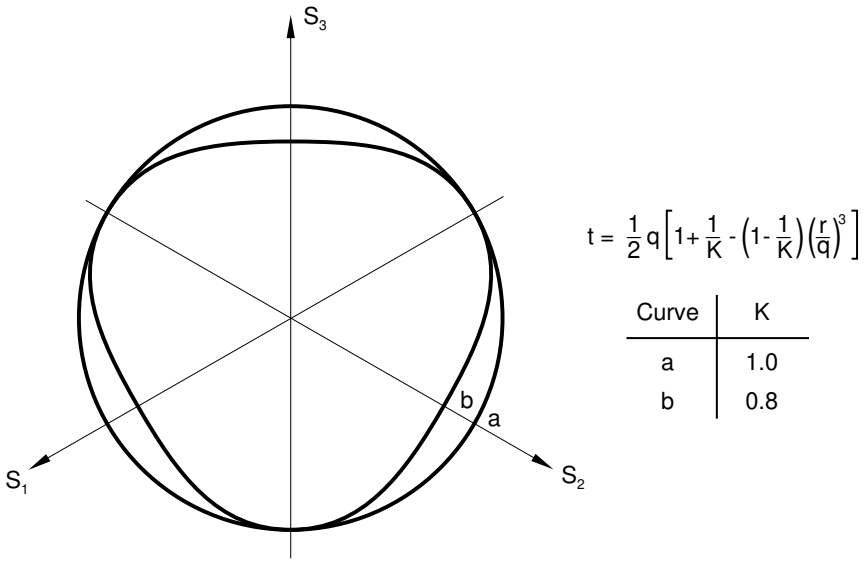


Figure 2: Typical yield/flow surfaces of the linear model in the deviatoric plane.

In the case of hardening defined in uniaxial compression, the linear yield criterion precludes friction angles $\beta >$ 71.5° ($\tan\beta >$ 3), which is unlikely to be a limitation for real materials.

When $K = 1$, $t = q$, which implies that the yield surface is the von Mises circle in the deviatoric principal stress plane (the Π-plane), in which case the yield stresses in triaxial tension and compression are the same. To ensure that the yield surface remains convex requires $0.778 \le K \le 1.0$.

The cohesion, d, of the material is related to the input data as

$$d = \left(1 - \frac{1}{3}\tan\beta\right)\sigma_c \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c;$$

$$= \left(\frac{1}{K} + \frac{1}{3}\tan\beta\right)\sigma_t \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t;$$

$$= d \quad \text{if hardening is defined by the cohesion, } d = \frac{\sqrt{3}}{2}\,\tau\left(1 + \frac{1}{K}\right).$$

Plastic Flow

G is the flow potential, chosen in this model as

$$G = t - p\tan\psi,$$

where $\psi(\theta, f_i)$ is the dilation angle in the p–t plane. A geometric interpretation of ψ is shown in the p–t diagram of *Figure 3*. In the case of hardening defined in uniaxial compression, this flow rule definition precludes dilation angles $\psi >$ 71.5° ($\tan\psi >$ 3). This restriction is not seen as a limitation since it is unlikely this will be the case for real materials.

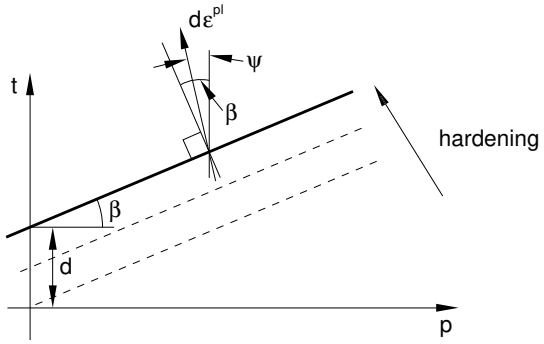


Figure 3: Linear Drucker-Prager model: yield surface and flow direction in the p–t plane.

For granular materials the linear model is normally used with nonassociated flow in the p–t plane, in the sense that the flow is assumed to be normal to the yield surface in the Π-plane but at an angle ψ to the t-axis in the p–t plane, where usually $\psi < \beta$, as illustrated in *Figure 3*. Associated flow results from setting $\psi = \beta$. The original Drucker-Prager model is available by setting $\psi = \beta$ and $K = 1$. Nonassociated flow is also generally assumed when the model is used for polymeric materials. If $\psi = 0$, the inelastic deformation is incompressible; if $\psi \ge 0$, the material dilates. Hence, ψ is referred to as the dilation angle.

The relationship between the flow potential and the incremental plastic strain for the linear model is discussed in detail in *Models for granular or polymer behavior*.

Input File Usage: **DRUCKER PRAGER*, SHEAR CRITERION=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Shear criterion: Linear**

Nonassociated Flow

Nonassociated flow implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see *Defining an Analysis*). If the difference between β and ψ is not large and the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence and the unsymmetric matrix scheme may not be needed.

Hyperbolic and General Exponent Models

The hyperbolic and general exponent models available are written in terms of the first two stress invariants only.

Hyperbolic Yield Criterion

The hyperbolic yield criterion is a continuous combination of the maximum tensile stress condition of Rankine (tensile cutoff) and the linear Drucker-Prager condition at high confining stress. It is written as

$$F = \sqrt{l_0^2 + q^2} - p\tan\beta - d' = 0,$$

where $l_0 = d'|_0 - p_t|_0 \tan\beta$ and

$p_t|_0$

is the initial hydrostatic tension strength of the material;

$d'(\overline{\sigma})$

is the hardening parameter;

$d'|_0$

is the initial value of d'; and

$\beta(\theta, f_i)$

is the friction angle measured at high confining pressure, as shown in *Figure 1*(b).

The hardening parameter, $d'(\overline{\sigma})$, can be obtained from test data as follows:

$$\begin{aligned} d' &= \sqrt{l_0^2 + {\sigma_c}^2} - \frac{\sigma_c}{3}\tan\beta \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c; \\ &= \sqrt{l_0^2 + {\sigma_t}^2} + \frac{\sigma_t}{3}\tan\beta \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t; \\ &= \sqrt{l_0^2 + d^2} \quad \text{if hardening is defined by the cohesion, } d. \end{aligned}$$

The isotropic hardening assumed in this model treats β as constant with respect to stress as depicted in *Figure 4*.

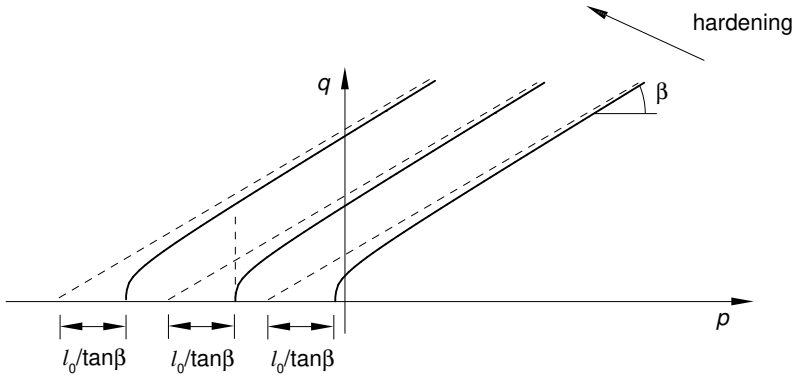


Figure 4: Hyperbolic model: yield surface and hardening in the p–q plane.

Input File Usage: **DRUCKER PRAGER*, SHEAR CRITERION=HYPERBOLIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Shear criterion: Hyperbolic**

General Exponent Yield Criterion

The general exponent form provides the most general yield criterion available in this class of models. The yield function is written as

$$F = aq^b - p - p_t = 0,$$

or

$$\omega_P = \frac{aq^b - p}{p_t} = 1,$$

where

- $a(\theta, f_i)$ and $b(\theta, f_i)$ are material parameters that are independent of plastic deformation; and
- $p_t(\overline{\sigma})$ is the hardening parameter that represents the hydrostatic tension strength of the material as shown in *Figure 1*(c).

$p_t(\overline{\sigma})$ is related to the input test data as

$$p_t = a\sigma_c^b - \frac{\sigma_c}{3} \quad \text{if hardening is defined by the uniaxial compression yield stress, } \sigma_c;$$

$$= a\sigma_t^b + \frac{\sigma_t}{3} \quad \text{if hardening is defined by the uniaxial tension yield stress, } \sigma_t;$$

$$= ad^b \quad \text{if hardening is defined by the cohesion, } d.$$

The isotropic hardening assumed in this model treats a and b as constant with respect to stress, as depicted in *Figure 5*.

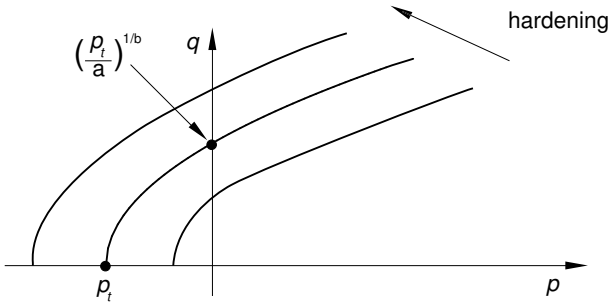


Figure 5: General exponent model: yield surface and hardening in the p–q plane.

The material parameters a and b can be given directly. Alternatively, if triaxial test data at different levels of confining pressure are available, Abaqus determines the material parameters from the triaxial test data, as discussed below.

Input File Usage: **DRUCKER PRAGER*, SHEAR CRITERION=EXPONENT FORM

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Shear criterion: Exponent Form**

Plastic Flow

G is the flow potential, chosen in these models as a hyperbolic function:

$$G = \sqrt{(\epsilon \bar{\sigma}|_0 \tan\psi)^2 + q^2} - p\tan\psi,$$

where

$\psi(\theta, f_i)$

is the dilation angle measured in the p–q plane at high confining pressure;

$\bar{\sigma}|_0 = \bar{\sigma}|_{\bar{\varepsilon}^{pl}=0, \dot{\bar{\varepsilon}}^{pl}=0}$

is the initial yield stress, taken from the user-specified Drucker-Prager hardening data; and

ϵ

is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero).

Suitable default values are provided for ϵ, as described below. The value of ϵ will depend on the yield stress used.

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane is shown in *Figure 6*. The flow potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

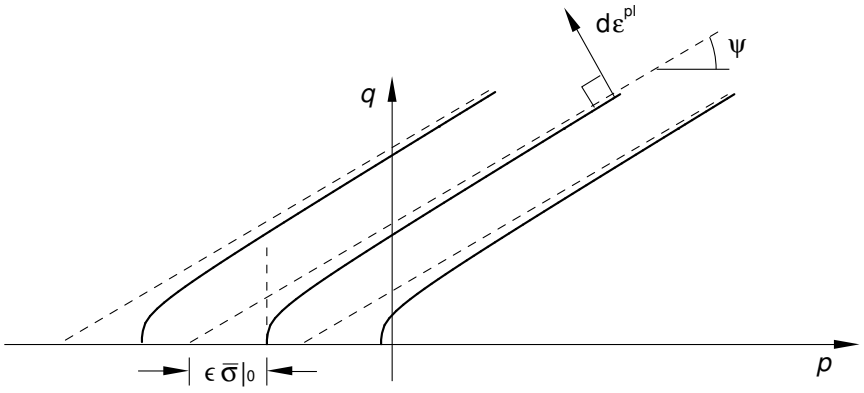


Figure 6: Family of hyperbolic flow potentials in the p–q plane.

For the hyperbolic model flow is nonassociated in the p–q plane if the dilation angle, ψ, and the material friction angle, β, are different. The hyperbolic model provides associated flow in the p–q plane only when $\beta = \psi$ and $d'|_0 / \tan\beta - p_t|_0 = \epsilon\bar{\sigma}|_0$. A default value of $\epsilon = (d'|_0 - p_t|_0 \tan\beta) / (\bar{\sigma}|_0 \tan\beta)$ is assumed if the flow potential is used with the hyperbolic model, so that associated flow is recovered when $\psi = \beta$.

For the general exponent model flow is always nonassociated in the p–q plane. The default flow potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the flow potential, implying that the dilation angle increases more rapidly as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures because of the very tight curvature of the flow potential locally where it intersects the p-axis.

The relationship between the flow potential and the incremental plastic strain for the hyperbolic and general exponent models is discussed in detail in *Models for granular or polymer behavior*.

Nonassociated Flow

Nonassociated flow implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see *Defining an Analysis*). If the difference between β and ψ in the hyperbolic model is not large and if the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence. In such cases the unsymmetric matrix scheme may not be needed.

Progressive Damage and Failure

In Abaqus/Explicit the extended Drucker-Prager models can be used in conjunction with the models of progressive damage and failure discussed in *About Damage and Failure for Ductile Metals*. The capability allows for the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The model offers two failure choices, including the removal of elements from the mesh as a result of tearing or ripping of the structure. The progressive damage models allow for a smooth degradation of the material stiffness, making them suitable for both quasi-static and dynamic situations.

Input File Usage: Use the following options:

**DAMAGE INITIATION*
**DAMAGE EVOLUTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*damage initiation type***: specify the damage initiation criterion: **Suboptions->Damage Evolution**: specify the damage evolution parameters

Matching Experimental Triaxial Test Data

Data for geological materials are most commonly available from triaxial testing. In such a test the specimen is confined by a pressure stress that is held constant during the test. The loading is an additional tension or compression stress applied in one direction. Typical results include stress-strain curves at different levels of confinement, as shown in *Figure 7*.

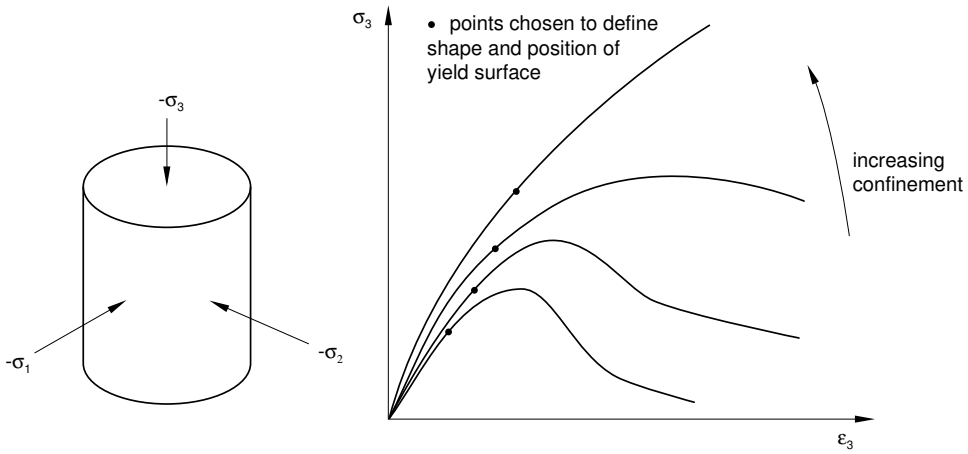


Figure 7: Triaxial tests with stress-strain curves for typical geological materials at different levels of confinement.

To calibrate the yield parameters for this class of models, you need to decide which point on each stress-strain curve will be used for calibration. For example, to calibrate the initial yield surface, the point in each stress-strain curve corresponding to initial deviation from elastic behavior should be used. Alternatively, to calibrate the ultimate yield surface, the point in each stress-strain curve corresponding to the peak stress should be used.

One stress data point from each stress-strain curve at a different level of confinement is plotted in the meridional stress plane (p–t plane if the linear model is to be used, or p–q plane if the hyperbolic or general exponent model will be used). This technique calibrates the shape and position of the yield surface, as shown in *Figure 8*, and is adequate to define a model if it is to be used as a failure surface (perfect plasticity).

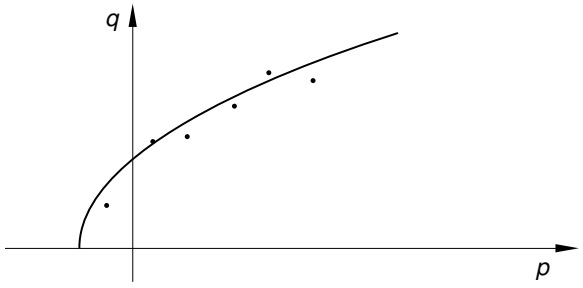


Figure 8: Yield surface in meridional plane.

The models are also available with isotropic hardening, in which case hardening data are required to complete the calibration. In an isotropic hardening model plastic flow causes the yield surface to change size uniformly; in other words, only one of the stress-strain curves depicted in *Figure 7* can be used to represent hardening. The

curve that represents hardening most accurately over the range of loading conditions anticipated should be selected (usually the curve for the average anticipated value of pressure stress).

As stated earlier, two types of triaxial test data are commonly available for geological materials. In a triaxial compression test the specimen is confined by pressure and an additional compression stress is superposed in one direction. Thus, the principal stresses are all negative, with $0 \geq \sigma_1 = \sigma_2 \geq \sigma_3$ (*Figure 9*a). In the preceding inequality σ_1, σ_2, and σ_3 are the maximum, intermediate, and minimum principal stresses, respectively.

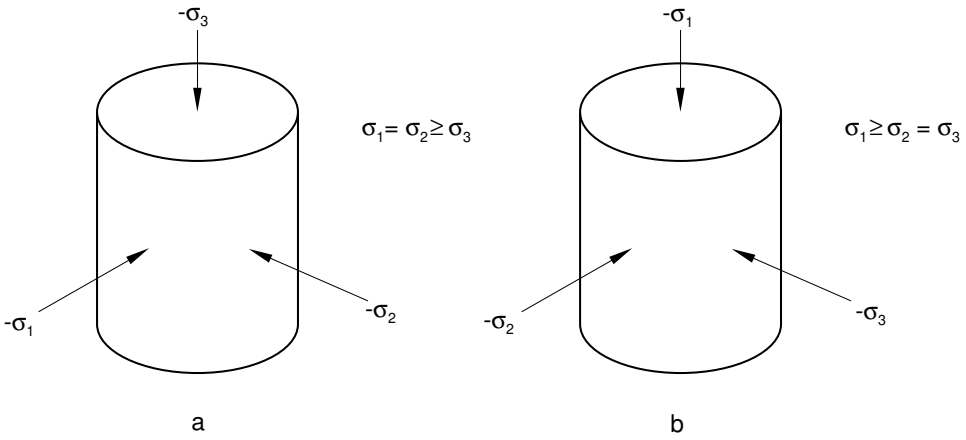


Figure 9: a) Triaxial compression and b) tension.

The values of the stress invariants are

$$p = -\frac{1}{3}(2\sigma_1 + \sigma_3),$$
$$q = \sigma_1 - \sigma_3,$$

and

$$r^3 = -(\sigma_1 - \sigma_3)^3,$$

so that

$$t = q = \sigma_1 - \sigma_3.$$

The triaxial compression results can, thus, be plotted in the meridional plane shown in *Figure 8*.

Linear Drucker-Prager Model

Fitting the best straight line through the triaxial compression results provides β and d for the linear Drucker-Prager model.

Triaxial tension data are also needed to define K in the linear Drucker-Prager model. Under triaxial tension the specimen is again confined by pressure, after which the pressure in one direction is reduced. In this case the principal stresses are $0 \geq \sigma_1 \geq \sigma_2 = \sigma_3$ (*Figure 9*b).

The stress invariants are now

$$p = -\frac{1}{3}(\sigma_1 + 2\sigma_3),$$
$$q = \sigma_1 - \sigma_3,$$

and

$$r^3 = (\sigma_1 - \sigma_3)^3,$$

so that

$$t = \frac{q}{K} = \frac{1}{K}(\sigma_1 - \sigma_3).$$

Thus, K can be found by plotting these test results as q versus p and again fitting the best straight line. The triaxial compression and tension lines must intercept the p-axis at the same point, and the ratio of values of q for triaxial tension and compression at the same value of p then gives K (*Figure 10*).

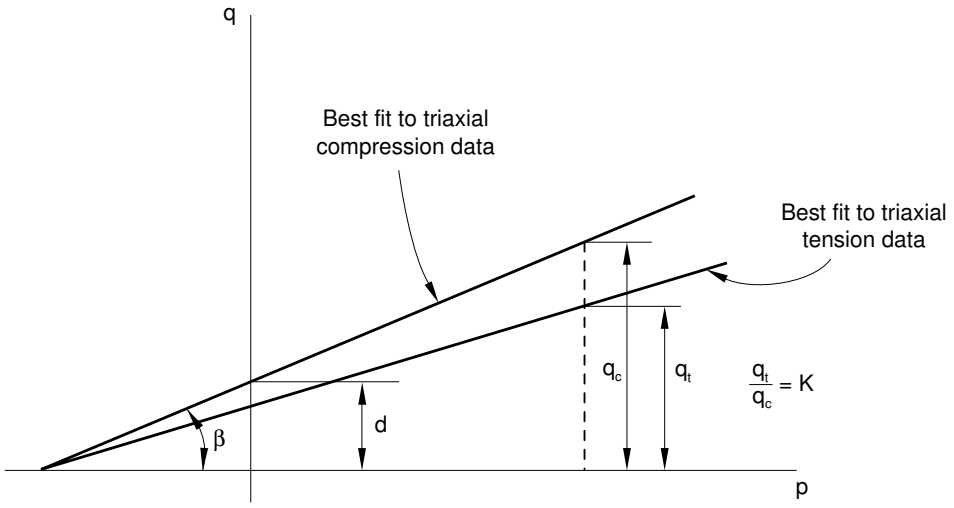


Figure 10: Linear model: fitting triaxial compression and tension data.

Hyperbolic Model

Fitting the best straight line through the triaxial compression results at high confining pressures provides β and d' for the hyperbolic model. This fit is performed in the same manner as that used to obtain β and d for the linear Drucker-Prager model. In addition, hydrostatic tension data are required to complete the calibration of the hyperbolic model so that the initial hydrostatic tension strength, $p_t|_0$, can be defined.

General Exponent Model

Given triaxial data in the meridional plane, Abaqus provides a capability to determine the material parameters a, b, and p_t required for the exponent model. The parameters are determined on the basis of a "best fit" of the triaxial test data at different levels of confining stress. A least-squares fit which minimizes the relative error in stress is used to obtain the "best fit" values for a, b, and p_t. The capability allows all three parameters to be calibrated or, if some of the parameters are known, only the remaining parameters to be calibrated. This ability is useful if only a few data points are available, in which case you may wish to fit the best straight line ($b = 1$) through the data points (effectively reducing the model to a linear Drucker-Prager model). Partial calibration can also be useful in a case when triaxial test data at low confinement are unreliable or unavailable, as is often the case for cohesionless materials. In this case a better fit may be obtained if the value of p_t is specified and only a and b are calibrated.

The data must be provided in terms of the principal stresses σ_1 $(= \sigma_2)$ and σ_3, where σ_1 is the confining stress and σ_3 is the stress in the loading direction. The Abaqus sign convention must be followed such that tensile stresses are positive and compressive stresses are negative. One pair of stresses must be entered from each triaxial test. As many data points as desired can be entered from triaxial tests at different levels of confining stress.

If the exponent model is used as a failure surface (perfect plasticity), the Drucker-Prager hardening behavior does not have to be specified. The hydrostatic tension strength, p_t, obtained from the calibration will then be used as the failure stress. However, if the Drucker-Prager hardening behavior is specified together with the triaxial test data, the value of p_t obtained from the calibration will be ignored. In this case Abaqus will interpolate p_t directly from the hardening data.

Input File Usage: Use both of the following options:

**DRUCKER PRAGER*, SHEAR CRITERION=EXPONENT FORM,
TEST DATA
**TRIAXIAL TEST DATA*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Shear criterion: Exponent Form**, toggle on **Use Suboption Triaxial Test Data**, and select **Suboptions->Triaxial Test Data**

Matching Mohr-Coulomb Parameters to the Drucker-Prager Model

Sometimes experimental data are not directly available. Instead, you are provided with the friction angle and cohesion values for the Mohr-Coulomb model. In that case the simplest way to proceed is to use the Mohr-Coulomb model (see *Mohr-Coulomb Plasticity*). In some situations it may be necessary to use the Drucker-Prager model instead of the Mohr-Coulomb model (such as when rate effects need to be considered), in which case we need to calculate values for the parameters of a Drucker-Prager model to provide a reasonable match to the Mohr-Coulomb parameters.

The Mohr-Coulomb failure model is based on plotting Mohr's circle for states of stress at failure in the plane of the maximum and minimum principal stresses. The failure line is the best straight line that touches these Mohr's circles (*Figure 11*).

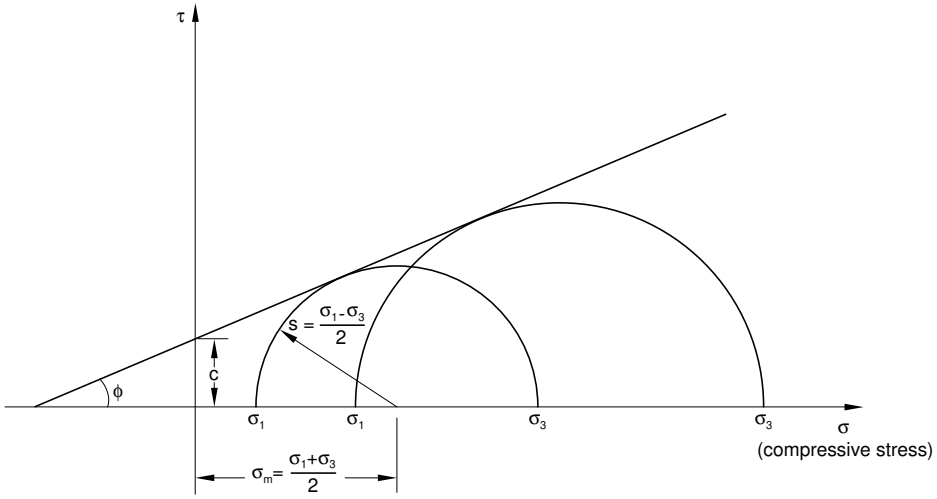


Figure 11: Mohr-Coulomb failure model.

Therefore, the Mohr-Coulomb model is defined by

$$\tau = c - \sigma \tan \phi,$$

where σ is negative in compression. From Mohr's circle,

$$\tau = s\cos\phi,$$
$$\sigma = \sigma_m + s\sin\phi.$$

Substituting for τ and σ, multiplying both sides by $\cos\phi$, and reducing, the Mohr-Coulomb model can be written as

$$s + \sigma_m \sin\phi - c\cos\phi = 0,$$

where

$$s = \frac{1}{2}(\sigma_1 - \sigma_3)$$

is half of the difference between the maximum principal stress, σ_1, and the minimum principal stress, σ_3 (and is, therefore, the maximum shear stress),

$$\sigma_m = \frac{1}{2}(\sigma_1 + \sigma_3)$$

is the average of the maximum and minimum principal stresses, and ϕ is the friction angle. Thus, the model assumes a linear relationship between deviatoric and pressure stress and, so, can be matched by the linear or hyperbolic Drucker-Prager models provided in Abaqus.

The Mohr-Coulomb model assumes that failure is independent of the value of the intermediate principal stress, but the Drucker-Prager model does not. The failure of typical geotechnical materials generally includes some small dependence on the intermediate principal stress, but the Mohr-Coulomb model is generally considered to be sufficiently accurate for most applications. This model has vertices in the deviatoric plane (see *Figure 12*).

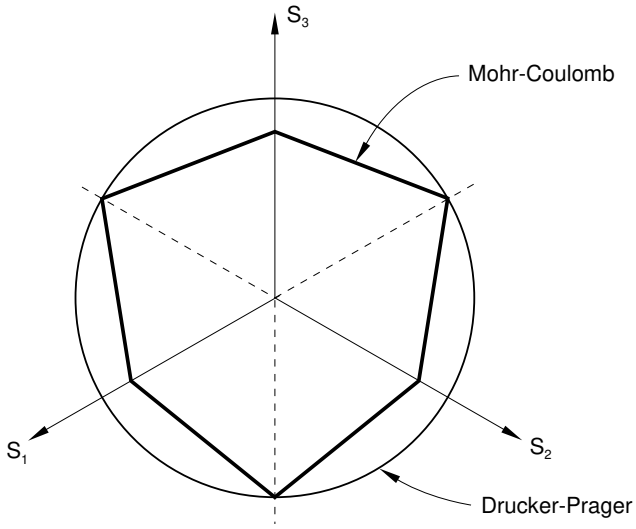


Figure 12: Mohr-Coulomb model in the deviatoric plane.

The implication is that, whenever the stress state has two equal principal stress values, the flow direction can change significantly with little or no change in stress. None of the models currently available in Abaqus can provide such behavior; even in the Mohr-Coulomb model the flow potential is smooth. This limitation is generally not a key concern in many design calculations involving Coulomb-like materials, but it can limit the accuracy of the calculations, especially in cases where flow localization is important.

Matching Plane Strain Response

Plane strain problems are often encountered in geotechnical analysis; for example, long tunnels, footings, and embankments. Therefore, the constitutive model parameters are often matched to provide the same flow and failure response in plane strain.

The matching procedure described below is carried out in terms of the linear Drucker-Prager model but is also applicable to the hyperbolic model at high levels of confining stress.

The linear Drucker-Prager flow potential defines the plastic strain increment as

$$d\boldsymbol{\varepsilon}^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{\left(1-\frac{1}{3}\tan\psi\right)} \frac{\partial}{\partial \boldsymbol{\sigma}} \left(t - p\tan\psi\right),$$

where $d\bar{\varepsilon}^{pl}$ is the equivalent plastic strain increment. Since we wish to match the behavior in only one plane, we can take $K = 1$, which implies that $t = q$. Thus,

$$d\boldsymbol{\varepsilon}^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{\left(1-\frac{1}{3}\tan\psi\right)} \left(\frac{\partial q}{\partial \boldsymbol{\sigma}} - \tan\psi \frac{\partial p}{\partial \boldsymbol{\sigma}}\right).$$

Writing this expression in terms of principal stresses provides

$$d\varepsilon_1^{pl} = d\bar{\varepsilon}^{pl} \frac{1}{\left(1-\frac{1}{3}\tan\psi\right)} \left(\frac{1}{2q}\left(2\sigma_1 - \sigma_2 - \sigma_3\right) + \frac{1}{3}\tan\psi\right),$$

with similar expressions for $d\varepsilon_2^{pl}$ and $d\varepsilon_3^{pl}$. Assume plane strain in the 1-direction. At limit load we must have $d\varepsilon_1^{pl} = 0$, which provides the constraint

$$\sigma_1 = \frac{1}{2}\left(\sigma_2 + \sigma_3\right) - \frac{1}{3}\tan\psi\, q.$$

Using this constraint we can rewrite q and p in terms of the principal stresses in the plane of deformation, σ_2 and σ_3, as

$$q = \frac{3\sqrt{3}}{2\sqrt{9-\tan^2\psi}}\left(\sigma_2 - \sigma_3\right),$$

and

$$p = -\frac{1}{2}\left(\sigma_2 + \sigma_3\right) + \frac{\tan\psi}{2\sqrt{3\left(9-\tan^2\psi\right)}}\left(\sigma_2 - \sigma_3\right).$$

With these expressions the Drucker-Prager yield surface can be written in terms of σ_2 and σ_3 as

$$\frac{9-\tan\beta\tan\psi}{2\sqrt{3\left(9-\tan^2\psi\right)}}\left(\sigma_2 - \sigma_3\right) + \frac{1}{2}\tan\beta\left(\sigma_2 + \sigma_3\right) - d = 0.$$

The Mohr-Coulomb yield surface in the $(2, 3)$ plane is

$$\sigma_2 - \sigma_3 + \sin\phi\left(\sigma_2 + \sigma_3\right) - 2c\cos\phi = 0.$$

By comparison,

$$\sin\phi = \frac{\tan\beta\sqrt{3\left(9-\tan^2\psi\right)}}{9-\tan\beta\tan\psi},$$

$$c\cos\phi = \frac{\sqrt{3\left(9-\tan^2\psi\right)}}{9-\tan\beta\tan\psi}\, d.$$

These relationships provide a match between the Mohr-Coulomb material parameters and linear Drucker-Prager material parameters in plane strain. Consider the two extreme cases of flow definition: associated flow, $\psi = \beta$, and nondilatant flow, when $\psi = 0$. For associated flow

$$\tan\beta = \frac{\sqrt{3}\sin\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}} \text{ and } \frac{d}{c} = \frac{\sqrt{3}\cos\phi}{\sqrt{1+\frac{1}{3}\sin^2\phi}},$$

and for nondilatant flow

$$\tan\beta = \sqrt{3}\sin\phi \text{ and } \frac{d}{c} = \sqrt{3}\cos\phi.$$

In either case σ_c^0 is immediately available as

$$\sigma_c^0 = \frac{1}{1-\frac{1}{3}\tan\beta}\, d.$$

The difference between these two approaches increases with the friction angle; however, the results are not very different for typical friction angles, as illustrated in *Table 1*.

Table 1: Plane strain matching of Drucker-Prager and Mohr-Coulomb models.

Mohr-Coulomb friction angle, ϕ	Associated flow		Nondilatant flow	
	Drucker-Prager friction angle, β	d/c	Drucker-Prager friction angle, β	d/c
10°	16.7°	1.70	16.7°	1.70
20°	30.2°	1.60	30.6°	1.63
30°	39.8°	1.44	40.9°	1.50
40°	46.2°	1.24	48.1°	1.33
50°	50.5°	1.02	53.0°	1.11

Limit load calculations with granular materials and *Finite deformation of an elastic-plastic granular material* show a comparison of the response of a simple loading of a granular material using the Drucker-Prager and Mohr-Coulomb models, using the plane strain approach to match the parameters of the two models.

Matching Triaxial Test Response

Another approach to matching Mohr-Coulomb and Drucker-Prager model parameters for materials with low friction angles is to make the two models provide the same failure definition in triaxial compression and tension. The following matching procedure is applicable only to the linear Drucker-Prager model since this is the only model in this class that allows for different yield values in triaxial compression and tension.

We can rewrite the Mohr-Coulomb model in terms of principal stresses:

$$\sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2c\cos\phi = 0.$$

Using the results above for the stress invariants p, q, and r in triaxial compression and tension allows the linear Drucker-Prager model to be written for triaxial compression as

$$\sigma_1 - \sigma_3 + \frac{\tan\beta}{2+\frac{1}{3}\tan\beta}(\sigma_1 + \sigma_3) - \frac{1-\frac{1}{3}\tan\beta}{1+\frac{1}{6}\tan\beta}\sigma_c^0 = 0,$$

and for triaxial tension as

$$\sigma_1 - \sigma_3 + \frac{\tan\beta}{\frac{2}{K}-\frac{1}{3}\tan\beta}(\sigma_1 + \sigma_3) - \frac{1-\frac{1}{3}\tan\beta}{\frac{1}{K}-\frac{1}{6}\tan\beta}\sigma_c^0 = 0.$$

We wish to make these expressions identical to the Mohr-Coulomb model for all values of (σ_1, σ_3). This is possible by setting

$$K = \frac{1}{1+\frac{1}{3}\tan\beta}.$$

By comparing the Mohr-Coulomb model with the linear Drucker-Prager model,

$$\tan\beta = \frac{6\sin\phi}{3-\sin\phi},$$

$$\sigma_c^0 = 2c\frac{\cos\phi}{1-\sin\phi},$$

and, hence, from the previous result

$$K = \frac{3-\sin\phi}{3+\sin\phi}.$$

These results for $\beta, K,$ and σ_c^0 provide linear Drucker-Prager parameters that match the Mohr-Coulomb model in triaxial compression and tension.

The value of *K* in the linear Drucker-Prager model is restricted to $K \geq 0.778$ for the yield surface to remain convex. The result for *K* shows that this implies $\phi \leq 22°$. Many real materials have a larger Mohr-Coulomb friction angle than this value. One approach in such circumstances is to choose $K = 0.778$ and then to use the remaining equations to define β and σ_c^0. This approach matches the models for triaxial compression only, while providing the closest approximation that the model can provide to failure being independent of the intermediate principal stress. If ϕ is significantly larger than 22°, this approach may provide a poor Drucker-Prager match of the Mohr-Coulomb parameters. Therefore, this matching procedure is not generally recommended; use the Mohr-Coulomb model instead.

While using one-element tests to verify the calibration of the model, it should be noted that the Abaqus output variables SP1, SP2, and SP3 correspond to the principal stresses σ_3, σ_2, and σ_1, respectively.

Creep Models for the Linear Drucker-Prager Model

Classical “creep” behavior of materials that exhibit plasticity according to the extended Drucker-Prager models can be defined in Abaqus/Standard. The creep behavior in such materials is intimately tied to the plasticity behavior (through the definitions of creep flow potentials and definitions of test data), so Drucker-Prager plasticity and Drucker-Prager hardening must be included in the material definition.

Creep and plasticity can be active simultaneously, in which case the resulting equations are solved in a coupled manner. To model creep only (without rate-independent plastic deformation), large values for the yield stress should be provided in the Drucker-Prager hardening definition: the result is that the material follows the Drucker-Prager model while it creeps, without ever yielding. When using this technique, a value must also be defined for the eccentricity, since, as described below, both the initial yield stress and eccentricity affect the creep potentials. This capability is limited to the linear model with a von Mises (circular) section in the deviatoric stress plane ($K = 1$; i.e., no third stress invariant effects are taken into account) and can be combined only with linear elasticity.

Creep behavior defined by the extended Drucker-Prager model is active only during soils consolidation, coupled temperature-displacement, and transient quasi-static procedures.

Creep Formulation

The creep potential is hyperbolic, similar to the plastic flow potentials used in the hyperbolic and general exponent plasticity models. If creep properties are defined in Abaqus/Standard, the linear Drucker-Prager plasticity model

also uses a hyperbolic plastic flow potential. As a consequence, if two analyses are run, one in which creep is not activated and another in which creep properties are specified but produce virtually no creep flow, the plasticity solutions will not be exactly the same: the solution with creep not activated uses a linear plastic potential, whereas the solution with creep activated uses a hyperbolic plastic potential.

Equivalent Creep Surface and Equivalent Creep Stress

We adopt the notion of the existence of creep isosurfaces of stress points that share the same creep "intensity," as measured by an equivalent creep stress. When the material plastifies, it is desirable to have the equivalent creep surface coincide with the yield surface; therefore, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane that translates into parallels to the yield surface, as depicted in *Figure 13*.

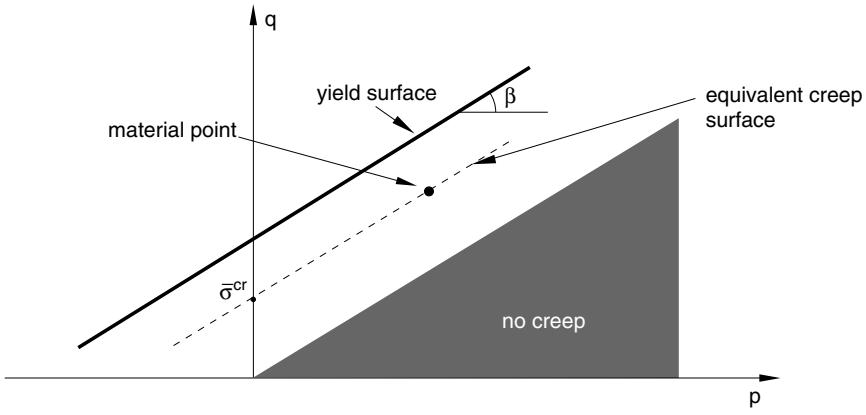


Figure 13: Equivalent creep stress defined as the shear stress.

Abaqus/Standard requires that creep properties be described in terms of the same type of data used to define work hardening properties. The equivalent creep stress, $\bar{\sigma}^{cr}$, is then determined as follows:

$$\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{\left(1 - \frac{1}{3}\tan\beta\right)} \quad \text{if creep is defined in terms of the uniaxial compression stress, } \sigma_c;$$

$$= \frac{(q - p\tan\beta)}{\left(1 + \frac{1}{3}\tan\beta\right)} \quad \text{if creep is defined in terms of the uniaxial tension stress, } \sigma_t;$$

$$= (q - p\tan\beta) \quad \text{if creep is defined in terms of the cohesion, } d.$$

Figure 13 shows how the equivalent point is determined when the material properties are in shear, with stress d. A consequence of these concepts is that there is a cone in p–q space inside which creep is not active since any point inside this cone would have a negative equivalent creep stress.

Creep Flow

The creep strain rate in Abaqus/Standard is assumed to follow from the same hyperbolic potential as the plastic strain rate (see *Figure 6*):

$$G^{cr} = \sqrt{(\epsilon\bar{\sigma}|_0 \tan\psi)^2 + q^2} - p\tan\psi,$$

where

$\psi(\theta, f_i)$

is the dilation angle measured in the p–q plane at high confining pressure;

$\overline{\sigma}|_0 = \overline{\sigma}|_{\bar{\varepsilon}^{pl}=0, \dot{\bar{\varepsilon}}^{pl}=0}$

is the initial yield stress taken from the user-specified Drucker-Prager hardening data; and

ϵ

is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the creep potential tends to a straight line as the eccentricity tends to zero).

Suitable default values are provided for ϵ, as described below. This creep potential, which is continuous and smooth, ensures that the creep flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. A family of hyperbolic potentials in the meridional stress plane was shown in *Figure 6*. The creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

The default creep potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the creep potential, implying that the dilation angle increases as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures, because of the very tight curvature of the creep potential locally where it intersects the p-axis. For details on the behavior of these models refer to *Verification of creep integration*.

If the creep material properties are defined by a compression test, numerical problems may arise for very low stress values. Abaqus/Standard protects for such a case, as described in *Models for granular or polymer behavior*.

Nonassociated Flow

The use of a creep potential different from the equivalent creep surface implies that the material stiffness matrix is not symmetric; therefore, the unsymmetric matrix storage and solution scheme should be used (see *Defining an Analysis*). If the difference between β and ψ is not large and the region of the model in which inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence and the unsymmetric matrix scheme may not be needed.

Specifying a Creep Law

The definition of creep behavior in Abaqus/Standard is completed by specifying the equivalent "uniaxial behavior"—the creep "law." In many practical cases the creep "law" is defined through user subroutine *CREEP* because creep laws are usually of very complex form to fit experimental data. Data input methods are provided for some simple cases. To avoid drawbacks of the time hardening and strain hardening forms, it is recommended that you use the time power law model rather than the time hardening form and the power law model rather than the strain hardening form, as discussed below.

User Subroutine CREEP

User subroutine *CREEP* provides a very general capability for implementing viscoplastic models in Abaqus/Standard in which the strain rate potential can be written as a function of the equivalent stress and any number of "solution-dependent state variables." When used in conjunction with these material models, the equivalent creep stress, $\overline{\sigma}^{cr}$, is made available in the routine. Solution-dependent state variables are any variables that are used in conjunction with the constitutive definition and whose values evolve with the solution. Examples

are hardening variables associated with the model. When a more general form is required for the stress potential, user subroutine *UMAT* can be used.

Input File Usage: **DRUCKER PRAGER CREEP*, LAW=USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Drucker Prager**: **Suboptions->Drucker Prager Creep**: **Law: User**

Time Power Law Model

The time power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left(\frac{\bar{\sigma}^{cr}}{q_0} \right)^n \left(\dot{\varepsilon}_0 t \right)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$ and $\bar{\sigma}^{cr}$ are defined above; and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

The model is equivalent to the "time hardening" form. It is recommended that you use the time power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent time power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: **DRUCKER PRAGER CREEP*, LAW=TIME POWER LAW

Abaqus/CAE Usage: Specifying the time power law model is not supported in Abaqus/CAE.

Power Law Model

The power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left[\left(\frac{\bar{\sigma}^{cr}}{q_0} \right)^n \left[(m+1)\, \bar{\varepsilon}^{cr} \right]^m \right)^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$, $\bar{\varepsilon}^{cr}$, and $\bar{\sigma}^{cr}$ are defined above; and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

This model is equivalent to the "strain hardening" form. It is recommended that you use the power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: **DRUCKER PRAGER CREEP*, LAW=POWER LAW

Abaqus/CAE Usage: Specifying the power law model is not supported in Abaqus/CAE.

Time Hardening Form

The "time hardening" form is

$$\dot{\bar{\varepsilon}}^{cr} = A(\bar{\sigma}^{cr})^n t^m,$$

where

$\dot{\bar{\varepsilon}}^{cr}$

is the equivalent creep strain rate, defined so that $\dot{\bar{\varepsilon}}^{cr} = |\dot{\varepsilon}_{11}^{cr}|$ if the equivalent creep stress is defined in uniaxial compression, $\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_{11}^{cr}$ if defined in uniaxial tension, and $\dot{\bar{\varepsilon}}^{cr} = \dot{\gamma}^{cr}/\sqrt{3}$ if defined in pure shear, where γ^{cr} is the engineering shear creep strain;

$\bar{\sigma}^{cr}$

is the equivalent creep stress;

t

is the total or the creep time; and

A, n, and m

are user-defined creep material parameters specified as functions of temperature and field variables.

Input File Usage: **DRUCKER PRAGER CREEP*, LAW=TIME

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Drucker Prager**: **Suboptions**->**Drucker Prager Creep**: **Law: Time**

Strain Hardening Form

The "strain hardening" form is defined as

$$\dot{\bar{\varepsilon}}^{cr} = (A(\bar{\sigma}^{cr})^n[(m+1)\,\bar{\varepsilon}^{cr}]^m)^{\frac{1}{m+1}}.$$

For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: **DRUCKER PRAGER CREEP*, LAW=STRAIN

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Drucker Prager**: **Suboptions**->**Drucker Prager Creep**: **Law: Strain**

Time-Dependent Behavior

In the “time hardening” form and the time power law model, the total time or the creep time can be used. The total time is the accumulated time over all general analysis steps. The creep time is the sum of the times of the procedures with time-dependent material behavior. If the total time is used, it is recommended that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage: Use one of the following options:

*CREEP, TIME=TOTAL (default)
*CREEP, TIME=CREEP

Abaqus/CAE Usage: Specifying the time type is not supported in Abaqus/CAE.

Singh-Mitchell Law

A second creep law available as data input is a variation of the Singh-Mitchell law:

$$\dot{\bar{\varepsilon}}^{cr} = Ae^{(\alpha\bar{\sigma}^{cr})}(t_1/t)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$, t, and $\bar{\sigma}^{cr}$ are defined above and A, α, t_1, and m are user-defined creep material parameters specified as functions of temperature and field variables. For physically reasonable behavior A and α must be positive, $0.0 < m \leq 1.0$, and t_1 should be small compared to the total time.

Input File Usage: *DRUCKER PRAGER CREEP, LAW=SINGHM

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Plasticity**->**Drucker Prager**: **Suboptions**->**Drucker Prager Creep**: **Law: SinghM**

Numerical Difficulties

Depending on the choice of units for the creep laws described above, the value of A may be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations. Therefore, use another system of units or use the time power law or power law model to avoid such difficulties in the calculation of creep strain increments.

Creep Integration

Abaqus/Standard provides both explicit and implicit time integration of creep and swelling behavior. The choice of the time integration scheme depends on the procedure type, the parameters specified for the procedure, the presence of plasticity, and whether or not a geometric linear or nonlinear analysis is requested, as discussed in *Rate-Dependent Plasticity: Creep and Swelling*.

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some work hardening, Abaqus allows you to prescribe initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$, by specifying the conditions directly (see *Initial Conditions*). For more complicated cases initial conditions can be defined in Abaqus/Standard through user subroutine *HARDINI*.

Input File Usage: Use the following option to specify the initial equivalent plastic strain directly:

*INITIAL CONDITIONS, TYPE=HARDENING

Use the following option in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine *HARDINI*:

*INITIAL CONDITIONS, TYPE=HARDENING, USER

Abaqus/CAE Usage: Use the following options to specify the initial equivalent plastic strain directly:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use the following options in Abaqus/Standard to specify the initial equivalent plastic strain in user subroutine *HARDINI*:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**; **Definition: User-defined**

Elements

The Drucker-Prager models can be used with the following element types: plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. All Drucker-Prager models are also available in plane stress (plane stress, shell, and membrane elements), except for the linear Drucker-Prager model with creep.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the Drucker-Prager plasticity/creep model:

PEEQ

Equivalent plastic strain.

For the linear Drucker-Prager plasticity model PEEQ is defined as $\bar{\varepsilon}^{pl}\big|_0 + \int_0^t \dot{\bar{\varepsilon}}^{pl} dt$; where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*) and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate.

For the hyperbolic and exponential Drucker-Prager plasticity models PEEQ is defined as $\bar{\varepsilon}^{pl}\big|_0 + \int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{pl}}{\sigma^0}$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain and σ^0 is the yield stress.

CEEQ

Equivalent creep strain, $\int \dot{\bar{\varepsilon}}^{cr} dt$.

YIELDS

Current value of yield stress, $\bar{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)$, consistent with the hardening definition option (uniaxial tension, uniaxial compression, or cohesion).

YIELDCRT

Yield criterion,

$$\omega_P = \frac{t - p\tan\beta}{d} \quad \text{for the linear model;}$$

$$= \frac{\sqrt{l_0^2 + q^2} - p\tan\beta}{d'} \quad \text{for the hyperbolic model;}$$

$$= \frac{aq^b - p}{p_t} \quad \text{for the exponent model.}$$

Modified Drucker-Prager/Cap Model

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Inelastic Behavior*
- *About the Material Library*
- *Rate-Dependent Plasticity: Creep and Swelling*
- *CREEP*
- **CAP PLASTICITY*
- **CAP HARDENING*
- **CAP CREEP*
- *Defining cap plasticity*

Overview

The modified Drucker-Prager/Cap plasticity/creep model:

- is intended to model cohesive geological materials that exhibit pressure-dependent yield, such as soils and rocks;
- is based on the addition of a cap yield surface to the Drucker-Prager plasticity model (*Extended Drucker-Prager Models*), which provides an inelastic hardening mechanism to account for plastic compaction and helps to control volume dilatancy when the material yields in shear;
- can be used in Abaqus/Standard to simulate creep in materials exhibiting long-term inelastic deformation through a cohesion creep mechanism in the shear failure region and a consolidation creep mechanism in the cap region;
- can be used with either the elastic material model (*Linear Elastic Behavior*) or, in Abaqus/Standard if creep is not defined, the porous elastic material model (*Elastic Behavior of Porous Materials*); and
- provides a reasonable response to large stress reversals in the cap region; however, in the failure surface region the response is reasonable for essentially monotonic loading only.

Yield Surface

The addition of the cap yield surface to the Drucker-Prager model serves two main purposes: it bounds the yield surface in hydrostatic compression, thus providing an inelastic hardening mechanism to represent plastic compaction; and it helps to control volume dilatancy when the material yields in shear by providing softening as a function of the inelastic volume increase created as the material yields on the Drucker-Prager shear failure surface.

The yield surface has two principal segments: a pressure-dependent Drucker-Prager shear failure segment and a compression cap segment, as shown in *Figure 1*. The Drucker-Prager failure segment is a perfectly plastic yield surface (no hardening). Plastic flow on this segment produces inelastic volume increase (dilation) that causes the cap to soften. On the cap surface, plastic flow causes the material to compact. The model is described in detail in *Drucker-Prager/Cap model for geological materials*.

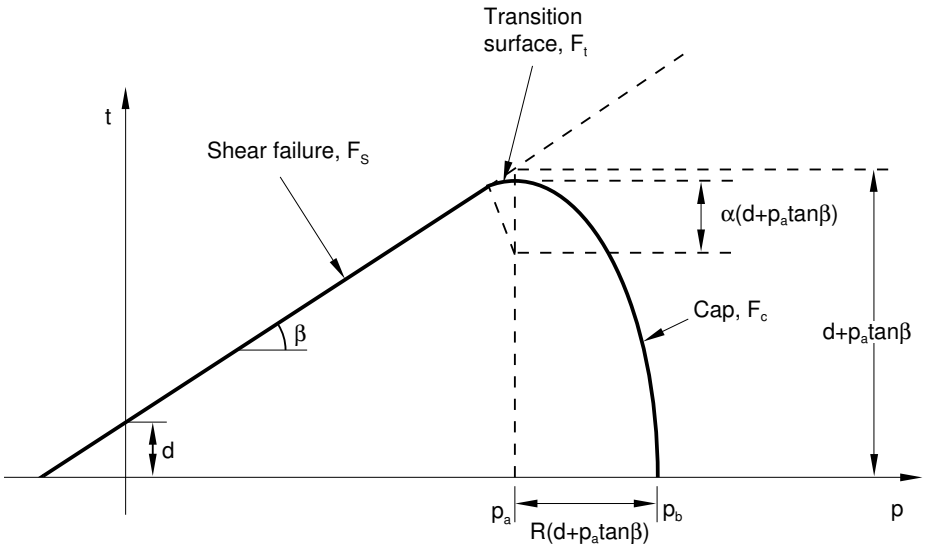


Figure 1: Modified Drucker-Prager/Cap model: yield surfaces in the *p–t* plane.

Failure Surface

The Drucker-Prager failure surface is written as

$$F_s = t - p\tan\beta - d = 0,$$

where $\beta(\theta, f_i)$ and $d(\theta, f_i)$ represent the angle of friction of the material and its cohesion, respectively, and can depend on temperature, θ, and other predefined fields $f_i, i = 1, 2, 3\ldots$. The deviatoric stress measure t is defined as

$$t = \frac{1}{2}q\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\left(\frac{r}{q}\right)^3\right];$$

and

$$p = -\frac{1}{3}\,\mathrm{trace}\ (\boldsymbol{\sigma})$$

is the equivalent pressure stress,

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$$

is the Mises equivalent stress,

$$r = \left(\frac{9}{2}\mathbf{S} : \mathbf{S} \cdot \mathbf{S}\right)^{\frac{1}{3}}$$

is the third stress invariant, and

$$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$$

is the deviatoric stress.

$K(\theta, f_i)$ is a material parameter that controls the dependence of the yield surface on the value of the intermediate principal stress, as shown in *Figure 2*.

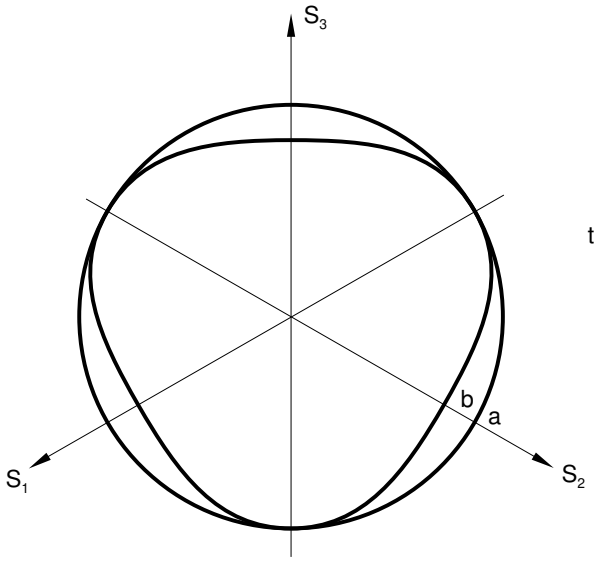


Figure 2: Typical yield/flow surfaces in the deviatoric plane.

The yield surface is defined so that K is the ratio of the yield stress in triaxial tension to the yield stress in triaxial compression. $K = 1$ implies that the yield surface is the von Mises circle in the deviatoric principal stress plane (the Π-plane), so that the yield stresses in triaxial tension and compression are the same; this is the default behavior in Abaqus/Standard and the only behavior available in Abaqus/Explicit. To ensure that the yield surface remains convex requires $0.778 \leq K \leq 1.0$.

Cap Yield Surface

The cap yield surface has an elliptical shape with constant eccentricity in the meridional (p–t) plane (*Figure 1*) and also includes dependence on the third stress invariant in the deviatoric plane (*Figure 2*). The cap surface hardens or softens as a function of the volumetric inelastic strain: volumetric plastic and/or creep compaction (when yielding on the cap and/or creeping according to the consolidation mechanism, as described later in this section) causes hardening, while volumetric plastic and/or creep dilation (when yielding on the shear failure surface and/or creeping according to the cohesion mechanism, as described later in this section) causes softening. The cap yield surface is

$$F_c = \sqrt{[p - p_a]^2 + \left[\frac{Rt}{(1+\alpha-\alpha/\cos\beta)}\right]^2} - R(d + p_a \tan\beta) = 0,$$

where $R(\theta, f_i)$ is a material parameter that controls the shape of the cap, $\alpha(\theta, f_i)$ is a small number that we discuss later, and $p_a\left(\varepsilon_{\text{vol}}^{pl} + \varepsilon_{\text{vol}}^{cr}\right)$ is an evolution parameter that represents the volumetric inelastic strain driven hardening/softening. The hardening/softening law is a user-defined piecewise linear function relating the hydrostatic compression yield stress, p_b, and volumetric inelastic strain (*Figure 3*):

$$p_b = p_b\left(\varepsilon_{\text{vol}}^{in}|_0 + \varepsilon_{\text{vol}}^{pl} + \varepsilon_{\text{vol}}^{cr}\right).$$

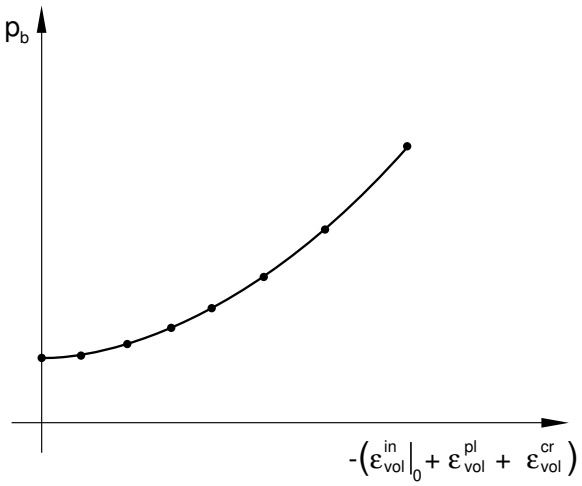


Figure 3: Typical Cap hardening.

The volumetric inelastic strain axis in *Figure 3* has an arbitrary origin: $\varepsilon_{\text{vol}}^{in}\big|_0 \left(= \varepsilon_{\text{vol}}^{pl}\big|_0 + \varepsilon_{\text{vol}}^{cr}\big|_0\right)$ is the position on this axis corresponding to the initial state of the material when the analysis begins, thus defining the position of the cap (p_b) in *Figure 1* at the start of the analysis. The evolution parameter p_a is given as

$$p_a = \frac{p_b - Rd}{(1 + R \tan\beta)}.$$

The parameter α is a small number (typically 0.01 to 0.05) used to define a transition yield surface,

$$F_t = \sqrt{[p - p_a]^2 + \left[t - \left(1 - \frac{\alpha}{\cos\beta}\right)(d + p_a \tan\beta)\right]^2} - \alpha(d + p_a \tan\beta) = 0,$$

so that the model provides a smooth intersection between the cap and failure surfaces.

Defining Yield Surface Variables

You provide the variables d, β, R, $\varepsilon_{vol}^{in}\big|_0$, α, and K to define the shape of the yield surface. In Abaqus/Standard $0.778 \le K \le 1.0$, while in Abaqus/Explicit $K = 1$ ($t = q$). If desired, combinations of these variables can also be defined as a tabular function of temperature and other predefined field variables.

Input File Usage: **CAP PLASTICITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**

Defining Hardening Parameters

The hardening curve specified for this model interprets yielding in the hydrostatic pressure sense: the hydrostatic pressure yield stress is defined as a tabular function of the volumetric inelastic strain, and, if desired, a function of temperature and other predefined field variables. The range of values for which p_b is defined should be sufficient to include all values of effective pressure stress that the material is subjected to during the analysis. If the absolute value of the volumetric inelastic strain becomes greater than the last user-specified value, Abaqus/Explicit extrapolates the stress-strain relationship based on the slope at the last point of the curve.

Input File Usage: **CAP HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**: **Suboptions->Cap Hardening**

Plastic Flow

Plastic flow is defined by a flow potential that is associated in the deviatoric plane, associated in the cap region in the meridional plane, and nonassociated in the failure surface and transition regions in the meridional plane. The flow potential surface that we use in the meridional plane is shown in *Figure 4*: it is made up of an elliptical portion in the cap region that is identical to the cap yield surface,

$$G_c = \sqrt{[p - p_a]^2 + \left[\frac{Rt}{(1+\alpha-\alpha/\cos\beta)}\right]^2},$$

and another elliptical portion in the failure and transition regions that provides the nonassociated flow component in the model,

$$G_s = \sqrt{[(p_a - p)\tan\beta]^2 + \left[\frac{t}{(1+\alpha-\alpha/\cos\beta)}\right]^2}.$$

The two elliptical portions form a continuous and smooth potential surface.

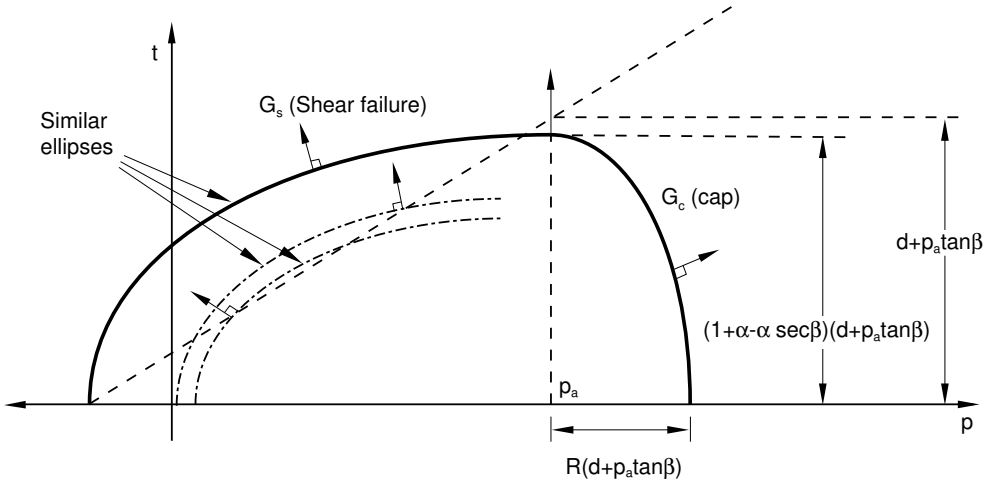


Figure 4: Modified Drucker-Prager/Cap model: flow potential in the p–t plane.

Nonassociated Flow

Nonassociated flow implies that the material stiffness matrix is not symmetric and the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see *Defining an Analysis*). If the region of the model in which nonassociated inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix gives an acceptable rate of convergence; in such cases the unsymmetric matrix scheme may not be needed.

Relative Density

The relative density of the material, r_d, is defined as the ratio of the volume of solid material to the total volume of the material. The void volume fraction or porosity, f, is defined as the ratio of the volume of voids to the total

volume of the material. It follows that $f = 1 - r_d$. The volumetric inelastic strain, $\varepsilon_{\text{vol}}^{pl}$, (defined as $\varepsilon_{\text{vol}}^{pl} = \int_0^t \dot{\varepsilon}_{kk}^{pl} dt$), drives the evolution of the relative density, r_d, as follows:

$$r_d = r_d^0 / \exp\left(\varepsilon_{\text{vol}}^{pl}\right),$$

where r_d^0 is the initial relative density of the material.

Specifying Initial Relative Density

You can specify the initial void volume fraction or porosity, f^0, at material points as initial conditions (*Initial Conditions*). This indirectly initializes r_d^0 at material points through the relation $r_d^0 = 1 - f^0$. If you do not specify f^0, Abaqus assigns f^0 a value of 0.0 and r_d^0 a value of 1.0.

Input File Usage: **INITIAL CONDITIONS*, TYPE=POROSITY

Abaqus/CAE Usage: Initial porosity is not supported in Abaqus/CAE.

Relative Density–Dependent Material Properties

You can define material properties that depend on relative density by using a field variable dependency and then associating this field variable with the scalar relative density output variable RD. This creates an explicit dependency of material properties on relative density; material properties are evaluated based on the value of relative density at the beginning of the increment.

Input File Usage: **USER DEFINED FIELD*, TYPE=SPECIFIED
field variable number, RD

Abaqus/CAE Usage: Defining relative density–dependent material properties is not supported in Abaqus/CAE.

Calibration

At least three experiments are required to calibrate the simplest version of the Cap model: a hydrostatic compression test (an odometer test is also acceptable) and either two triaxial compression tests or one triaxial compression test and one uniaxial compression test (more than two tests are recommended for a more accurate calibration).

The hydrostatic compression test is performed by pressurizing the sample equally in all directions. The applied pressure and the volume change are recorded.

The uniaxial compression test involves compressing the sample between two rigid platens. The load and displacement in the direction of loading are recorded. The lateral displacements should also be recorded so that the correct volume changes can be calibrated.

Triaxial compression experiments are performed using a standard triaxial machine where a fixed confining pressure is maintained while the differential stress is applied. Several tests covering the range of confining pressures of interest are usually performed. Again, the stress and strain in the direction of loading are recorded, together with the lateral strain so that the correct volume changes can be calibrated.

Unloading measurements in these tests are useful to calibrate the elasticity, particularly in cases where the initial elastic region is not well defined.

The hydrostatic compression test stress-strain curve gives the evolution of the hydrostatic compression yield stress, $p_b\left(\varepsilon_{vol}^{pl}\right)$, required for the cap hardening curve definition.

The friction angle, β, and cohesion, d, which define the shear failure dependence on hydrostatic pressure, are calculated by plotting the failure stresses of the two triaxial compression tests (or the triaxial compression test and the uniaxial compression test) in the pressure stress (p) versus shear stress (q) space: the slope of the straight line passing through the two points gives the angle β and the intersection with the q-axis gives d. For more details on the calibration of β and d, see the discussion on calibration in *Extended Drucker-Prager Models*.

R represents the curvature of the cap part of the yield surface and can be calibrated from a number of triaxial tests at high confining pressures (in the cap region). R must be between 0.0001 and 1000.0.

Abaqus/Standard Creep Model

Classical "creep" behavior of materials that exhibit plasticity according to the capped Drucker-Prager plasticity model can be defined in Abaqus/Standard. The creep behavior in such materials is intimately tied to the plasticity behavior (through the definitions of creep flow potentials and definitions of test data), so cap plasticity and cap hardening must be included in the material definition. If no rate-independent plastic behavior is desired in the model, large values for the cohesion, d, as well as large values for the compression yield stress, p_b, should be provided in the plasticity definition: as a result the material follows the capped Drucker-Prager model while it creeps, without ever yielding. This capability is limited to cases in which there is no third stress invariant dependence of the yield surface ($K = 1$) and cases in which the yield surface has no transition region ($\alpha = 0$). The elastic behavior must be defined using linear isotropic elasticity (see *Defining Isotropic Elasticity*).

Creep behavior defined for the modified Drucker-Prager/Cap model is active only during soils consolidation, coupled temperature-displacement, and transient quasi-static procedures.

Creep Formulation

This model has two possible creep mechanisms that are active in different loading regions: one is a cohesion mechanism, which follows the type of plasticity active in the shear-failure plasticity region, and the other is a consolidation mechanism, which follows the type of plasticity active in the cap plasticity region. *Figure 5* shows the regions of applicability of the creep mechanisms in p–q space.

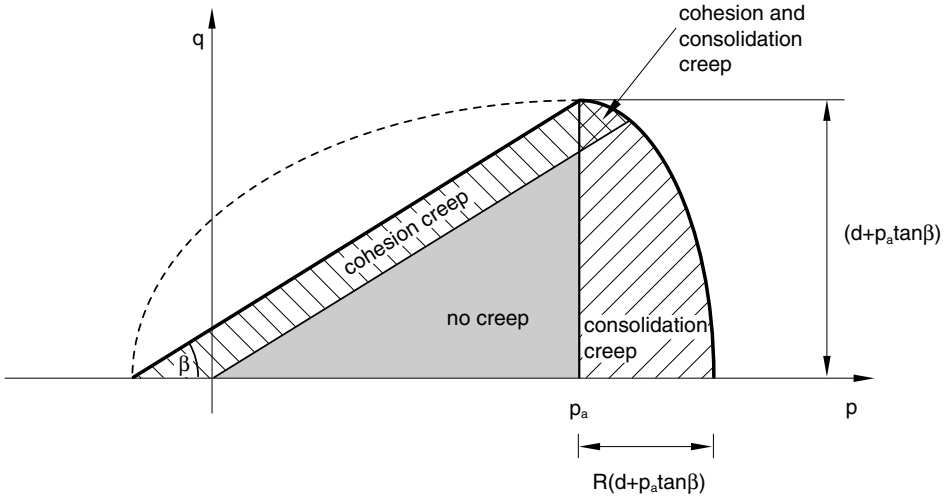


Figure 5: Regions of activity of creep mechanisms.

Equivalent Creep Surface and Equivalent Creep Stress for the Cohesion Creep Mechanism

Consider the cohesion creep mechanism first. We adopt the notion of the existence of creep isosurfaces of stress points that share the same creep "intensity," as measured by an equivalent creep stress. Since it is desirable to have the equivalent creep surface coincide with the yield surface, we define the equivalent creep surfaces by homogeneously scaling down the yield surface. In the p–q plane the equivalent creep surfaces translate into surfaces that are parallel to the yield surface, as depicted in *Figure 6*.

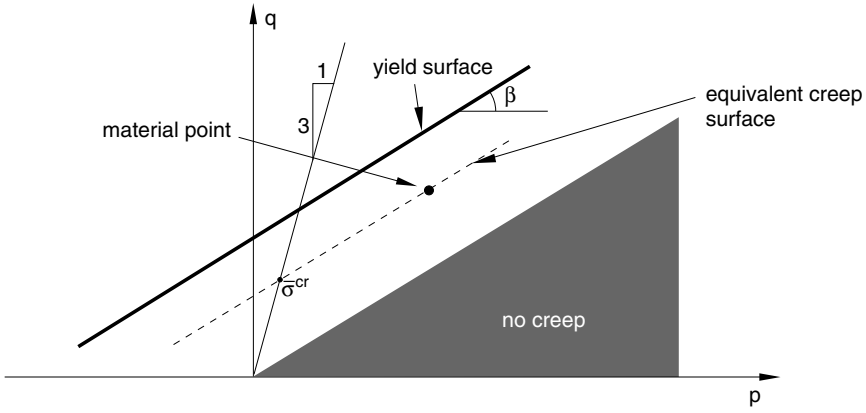


Figure 6: Equivalent creep stress for cohesion creep.

Abaqus/Standard requires that cohesion creep properties be measured in a uniaxial compression test. The equivalent creep stress, $\overline{\sigma}^{cr}$, is determined as follows:

$$\overline{\sigma}^{cr} = \frac{(q - p\tan\beta)}{\left(1 - \frac{1}{3}\tan\beta\right)}.$$

Abaqus/Standard also requires that $\overline{\sigma}^{cr}$ be positive. *Figure 6* shows such an equivalent creep stress. A consequence of these concepts is that there is a cone in p–q space inside which creep is not active. Any point inside this cone would have a negative equivalent creep stress.

Equivalent Creep Surface and Equivalent Creep Stress for the Consolidation Creep Mechanism

Next, consider the consolidation creep mechanism. In this case we wish to make creep dependent on the hydrostatic pressure above a threshold value of p_a, with a smooth transition to the areas in which the mechanism is not active ($p < p_a$). Therefore, we define equivalent creep surfaces as constant hydrostatic pressure surfaces (vertical lines in the p–q plane). Abaqus/Standard requires that consolidation creep properties be measured in a hydrostatic compression test. The effective creep pressure, $\overline{p}^{cr}$, is then the point on the p-axis with a relative pressure of $\overline{p}^{cr} = p - p_a$. This value is used in the uniaxial creep law. The equivalent volumetric creep strain rate produced by this type of law is defined as positive for a positive equivalent pressure. The internal tensor calculations in Abaqus/Standard account for the fact that a positive pressure will produce negative (that is, compressive) volumetric creep components.

Creep Flow

The creep strain rate produced by the cohesion mechanism is assumed to follow a potential that is similar to that of the creep strain rate in the Drucker-Prager creep model (*Extended Drucker-Prager Models*); that is, a hyperbolic function:

$$G_s^{cr} = \sqrt{\left(0.1\frac{d}{\left(1-\frac{1}{3}\tan\beta\right)}\tan\beta\right)^2 + q^2} - p\tan\beta.$$

This creep flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches a parallel to the shear-failure yield surface asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at a right angle. A family of hyperbolic potentials in the meridional stress plane is shown in *Figure 7*. The cohesion creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane).

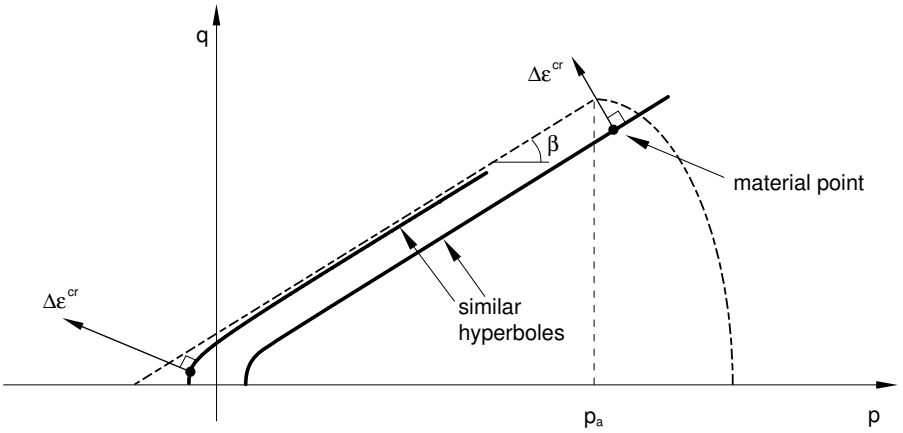


Figure 7: Cohesion creep potentials in the p–q plane.

Abaqus/Standard protects for numerical problems that may arise for very low stress values. See *Drucker-Prager/Cap model for geological materials* for details.

The creep strain rate produced by the consolidation mechanism is assumed to follow a potential that is similar to that of the plastic strain rate in the cap yield surface (*Figure 8*):

$$G_c^{cr} = \sqrt{[p - p_a]^2 + [Rq]^2}.$$

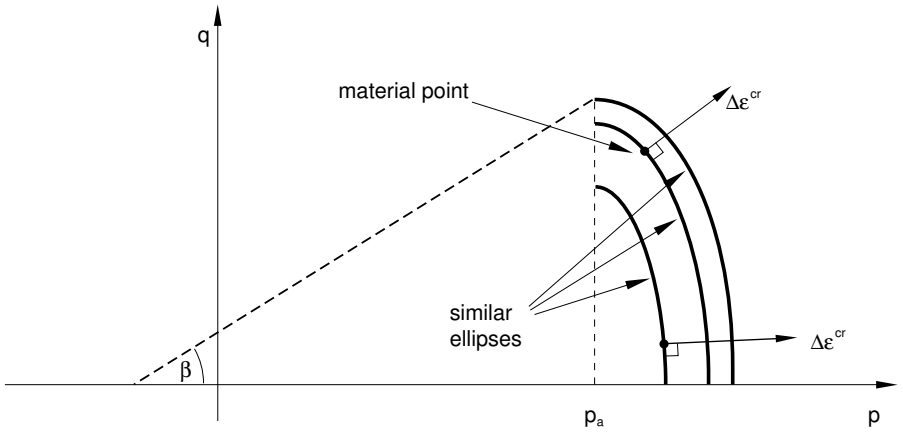


Figure 8: Consolidation creep potentials in the p–q plane.

The consolidation creep potential is the von Mises circle in the deviatoric stress plane (the Π-plane). The volumetric components of creep strain from both mechanisms contribute to the hardening/softening of the cap, as described previously. For details on the behavior of these models refer to *Verification of creep integration*.

Nonassociated Flow

The use of a creep potential for the cohesion mechanism different from the equivalent creep surface implies that the material stiffness matrix is not symmetric, and the unsymmetric matrix storage and solution scheme should be used (see *Defining an Analysis*). If the region of the model in which cohesive inelastic deformation is occurring

is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence; in such cases the unsymmetric matrix scheme may not be needed.

Specifying Creep Laws

The definition of the creep behavior is completed by specifying the equivalent "uniaxial behavior"—the creep "laws." In many practical cases the creep laws are defined through user subroutine *CREEP* because creep laws are usually of complex form to fit experimental data. Data input methods are provided for some simple cases. To avoid drawbacks of the time hardening and strain hardening forms, it is recommended that you use the time power law model rather than the time hardening form and the power law model rather than the strain hardening form, as discussed below.

User Subroutine CREEP

User subroutine *CREEP* provides a general capability for implementing viscoplastic models in which the strain rate potential can be written as a function of the equivalent stress and any number of "solution-dependent state variables." When used with these materials, the equivalent cohesion creep stress, $\overline{\sigma}^{cr}$, and the effective creep pressure, $\overline{p}^{cr}$, are made available in the routine. Solution-dependent state variables are any variables that are used with the constitutive definition and whose values evolve with the solution. Examples are hardening variables associated with the model. When a more general form is required for the stress potential, user subroutine *UMAT* can be used.

Input File Usage: Use either or both of the following options:

**CAP CREEP*, MECHANISM=COHESION, LAW=USER
**CAP CREEP*, MECHANISM=CONSOLIDATION, LAW=USER

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: User**
Suboptions->Cap Creep Consolidation: **Law: User**

Time Hardening Form

With respect to the cohesion mechanism, the "time hardening" form is available

$$\dot{\overline{\varepsilon}}^{cr} = A(\overline{\sigma}^{cr})^n t^m,$$

where

$\dot{\overline{\varepsilon}}^{cr}$

is the equivalent creep strain rate;

$\overline{\sigma}^{cr}$

is the equivalent cohesion creep stress;

t

is the total or the creep time; and

A, n, and m

are user-defined creep material parameters specified as functions of temperature and field variables.

In using this form with the consolidation mechanism, $\overline{\sigma}^{cr}$ can be replaced by $\overline{p}^{cr}$, the effective creep pressure, in the above relation.

Input File Usage: Use either or both of the following options:

**CAP CREEP*, MECHANISM=COHESION, LAW=TIME
**CAP CREEP*, MECHANISM=CONSOLIDATION, LAW=TIME

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: Time**
Suboptions->Cap Creep Consolidation: **Law: Time**

Strain Hardening Form

For the cohesion mechanism the "strain hardening" form is

$$\dot{\overline{\varepsilon}}^{cr} = (A(\overline{\sigma}^{cr})^n[(m+1)\,\overline{\varepsilon}^{cr}]^m)^{\frac{1}{m+1}}.$$

In using this form with the consolidation mechanism, $\overline{\sigma}^{cr}$ can be replaced by $\overline{p}^{cr}$, the effective creep pressure, in the above relation.

For physically reasonable behavior A and n must be positive and $-1 < m \leq 0$.

Input File Usage: Use either or both of the following options:

**CAP CREEP*, MECHANISM=COHESION, LAW=STRAIN
**CAP CREEP*, MECHANISM=CONSOLIDATION, LAW=STRAIN

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: Strain**
Suboptions->Cap Creep Consolidation: **Law: Strain**

Time Power Law Model

The time power law model has the following form:

$$\dot{\overline{\varepsilon}}^{cr} = \dot{\varepsilon}_0\left(\frac{\overline{\sigma}^{cr}}{q_0}\right)^n\left(\dot{\varepsilon}_0 t\right)^m,$$

where $\dot{\overline{\varepsilon}}^{cr}$ and $\overline{\sigma}^{cr}$ are defined above; and $\dot{\varepsilon}_0$, n, m, and q_0 are material parameters.

The model is equivalent to the “time hardening” form. It is recommended that you use the time power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent time power law model

is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: Use either or both of the following options:

**CAP CREEP*, MECHANISM=COHESION, LAW=TIME POWER LAW
**CAP CREEP*, MECHANISM=CONSOLIDATION, LAW=TIME POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: Time Power**
Suboptions->Cap Creep Consolidation: **Law: Time Power**

Power Law Model

The power law model has the following form:

$$\dot{\bar{\varepsilon}}^{cr} = \dot{\varepsilon}_0 \left[\left(\frac{\bar{\sigma}^{cr}}{q_0} \right)^n \left[(m+1)\, \bar{\varepsilon}^{cr} \right]^m \right]^{\frac{1}{m+1}},$$

where $\dot{\bar{\varepsilon}}^{cr}$, $\bar{\varepsilon}^{cr}$ and $\bar{\sigma}^{cr}$ are defined above; and $\dot{\varepsilon}_0$, n, m and q_0 are material parameters.

This model is equivalent to the "strain hardening" form. It is recommended that you use the power law model when the value of the parameter A is very small ($A \leq 10^{-27}$). In this case the equivalent power law model is obtained by setting $q_0 = \dot{\varepsilon}_0^{\frac{m+1}{n}} A^{-\frac{1}{n}}$, keeping the parameters n and m unchanged, and setting $\dot{\varepsilon}_0$ to an arbitrary value greater than zero (typically, $\dot{\varepsilon}_0$ is set to one).

Input File Usage: Define one or both of the following:

**CAP CREEP*, MECHANISM=COHESION, LAW=POWER LAW
**CAP CREEP*, MECHANISM=CONSOLIDATION, LAW=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: Power**
Suboptions->Cap Creep Consolidation: **Law: Power**

Singh-Mitchell Law

A second cohesion creep law available as data input is a variation of the Singh-Mitchell law:

$$\dot{\bar{\varepsilon}}^{cr} = A e^{(\alpha \bar{\sigma}^{cr})} (t_1/t)^m,$$

where $\dot{\bar{\varepsilon}}^{cr}$, t, and $\bar{\sigma}^{cr}$ are defined above and A, α, t_1, and m are user-defined creep material parameters specified as functions of temperature and field variables. For physically reasonable behavior A and α must be positive, $0.0 < m \leq 1.0$, and t_1 should be small compared to the total time.

In using this variation of the Singh-Mitchell law with the consolidation mechanism, $\bar{\sigma}^{cr}$ can be replaced by $\bar{p}^{cr}$, the effective creep pressure, in the above relation.

Input File Usage: Use either or both of the following options:

*CAP CREEP, MECHANISM=COHESION, LAW=SINGHM
*CAP CREEP, MECHANISM=CONSOLIDATION, LAW=SINGHM

Abaqus/CAE Usage: Define one or both of the following:

Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Law: SinghM**
Suboptions->Cap Creep Consolidation: **Law: SinghM**

Time-Dependent Behavior

In the "time hardening" form, the time power law model, and the Singh-Mitchell law model, the total time or the creep time can be used. The total time is the accumulated time over all general analysis steps. The creep time is the sum of the times of the procedures with time-dependent material behavior. If the total time is used, it is recommended that small step times compared to the creep time be used for any steps for which creep is not active in an analysis; this is necessary to avoid changes in hardening behavior in subsequent steps.

Input File Usage: Use one of the following options:

*CAP CREEP, TIME=TOTAL (default)
*CAP CREEP, TIME=CREEP

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Cap Plasticity**:
Suboptions->Cap Creep Cohesion: **Time: Total** or **Creep**
Suboptions->Cap Creep Consolidation: **Law: Total** or **Creep**

Numerical Difficulties

Depending on the choice of units for the creep laws described above, the value of A may be very small for typical creep strain rates. If A is less than 10^{-27}, numerical difficulties can cause errors in the material calculations. Therefore, use another system of units or use the time power law or power law model to avoid such difficulties in the calculation of creep strain increments.

Creep Integration

Abaqus/Standard provides both explicit and implicit time integration of creep and swelling behavior. The choice of the time integration scheme depends on the procedure type, the parameters specified for the procedure, the presence of plasticity, and whether or not a geometric linear or nonlinear analysis is requested, as discussed in *Rate-Dependent Plasticity: Creep and Swelling*.

Initial Conditions

The initial stress at a point can be defined (see *Defining Initial Stresses*). If such a stress point lies outside the initially defined cap or transition yield surfaces and under the projection of the shear failure surface in the p–t plane (illustrated in *Figure 1*), Abaqus will try to adjust the initial position of the cap to make the stress point lie on the yield surface and a warning message will be issued. If the stress point lies outside the Drucker-Prager failure surface (or above its projection), an error message will be issued and execution will be terminated.

Elements

The modified Drucker-Prager/Cap material behavior can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning in the cap plasticity/creep model:

PBCAP

Cap position, p_b.

PEQC

Equivalent plastic strains for all three possible yield/failure surfaces (Drucker-Prager failure surface - PEQC1, cap surface - PEQC2, and transition surface - PEQC3) and the total volumetric inelastic strain (PEQC4). For each yield/failure surface, the equivalent plastic strain is $\bar{\varepsilon}^{pl} = \int_0^t \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{pl} : \dot{\boldsymbol{\varepsilon}}^{pl}}\, dt$, where $\dot{\boldsymbol{\varepsilon}}^{pl}$ is the corresponding rate of plastic flow. The total volumetric inelastic strain is defined as $\varepsilon_{\text{vol}}^{in} = \int_0^t \dot{\varepsilon}_{kk}^{pl} dt + \int_0^t \dot{\varepsilon}_{kk}^{cr} dt$.. All identifiers also provide a yes/no flag (1/0 on the output database) to indicate whether the yield surface is currently active; that is, whether or not the plastic strain changed during the increment. This flag is named AC YIELD for “actively yielding.”

CEEQ

Equivalent creep strain produced by the cohesion creep mechanism, defined as $\int \frac{\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{cr}}{\bar{\sigma}^{cr}}$, where $\bar{\sigma}^{cr} = \frac{(q - p\tan\beta)}{\left(1 - \frac{1}{3}\tan\beta\right)}$ is the equivalent creep stress.

CESW

Equivalent creep strain produced by the consolidation creep mechanism, defined as $\int \frac{\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{cr}}{\bar{p}}$, where $\bar{p} = \frac{R^2 \quad q^2 + p \quad (p - p_a)}{G_c^{cr}}$ is the equivalent creep pressure.

RD

Relative density, r_d.

VVF

Void volume fraction, f.

Soft Rock Plasticity Model

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **SOFT ROCK PLASTICITY*
- **SOFT ROCK HARDENING*
- **SOFTENING REGULARIZATION*

Overview

The soft rock plasticity model provided in Abaqus:

- is intended to model the mechanical response of soft rock and weakly consolidated sands;
- describes the inelastic behavior of the material by an isotropic yield function that depends on the three stress invariants, a non-associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain;
- is based on the constitutive model proposed by Crook et al. (2006);
- requires that the elastic part of the deformation be defined by using the isotropic linear elastic material model (*Linear Elastic Behavior*) or, in Abaqus/Standard, the power-law based porous elastic material model (*Elastic Behavior of Porous Materials*) within the same material definition;
- captures the transition of the yield surface in the Π-plane from a rounded-triangular shape to a circular shape with an increase in pressure;
- allows for the hardening law to be defined by a piecewise linear form;
- may optionally include hardening in hydrostatic tension; and
- can be used in conjunction with a regularization scheme for mitigating mesh dependence in situations where the material exhibits strain localization with increasing plastic deformation.

Yield Surface

The model is based on the yield surface (*Figure 1*)

$$\sqrt{\left(e_0 p_c^0 \tan\beta\right)^2 + \left(g(\sigma)q\right)^2} - \left(p - p_t\right)\tan\beta\left(\frac{p-p_c}{p_t-p_c}\right)^{\frac{1}{n_y}} - e_0 p_c^0 \tan\beta = 0,$$

where

$p = -\frac{1}{3}\,\mathrm{trace}\,\sigma$

is the equivalent pressure stress;

$q = \sqrt{\frac{3}{2} S : S}$

is the Mises equivalent stress;

p_c

is the yield stress in hydrostatic compression;

p_c^0

is the initial value of p_c;

p_t

is the yield stress in hydrostatic tension;

β

is the friction angle;

n_y

is the material parameter that controls the shape of the yield surface in the p-q plane (*Figure 1*); and

e_0

is the eccentricity parameter.

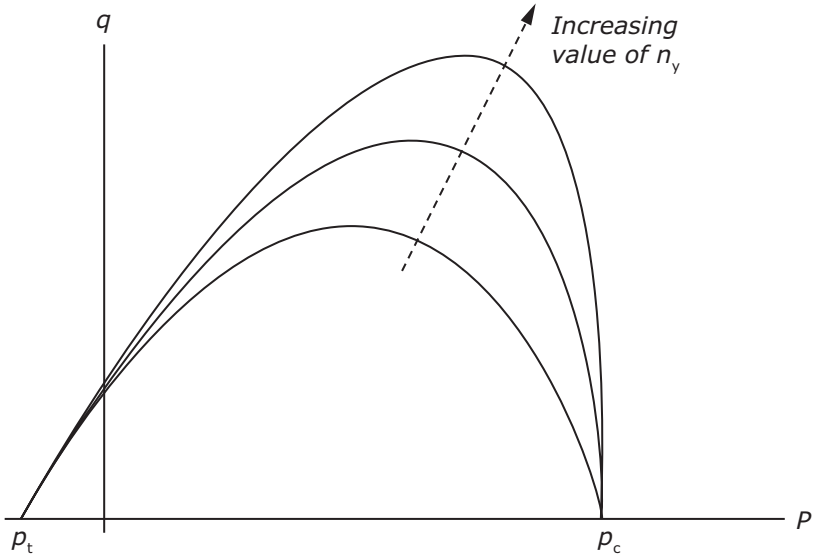


Figure 1: Yield surface of the soft rock plasticity model in the p-q plane.

$g(\sigma)$ captures the transition of the yield surface in the Π-plane from a rounded-triangular shape to a circular shape with an increase in pressure and is defined as follows (*Figure 2*):

$$g(\sigma) = \left(\frac{1}{1-f(p)}\left(1 + f\left(p\right)\left(\frac{r}{q}\right)^3\right)\right)^{\alpha},$$

$$f(p) = f_0 \exp\left(f_1 p \frac{p_c^0}{p_c}\right),$$

where $r = \left(\frac{9}{2} S : S \cdot S\right)^{\frac{1}{3}}$ is the third stress invariant; and α, f_0, and f_1 are material parameters.

Abaqus requires that the function $f(p)$ should satisfy the following conditions to ensure that the yield surface remains convex (Bigoni and Piccolroaz, 2004).

$$0 \le f(p) \le \frac{1}{9\alpha - 1} \qquad \alpha \ge \frac{3}{11},$$

$$0 \le f(p) \le \left(\frac{4 - 13\alpha}{4(\alpha - 1)(9\alpha^2 - 1)}\right)^{\frac{1}{2}} \qquad 0 < \alpha < \frac{3}{11}.$$

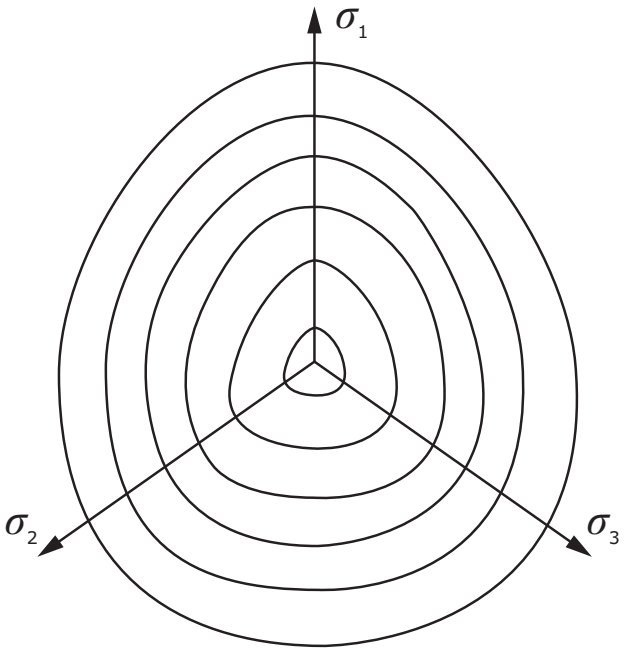


Figure 2: Yield surface of the soft rock plasticity model in the Π plane (Crook, et al., 2006).

Input File Usage: Use the following option to define the soft rock plasticity model:

**SOFT ROCK PLASTICITY*

Abaqus/CAE Usage: Use the following option to define the soft rock plasticity model:

Property module: material editor: **Mechanical->Plasticity->Soft Rock Plasticity**

Plastic Flow

Plastic flow is based on a nonassociated rule with iso-surfaces of the flow potential being used for the calculation of plastic strain rate. The flow potential surface is defined by the following function:

$$\Psi = \sqrt{\left(e_0 p_c^0 \tan\psi\right)^2 + \left(g(\sigma)q\right)^2} - (p - p_t)\tan\psi\left(\frac{p-p_c}{p_t-p_c}\right)^{\frac{1}{n_y}} - e_0 p_c^0 \tan\psi,$$

where ψ is the dilation angle and all other constants are same as in the expression for the yield surface.

Nonassociated Flow

Nonassociated flow implies that the material stiffness matrix is not symmetric and the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see *Defining an Analysis*). If the region of the model in which nonassociated inelastic deformation is occurring is confined, it is possible that a symmetric approximation to the material stiffness matrix will give an acceptable rate of convergence; in such cases the unsymmetric matrix scheme may not be needed.

Eccentricity

The eccentricity parameter, e_0, is used to ensure uniqueness of the plastic flow at $p = p_c$ and $p = p_t$. The flow potential tends to a straight line as the eccentricity tends to zero (*Figure 3*). The default value is 0.001.

Input File Usage: Use the following option to define the soft rock plasticity model using the default eccentricity value:

**SOFT ROCK PLASTICITY*

Use the following option to specify eccentricity directly:

**SOFT ROCK PLASTICITY*, ECCENTRICITY= e_0

Abaqus/CAE Usage: Use the following option to directly specify the value of eccentricity:

Property module: material editor: **Mechanical->Plasticity->Soft Rock Plasticity**: **Eccentricity: e_0**

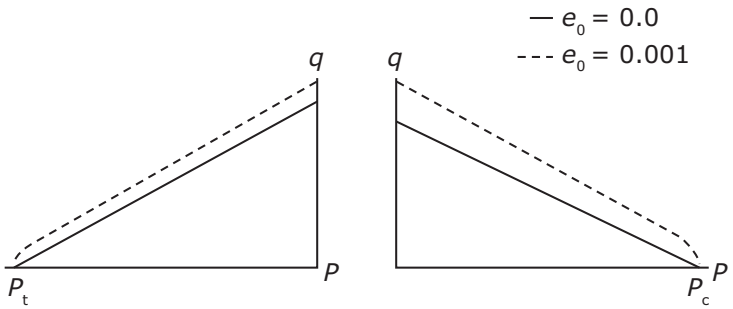


Figure 3: Effect of eccentricity on the shape of the yield surface.

Hardening Law

The hardening law has a piecewise linear form. The user-defined relationship relates the yield stress in hydrostatic compression, p_c, and, optionally, the yield stress in hydrostatic tension, p_t, to the corresponding volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}$ (*Figure 4*):

$$p_c = p_c\left(\varepsilon_{\text{vol}}^{pl}\right),$$

$$p_t = p_t\left(\varepsilon_{\text{vol}}^{pl}\right).$$

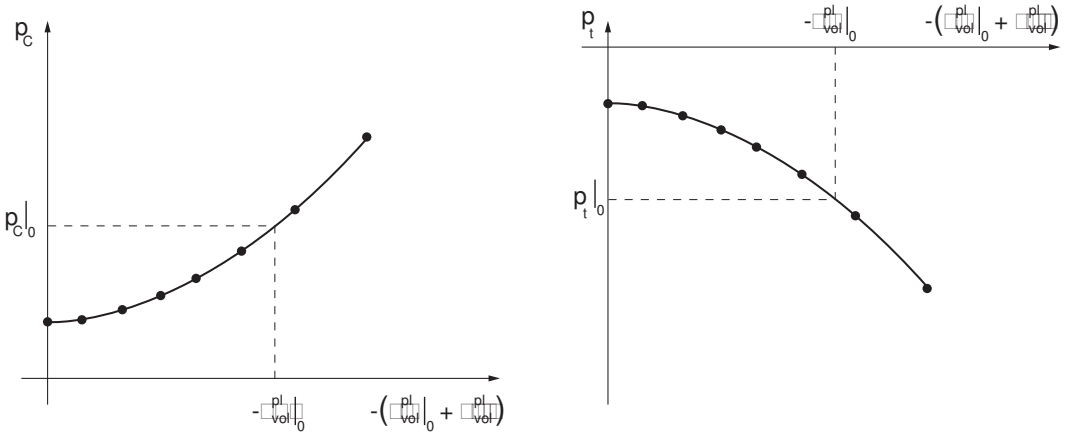


Figure 4: Typical piecewise linear soft rock hardening/softening curve.

The volumetric plastic strain axis has an arbitrary origin: $\varepsilon_{\text{vol}}^{pl}\big|_0$ is the position on this axis corresponding to the initial state of the material, thus defining the initial hydrostatic pressure in compression, $p_c|_0$, and, optionally, in tension, $p_t|_0$. This relationship is defined in tabular form as soft rock hardening data. The range of values for which p_c and p_t are defined should be sufficient to include all values of equivalent pressure stress to which the material will be subjected during the analysis. Data for p_c must be specified; data for p_t is optional.

Input File Usage: Use both of the following options to define the hardening behavior by providing the hydrostatic compression yield stress as a function of volumetric plastic strain:

**SOFT ROCK PLASTICITY*
**SOFT ROCK HARDENING*, TYPE=COMPRESSION (default)

Optionally, add the following option to define the tensile hardening behavior by providing the hydrostatic tension yield stress as a function of volumetric plastic strain:

**SOFT ROCK HARDENING*, TYPE=TENSION

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Soft Rock Plasticity**: **SuboptionsCompressive Soft Rock Hardening** and/or **Tensile Soft Rock Hardening**

Softening Regularization

Granular materials often exhibit strain localization with increasing plastic deformation. Post-failure solutions from conventional finite element methods can be strongly mesh dependent. To mitigate the mesh dependency of the solutions, a regularization method is often used to introduce a micro-structural length scale into the constitutive formulation. Let $l_c^{(m)}$ denote the characteristic width of a shear band or a crack band, $l_c^{(e)}$ denote the characteristic length of the element, and $\varepsilon_{\text{vol},e}^{pl}$ denote the inelastic strain for the element. Then the inelastic strain in the localization band, $\varepsilon_{\text{vol},m}^{pl}$, is defined to be

$$\varepsilon^{pl}_{\text{vol},m} = \varepsilon^{pl}_{\text{vol},e} \min\left(\left(\frac{l_c^{(e)}}{l_c^{(m)}}\right)^{n_r}, f_{max}\right),$$

where n_r is a material parameter and f_{max} is a positive number used for bounding the magnitude of regularization. This strain regularization method is valid only when the characteristic length of the element is greater than the width of the localization band; i.e., $l_c^{(e)} \geq l_c^{(m)}$.

If softening regularization is included, it is applied to all hardening data (tension and compression) by default. You can optionally turn off softening regularization for a specific type of hardening.

Input File Usage: Use the following options to include softening regularization:

**SOFT ROCK PLASTICITY*
**SOFT ROCK HARDENING*, SR=ON (default)
**SOFTENING REGULARIZATION*

Use the following option to turn off softening regularization:

**SOFT ROCK HARDENING*, SR=OFF

Abaqus/CAE Usage: Use the following options to include softening regularization:

Property module: material editor: **Mechanical**->**Plasticity**->**Soft Rock Plasticity**: **Suboptions**->**Softening Regularization**

Initial Conditions

If an initial stress at a point is given (see *Defining Initial Stresses*) such that the stress point lies outside the initially defined yield surface, Abaqus will try to adjust the initial position of the surface to make the stress point lie on it and issue a warning. However, if the yield stress in hydrostatic tension, p_t, is zero and does not evolve with volumetric plastic strain and the stress point is such that the equivalent pressure stress, p, is negative, Abaqus issues an error message and execution is terminated.

The initial condition on volumetric plastic strain, $\varepsilon^{pl}_{\text{vol}}\big|_0$, can be defined in the definition of the soft rock plasticity model. Abaqus also allows a general method of specifying the initial plastic strain field on elements (see *Defining Initial Values of Plastic Strain*). The volumetric plastic strain is then calculated as

$$\varepsilon^{pl}_{\text{vol}} = \varepsilon^{pl}_{\text{vol}}\big|_0 - tr\left(\varepsilon^{pl}\right).$$

Elements

The soft rock plasticity model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables are available for the soft rock plasticity model:

YIELDSC

Yield stress in hydrostatic compression, p_c.

YIELDST

Yield stress in hydrostatic tension, p_t.

PEVOL

Volumetric plastic strain, ϵ_{vol}^{pl}.

References

- Bigoni, D., and A. Piccolroaz, "Yield Criteria for Quasibrittle and Frictional Materials," International Journal of Solids and Structures, 41, 2855–2878, 2004.
- Crook, T., S. M. Willson, J.G. Yu, and D.R.J. Owen, "Predictive Modeling of Structure Evolution in Sandbox Experiments," Journal of Structural Geology, 28, pp. 729, 2006.

Mohr-Coulomb Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **MOHR COULOMB*
- **MOHR COULOMB HARDENING*
- **TENSION CUTOFF*
- *Defining Mohr-Coulomb plasticity*

Overview

The Mohr-Coulomb plasticity model:

- is used to model materials with the classical Mohr-Coloumb yield criterion;
- allows the material to harden and/or soften isotropically;
- uses a smooth flow potential that has a hyperbolic shape in the meridional stress plane and a piecewise elliptic shape in the deviatoric stress plane;
- is used with the linear elastic material model (*Linear Elastic Behavior*);
- can be used with the Rankine surface (tension cutoff) to limit load carrying capacity near the tensile region; and
- can be used for design applications in the geotechnical engineering area to simulate material response under essentially monotonic loading.

Elastic Behavior

The elastic part of the response is specified as described in *Linear Elastic Behavior*. Linear isotropic elasticity is assumed.

Plastic Behavior: Yield Criteria

The yield surface is a composite of two different criteria: a shear criterion, known as the Mohr-Coulomb surface, and an optional tension cutoff criterion, modeled using the Rankine surface.

Mohr-Coulomb Surface

The Mohr-Coulomb criterion assumes that yield occurs when the shear stress on any point in a material reaches a value that depends linearly on the normal stress in the same plane. The Mohr-Coulomb model is based on plotting Mohr's circle for states of stress at yield in the plane of the maximum and minimum principal stresses. The yield line is the best straight line that touches these Mohr's circles (*Figure 1*).

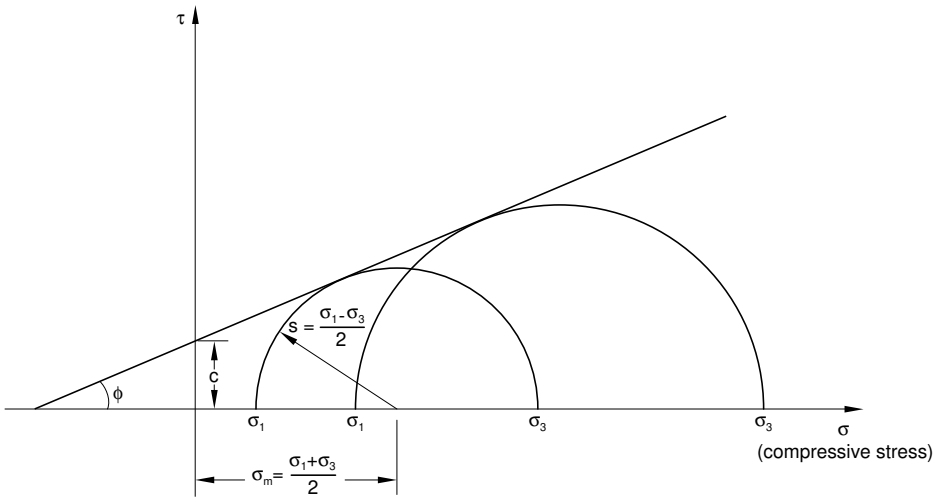


Figure 1: Mohr-Coulomb yield model.

Therefore, the Mohr-Coulomb model is defined by

$$\tau = c - \sigma \tan \phi,$$

where σ is negative in compression. From Mohr's circle,

$$\begin{aligned} \tau &= s \cos \phi, \\ \sigma &= \sigma_m + s \sin \phi. \end{aligned}$$

Substituting for τ and σ, multiplying both sides by $\cos \phi$, and reducing, the Mohr-Coulomb model can be written as

$$s + \sigma_m \sin \phi - c \cos \phi = 0,$$

where

$$s = \frac{1}{2} (\sigma_1 - \sigma_3)$$

is half of the difference between the maximum principal stress, σ_1, and the minimum principal stress, σ_3 (and is, therefore, the maximum shear stress),

$$\sigma_m = \frac{1}{2} (\sigma_1 + \sigma_3)$$

is the average of the maximum and minimum principal stresses, and ϕ is the friction angle.

For general states of stress the model is more conveniently written in terms of three stress invariants as

$$F = R_{mc} q - p \tan \phi - c = 0,$$

where

$$R_{mc} (\Theta, \phi) = \frac{1}{\sqrt{3} \cos \phi} \sin \left(\Theta + \frac{\pi}{3} \right) + \frac{1}{3} \cos \left(\Theta + \frac{\pi}{3} \right) \tan \phi,$$

ϕ

is the slope of the Mohr-Coulomb yield surface in the p–$R_{mc} q$ stress plane (see *Figure 2*), which is commonly referred to as the friction angle of the material and can depend on temperature and predefined field variables;

c

is the cohesion of the material; and

Θ

is the deviatoric polar angle defined as

$$\cos(3\Theta) = \left(\frac{r}{q}\right)^3,$$

and

$p = -\frac{1}{3}\,\text{trace}\ (\boldsymbol{\sigma})$

is the equivalent pressure stress,

$q = \sqrt{\frac{3}{2}(\mathbf{S} : \mathbf{S})}$

is the Mises equivalent stress,

$r = \left(\frac{9}{2}\mathbf{S} \cdot \mathbf{S} : \mathbf{S}\right)^{\frac{1}{3}}$

is the third invariant of deviatoric stress, and

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$

is the deviatoric stress.

The friction angle, ϕ, controls the shape of the yield surface in the deviatoric plane as shown in *Figure 2*. The tension cutoff surface is shown for a meridional angle of $\Theta = 0$. The friction angle range is $0° \leq \phi < 90°$. In the case of $\phi = 0°$ the Mohr-Coulomb model reduces to the pressure-independent Tresca model with a perfectly hexagonal deviatoric section. In the case of $\phi = 90°$ the Mohr-Coulomb model reduces to the "tension cutoff" Rankine model with a triangular deviatoric section and $R_{mc} = \infty$ (this limiting case is not permitted within the Mohr-Coulomb model described here).

When using one-element tests to verify the calibration of the model, the output variables SP1, SP2, and SP3 correspond to the principal stresses σ_3, σ_2, and σ_1, respectively.

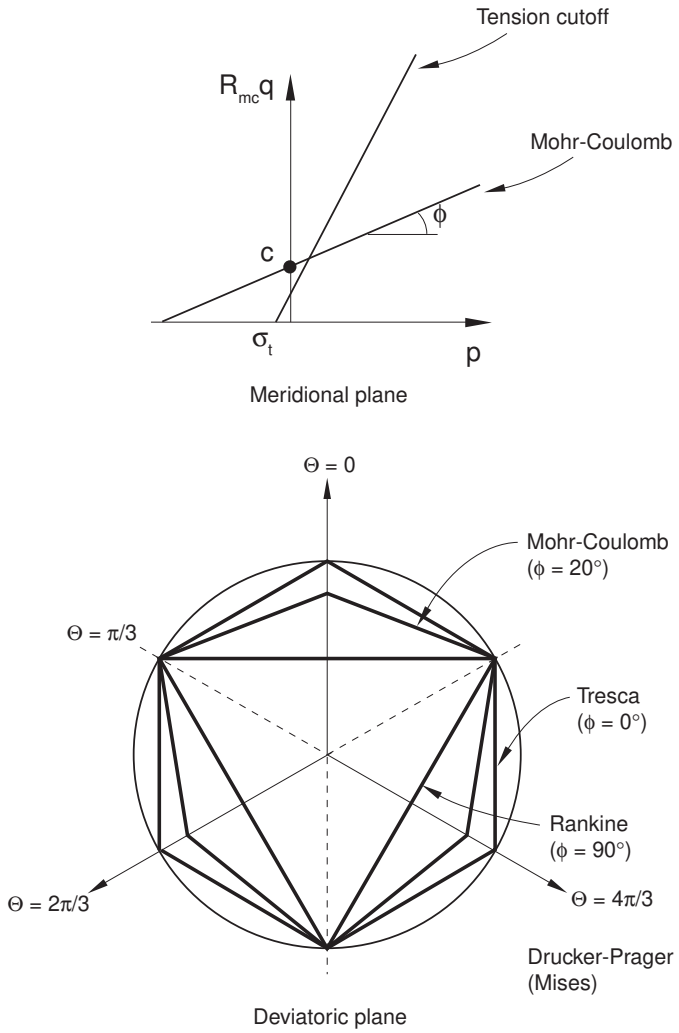


Figure 2: Mohr-Coulomb and tension cutoff surfaces in meridional and deviatoric planes.

Isotropic cohesion hardening is assumed for the hardening behavior of the Mohr-Coulomb yield surface. The hardening curve must describe the cohesion yield stress as a function of plastic strain and, possibly, temperature and predefined field variables. In defining this dependence at finite strains, “true” (Cauchy) stress and logarithmic strain values should be given. An optional tension cutoff hardening (or softening) curve can be specified

Rate dependency effects are not accounted for in this plasticity model.

Input File Usage: Use the following options to specify the Mohr-Coulomb yield surface and cohesion hardening:

**MOHR COULOMB*

**MOHR COULOMB HARDENING*

Abaqus/CAE Usage: Use the following options to specify the Mohr-Coulomb yield surface and cohesion hardening:

Property module: material editor: **Mechanical->Plasticity->Mohr Coulomb Plasticity**

Property module: material editor: **Mechanical**->**Plasticity**->**Mohr Coulomb Plasticity**: **Cohesion**

Rankine Surface

In Abaqus tension cutoff is modeled using the Rankine surface, which is written as

$$F_t = R_r\left(\Theta\right) q - p - \sigma_t\left(\overline{\varepsilon_t}^{pl}\right) = 0,$$

where $R_r\left(\Theta\right) = \left(2/3\right)\cos\Theta$, and σ_t is the tension cutoff value representing softening (or hardening) of the Rankine surface, as a function of tensile equivalent plastic strain, $\overline{\varepsilon_t}^{pl}$.

Input File Usage: Use the following option to specify hardening or softening for the Rankine surface:

**TENSION CUTOFF*

Abaqus/CAE Usage: Use the following option to specify hardening or softening for the Rankine surface:

Property module: material editor: **Mechanical**->**Plasticity**->**Mohr Coulomb Plasticity**: toggle on **Specify tension cutoff**; **Tension Cutoff**

Plastic Behavior: Flow Potentials

The flow potentials used for the Mohr-Coulomb yield surface and the tension cutoff surface are described below.

Plastic Flow on the Mohr-Coulomb Yield Surface

The flow potential, *G*, for the Mohr-Coulomb yield surface is chosen as a hyperbolic function in the meridional stress plane and the smooth elliptic function proposed by Menétrey and Willam (1995) in the deviatoric stress plane:

$$G = \sqrt{\left(\epsilon c|_0 \tan\psi\right)^2 + \left(R_{mw} q\right)^2} - p\tan\psi,$$

where

$$R_{mw}\left(\Theta, e\right) = \frac{4\left(1-e^2\right)\cos^2\Theta + \left(2e-1\right)^2}{2\left(1-e^2\right)\cos\Theta + \left(2e-1\right)\sqrt{4\left(1-e^2\right)\cos^2\Theta + 5e^2 - 4e}} R_{mc}\left(\frac{\pi}{3}, \phi\right),$$

and

$$R_{mc}\left(\frac{\pi}{3}, \phi\right) = \frac{3-\sin\phi}{6\cos\phi},$$

ψ

is the dilation angle measured in the p–$R_{mw}q$ plane at high confining pressure and can depend on temperature and predefined field variables;

$c|_0$

is the initial cohesion yield stress, $c|_0 = c|_{\bar{\varepsilon}^{pl}=0}$;

Θ

is the deviatoric polar angle defined previously;

ϵ

is a parameter, referred to as the meridional eccentricity, that defines the rate at which the hyperbolic function approaches the asymptote (the flow potential tends to a straight line in the meridional stress plane as the meridional eccentricity tends to zero); and

e

is a parameter, referred to as the deviatoric eccentricity, that describes the "out-of-roundedness" of the deviatoric section in terms of the ratio between the shear stress along the extension meridian ($\Theta = 0$) and the shear stress along the compression meridian ($\Theta = \frac{\pi}{3}$).

A default value of 0.1 is provided for the meridional eccentricity, ϵ.

By default, the deviatoric eccentricity, e, is calculated as

$$e = \frac{3-\sin\phi}{3+\sin\phi},$$

where ϕ is the Mohr-Coulomb friction angle; this calculation corresponds to matching the flow potential to the yield surface in both triaxial tension and compression in the deviatoric plane. Alternatively, Abaqus allows you to consider this deviatoric eccentricity as an independent material parameter; in this case you provide its value directly. Convexity and smoothness of the elliptic function requires that $1/2 < e \leq 1$. The upper limit, $e = 1$ (or $\phi = 0°$ when you do not specify the value of e), leads to $R_{mw}(\Theta, e = 1) = R_{mc}\left(\frac{\pi}{3}, \phi\right)$, which describes the Mises circle in the deviatoric plane. The lower limit, $e = 1/2$ (or $\phi = 90°$ when you do not specify the value of e), leads to $R_{mw}(\Theta, e = 1/2) = 2R_{mc}\left(\frac{\pi}{3}, \phi\right)\cos\Theta$ and would describe the Rankine triangle in the deviatoric plane (this limiting case is not permitted within the Mohr-Coulomb model described here).

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. A family of hyperbolic potentials in the meridional stress plane is shown in *Figure 3*, and the flow potential in the deviatoric stress plane is shown in *Figure 4*.

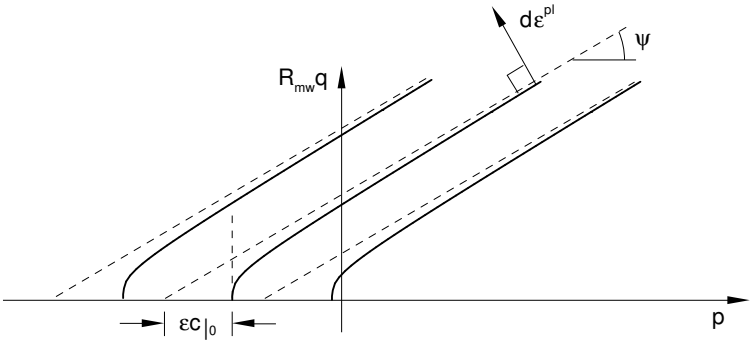


Figure 3: Family of hyperbolic flow potentials in the meridional stress plane.

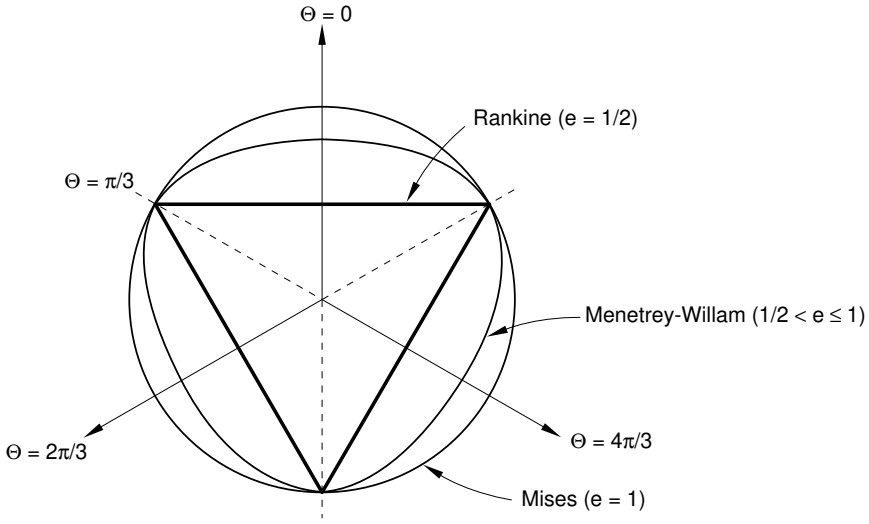


Figure 4: Menétrey-Willam flow potential in the deviatoric stress plane.

Flow in the meridional stress plane can be close to associated when the angle of friction, ϕ, and the angle of dilation, ψ, are equal and the meridional eccentricity, ϵ, is very small; however, flow in this plane is, in general, nonassociated. Flow in the deviatoric stress plane is always nonassociated.

Input File Usage: Use the following option to allow Abaqus to calculate the value of e (default):

**MOHR COULOMB*

Use the following option to specify the value of e directly:

**MOHR COULOMB*, DEVIATORIC ECCENTRICITY=e

Abaqus/CAE Usage: Use the following option to allow Abaqus to calculate the value of e (default):

Property module: material editor: **Mechanical->Plasticity->Mohr Coulomb Plasticity**: **Plasticity**: **Deviatoric eccentricity: Calculated default**

Use the following option to specify the value of e directly:

Property module: material editor: **Mechanical->Plasticity->Mohr Coulomb Plasticity**: **Plasticity**: **Deviatoric eccentricity: Specify:** e

Plastic Flow on the Rankine Surface

A flow potential that results in a nearly associative flow is chosen for the Rankine surface and is constructed by modifying the Menétrey-Willam potential described earlier:

$$G_t = \sqrt{(\epsilon_t \sigma_t|_0)^2 + (R_t q)^2} - p,$$

where

$$R_t\left(\Theta, e_t\right) = \frac{1}{3}\frac{4\left(1-e_t^2\right)\cos^2\Theta+\left(2e_t-1\right)^2}{2\left(1-e_t^2\right)\cos\Theta+\left(2e_t-1\right)\sqrt{4\left(1-e_t^2\right)\cos^2\Theta+5e_t^2-4e_t}},$$

$\sigma_t|_0$

is the initial value of tension cutoff;

ϵ_t

is the meridional eccentricity, similar to ϵ defined earlier; and

e_t

is the deviatoric eccentricity, similar to e defined earlier.

Abaqus uses values of 0.1 and 0.6, for ϵ_t and e_t, respectively.

Nonassociated Flow

Since the plastic flow is nonassociated in general, the use of this Mohr-Coulomb model generally requires the unsymmetric matrix storage and solution scheme in Abaqus/Standard (see *Defining an Analysis*).

Elements

The Mohr-Coulomb plasticity model can be used with any stress/displacement elements other than one-dimensional elements (beam, pipe, and truss elements) or elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables are available for the Mohr-Coulomb plasticity model:

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \int \frac{1}{c}\boldsymbol{\sigma} : d\boldsymbol{\varepsilon}^{pl}$, where c is the cohesion yield stress.

PEEQT

Tensile equivalent plastic strain, $\overline{\varepsilon_t}^{pl}$, on the tension cutoff yield surface.

References

- Menétrey, Ph., , and K. J. Willam, “Triaxial Failure Criterion for Concrete and its Generalization,” *ACI Structural Journal*, vol. 92, pp. 311–318, May/June 1995.

Critical State (Clay) Plasticity Model

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **CLAY PLASTICITY*
- **CLAY HARDENING*
- **POTENTIAL*
- **SOFTENING REGULARIZATION*
- *Defining clay plasticity*
- *Critical state models*

Overview

The clay plasticity model provided in Abaqus:

- describes the inelastic behavior of the material by a yield function that depends on the three stress invariants, an associated flow assumption to define the plastic strain rate, and a strain hardening theory that changes the size of the yield surface according to the inelastic volumetric strain;
- can have an isotropic or an anisotropic yield function;
- requires that the elastic part of the deformation be defined by using the isotropic or orthotropic linear elastic material model (*Linear Elastic Behavior*) or, in Abaqus/Standard, the porous elastic material model (*Elastic Behavior of Porous Materials*) within the same material definition (porous elasticity is supported only for isotropic yield functions);
- allows for the hardening law to be defined by a piecewise linear form or, in Abaqus/Standard, by an exponential form;
- Might optionally include hardening in hydrostatic tension; and
- can be used in conjunction with a regularization scheme for mitigating mesh dependence in situations where the material exhibits strain localization with increasing plastic deformation.

Yield Surface

The model is based on the yield surface

$$\frac{1}{\beta^2}\left(\frac{p-p_t}{a}-1\right)^2+\left(\frac{\tilde{t}}{Ma}\right)^2-1=0,$$

where

$p=-\frac{1}{3}\,\mathrm{trace}\,\boldsymbol{\sigma}$

is the equivalent pressure stress;

$\tilde{t} = \frac{1}{2}\tilde{q}\left[1 + \frac{1}{K} - \left(1 - \frac{1}{K}\right)\cos 3\Theta\right]$
is the deviatoric stress measure;

Θ
is the deviatoric polar angle defined as $\cos 3\Theta = \left(\frac{r}{q}\right)^3$;

$\tilde{q}$
is the Hill's potential, as defined in *Hill Anisotropic Yield/Creep*;

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$
is the Mises equivalent stress;

$r = \left(\frac{9}{2}\mathbf{S} : \mathbf{S} \cdot \mathbf{S}\right)^{\frac{1}{3}}$
is the third stress invariant;

M
is a constant that defines the slope of the critical state line;

β
is a constant that is equal to 1.0 on the "dry" side of the critical state line ($p < a + p_t$) but might be different from 1.0 on the "wet" side of the critical state line ($\beta \neq 1.0$ introduces a different ellipse on the wet side of the critical state line; that is, a tighter "cap" is obtained if $\beta < 1.0$ as shown in *Figure 1*);

$a = \frac{p_c - p_t}{(1+\beta)}$
is a measure of the size of the yield surface (*Figure 1*);

p_c
is the yield stress in hydrostatic compression;

p_t
is the yield stress in hydrostatic tension; and

K

is the ratio of the flow stress in triaxial tension to the flow stress in triaxial compression and determines the shape of the yield surface in the plane of principal deviatoric stresses (the "Π-plane": see *Figure 2*); Abaqus requires that $0.778 \leq K \leq 1.0$ to ensure that the yield surface remains convex.

The user-defined parameters M, β, and K can depend on temperature θ as well as other predefined field variables, f_i. For the isotropic model, the expression for $\tilde{q}$ reduces to the Mises equivalent stress, q. The model is described in detail in *Critical state models*.

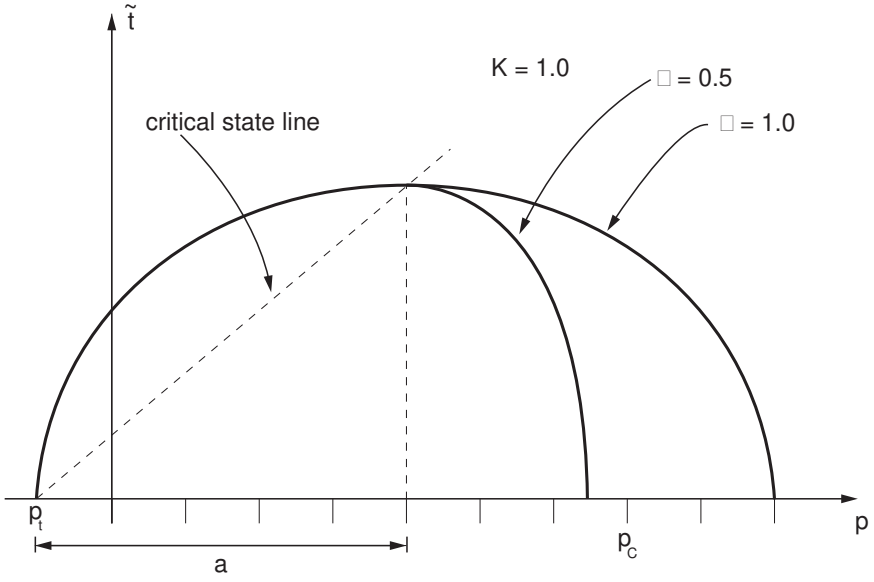


Figure 1: Clay yield surfaces in the p–$\tilde{t}$ plane.

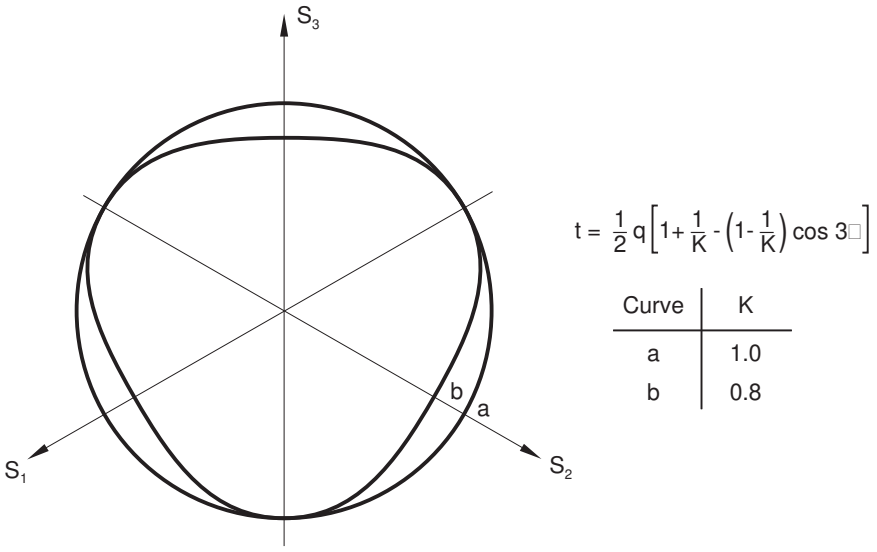


Figure 2: Isotropic clay yield surface sections in the Π-plane ($\tilde{t} = t$ and $\tilde{q} = q$ for the isotropic yield function).

Input File Usage: Use the following option to define the isotropic model:

**CLAY PLASTICITY*

Use the following options to define the orthotropic model:

**CLAY PLASTICITY*
**POTENTIAL*

Abaqus/CAE Usage: Use the following option to define the isotropic model:

Property module: material editor: **Mechanical->Plasticity->Clay Plasticity**

Use the following options to define the orthotropic model:

Mechanical->Plasticity->Clay Plasticity: **Suboptions->Potential**

Hardening Law

The hardening law can have an exponential form (Abaqus/Standard only) or a piecewise linear form.

Exponential Form in Abaqus/Standard

The exponential form of the hardening law can be used only in conjunction with the Abaqus/Standard porous elastic material model and the isotropic form of the yield surface with $p_t = 0$. The size of the yield surface at any time is determined by the initial value of the hardening parameter, a_0, and the amount of inelastic volume change that occurs according to the equation

$$a = a_0 \exp\left[(1 + e_0)\,\frac{1 - J^{pl}}{\lambda - \kappa J^{pl}}\right],$$

where

J^{pl}

is the inelastic volume change (that part of J, the ratio of current volume to initial volume, attributable to inelastic deformation);

$\kappa(\theta, f_i)$

is the logarithmic bulk modulus of the material defined for the porous elastic material behavior;

$\lambda(\theta, f_i)$

is the logarithmic hardening constant defined for the clay plasticity material behavior; and

e_0

is the user-defined initial void ratio (*Defining Initial Void Ratios in a Porous Medium*).

Specifying the Initial Size of the Yield Surface Directly

The initial size of the yield surface is defined for clay plasticity by specifying the hardening parameter, a_0, as a tabular function or by defining it analytically.

α_0 can be defined along with λ, M, β, and K, as a tabular function of temperature and other predefined field variables. However, α_0 is a function only of the initial conditions; it will not change if temperatures and field variables change during the analysis.

Input File Usage: Use all the following options:

**INITIAL CONDITIONS*, TYPE=RATIO
**POROUS ELASTIC*
**CLAY PLASTICITY*, HARDENING=EXPONENTIAL

Abaqus/CAE Usage: Use all the following options:

Property module: material editor:
Mechanical->Elasticity->Porous Elastic
Mechanical->Plasticity->Clay Plasticity: **Hardening: Exponential**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Specifying the Initial Size of the Yield Surface Indirectly

The hardening parameter α_0 can be defined indirectly by specifying e_1, which is the intercept of the virgin consolidation line with the void ratio axis in the plot of void ratio, e, versus the logarithm of the effective pressure stress, $\ln p$ (*Figure 3*).

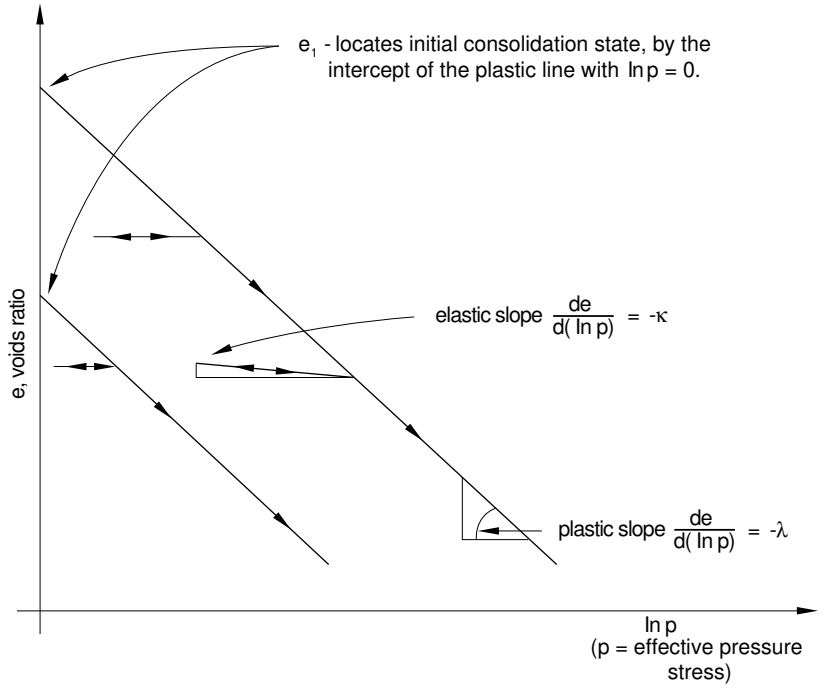


Figure 3: Pure compression behavior for clay model.

If this method is used, α_0 is defined by

$$\alpha_0 = \frac{1}{1+\beta} \exp\left(\frac{e_1 - e_0 - \kappa \ln p_0}{\lambda - \kappa}\right),$$

where p_0 is the user-defined initial value of the equivalent hydrostatic pressure stress (see *Defining Initial Stresses*). You define e_1, λ, M, β, and K; all the parameters except e_1 can be dependent on temperature and other predefined field variables. However, a_0 is a function only of the initial conditions; it will not change if temperatures and field variables change during the analysis.

Input File Usage: Use all the following options:

**INITIAL CONDITIONS*, TYPE=RATIO
**INITIAL CONDITIONS*, TYPE=STRESS
**POROUS ELASTIC*
**CLAY PLASTICITY*, HARDENING=EXPONENTIAL, INTERCEPT=e_1

Abaqus/CAE Usage: Use all the following options:

Property module: material editor:
Mechanical->Elasticity->Porous Elastic
Mechanical->Plasticity->Clay Plasticity: **Hardening: Exponential**, **Intercept:** e_1

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Void ratio** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Stress** for the **Types for Selected Step**

Piecewise Linear Form

If the piecewise linear form of the hardening rule is used, the user-defined relationship relates the yield stress in hydrostatic compression, p_c, and, optionally, the yield stress in hydrostatic tension, p_t, to the corresponding volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}$ (*Figure 4*):

$$p_c = p_c\left(\varepsilon_{\text{vol}}^{pl}\right),$$

$$p_t = p_t\left(\varepsilon_{\text{vol}}^{pl}\right).$$

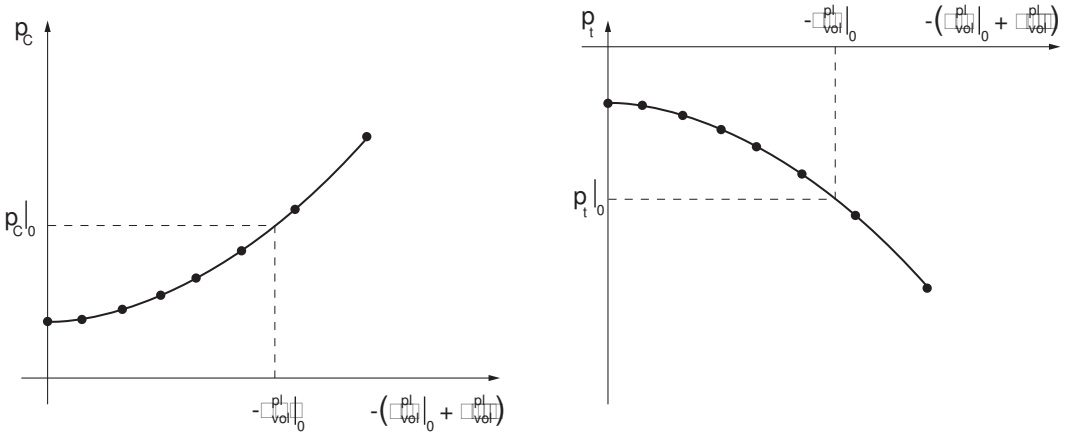


Figure 4: Typical piecewise linear clay hardening/softening curve.

The evolution parameter, a, is then given by

$$a = \frac{p_c - p_t}{(1+\beta)}.$$

The volumetric plastic strain axis has an arbitrary origin: $\varepsilon^{pl}_{\text{vol}}\big|_0$ is the position on this axis corresponding to the initial state of the material, thus defining the initial hydrostatic pressure in compression, $p_c|_0$, and, optionally, in tension, $p_t|_0$ and, hence, the initial yield surface size, a_0. This relationship is defined in tabular form as clay hardening data. The range of values for which p_c and p_t is defined should be sufficient to include all values of equivalent pressure stress to which the material will be subjected during the analysis. Data for p_c must be specified; data for p_t is optional.

This form of the hardening law can be used in conjunction with either the linear elastic or, in Abaqus/Standard, the porous elastic material models. This is the only form of the hardening law supported in Abaqus/Explicit.

Input File Usage: Use both of the following options to define the hardening behavior by providing the hydrostatic compression yield stress as a function of volumetric plastic strain:

**CLAY PLASTICITY*, HARDENING=TABULAR
**CLAY HARDENING*, TYPE=COMPRESSION (default)

Optionally, add the following option to define the tensile hardening behavior by providing the hydrostatic tension yield stress as a function of volumetric plastic strain:

**CLAY HARDENING*, TYPE=TENSION

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Clay Plasticity**: **Hardening: Tabular**, **Suboptions->Compressive Clay Hardening** and/or **Tensile Clay Hardening**

Softening Regularization

Granular materials often exhibit strain localization with increasing plastic deformation. Post-failure solutions from conventional finite element methods can be strongly mesh dependent. To mitigate the mesh dependency of the solutions, a regularization method is often used to introduce a micro-structural length scale into the constitutive formulation. Let $l_c^{(m)}$ denote the characteristic width of a shear band or a crack band, $l_c^{(e)}$ the characteristic length of the element, and $\varepsilon^{pl}_{\text{vol,e}}$ the inelastic strain for the element. Then the inelastic strain in the localization band, $\varepsilon^{pl}_{\text{vol,m}}$, is defined to be

$$\varepsilon^{pl}_{\text{vol,m}} = \varepsilon^{pl}_{\text{vol,e}} \min\left(\left(\frac{l_c^{(e)}}{l_c^{(m)}}\right)^{n_r}, f_{max}\right),$$

where n_r is a material parameter and f_{max} is a positive number used for bounding the magnitude of regularization. This strain regularization method is valid only when the characteristic length of the element is greater than the width of the localization band; that is, $l_c^{(e)} \geq l_c^{(m)}$.

If softening regularization is included, it is applied to all hardening data (tension and compression) by default. You can optionally turn off softening regularization for a specific type of hardening.

Input File Usage: Use the following options to include softening regularization:

*CLAY PLASTICITY
*CLAY HARDENING, SR=ON (default)
*SOFTENING REGULARIZATION

Use the following option to turn off softening regularization:

*CLAY HARDENING, SR=OFF

Abaqus/CAE Usage: Use the following options to include softening regularization:

Property module: material editor: **Mechanical**->**Plasticity**->**Clay Plasticity**: **Suboptions**->**Softening Regularization**

Calibration

At least two experiments are required to calibrate the simplest version of the Cam-clay model: a hydrostatic compression test (an oedometer test is also acceptable) and a triaxial compression test (more than one triaxial test is useful for a more accurate calibration).

Hydrostatic Compression Tests

The hydrostatic compression test is performed by pressurizing the sample equally in all directions. The applied pressure and the volume change are recorded.

The onset of yielding in the hydrostatic compression test immediately provides the initial position of the yield surface, a_0. The logarithmic bulk moduli, κ and λ, are determined from the hydrostatic compression experimental data by plotting the logarithm of pressure versus void ratio. The void ratio, e, is related to the measured volume change as

$$J = \exp\left(\varepsilon_{vol}\right) = \frac{1+e}{1+e_0}.$$

The slope of the line obtained for the elastic regime is $-\kappa$, and the slope in the inelastic range is $-\lambda$. For a valid model $\lambda > \kappa$.

Triaxial Tests

Triaxial compression experiments are performed using a standard triaxial machine where a fixed confining pressure is maintained while the differential stress is applied. Several tests covering the range of confining pressures of interest are usually performed. Again, the stress and strain in the direction of loading are recorded, together with the lateral strain so that the correct volume changes can be calibrated.

The triaxial compression tests allow the calibration of the yield parameters M and β. M is the ratio of the shear stress, q, to the pressure stress, p, at critical state and can be obtained from the stress values when the material has become perfectly plastic (critical state). β represents the curvature of the cap part of the yield surface and can be calibrated from a number of triaxial tests at high confining pressures (on the "wet" side of critical state). β must be between 0.0 and 1.0.

To calibrate the parameter K, which controls the yield dependence on the third stress invariant, experimental results obtained from a true triaxial (cubical) test are necessary. These results are generally not available, and you might have to guess (the value of K is generally between 0.8 and 1.0) or ignore this effect.

To calculate the yield stress in hydrostatic tension, you can plot the data obtained from the triaxial compression tests on the p–q plane and extend the curve obtained from fitting these experimental data to the pressure axis in the tensile region.

Unloading Measurements

Unloading measurements in hydrostatic and triaxial compression tests are useful to calibrate the elasticity, particularly in cases where the initial elastic region is not well defined. From these we can identify whether a constant shear modulus or a constant Poisson's ratio should be used and what their values are.

Initial Conditions

If an initial stress at a point is given (see *Defining Initial Stresses*) such that the stress point lies outside the initially defined yield surface, Abaqus will try to adjust the initial position of the surface to make the stress point lie on it and issue a warning. However, if the yield stress in hydrostatic tension, p_t, is zero and does not evolve with volumetric plastic strain and the stress point is such that the equivalent pressure stress, p, is negative, an error message will be issued and execution will be terminated.

The initial condition on volumetric plastic strain, $\varepsilon^{pl}_{\text{vol}}\big|_0$, can be defined in the definition of the clay plasticity model. Abaqus also allows a general method of specifying the initial plastic strain field on elements (see *Defining Initial Values of Plastic Strain*). The volumetric plastic strain is then calculated as

$$\varepsilon^{pl}_{\text{vol}} = \varepsilon^{pl}_{\text{vol}}\big|_0 - tr\left(\varepsilon^{pl}\right).$$

Elements

The clay plasticity model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variable has special meaning for material points in the clay plasticity model:

PEEQ Center of the yield surface, a.

Crushable Foam Plasticity Models

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- *Rate-Dependent Yield*
- **CRUSHABLE FOAM*
- **CRUSHABLE FOAM HARDENING*
- **RATE DEPENDENT*
- *Defining crushable foam plasticity*

Overview

The crushable foam plasticity models:

- are intended for the analysis of crushable foams that are typically used as energy absorption structures;
- can be used to model crushable materials other than foams (such as balsa wood);
- are used to model the enhanced ability of a foam material to deform in compression due to cell wall buckling processes (it is assumed that the resulting deformation is not recoverable instantaneously and can, thus, be idealized as being plastic for short duration events);
- can be used to model the difference between a foam material's compressive strength and its much smaller tensile bearing capacity resulting from cell wall breakage in tension;
- must be used in conjunction with the linear elastic material model (*Linear Elastic Behavior*);
- can be used when rate-dependent effects are important; and
- are intended to simulate material response under essentially monotonic loading.

Elastic and Plastic Behavior

The elastic part of the response is specified as described in *Linear Elastic Behavior*.

For the plastic part of the behavior, the yield surface is a Mises circle in the deviatoric stress plane and an ellipse in the meridional (p–q) stress plane. Two hardening models are available: the volumetric hardening model, where the point on the yield ellipse in the meridional plane that represents hydrostatic tension loading is fixed and the evolution of the yield surface is driven by the volumetric compacting plastic strain, and the isotropic hardening model, where the yield ellipse is centered at the origin in the p–q stress plane and evolves in a geometrically self-similar manner. This phenomenological isotropic model was originally developed for metallic foams by Deshpande and Fleck (2000).

The hardening curve must describe the uniaxial compression yield stress as a function of the corresponding plastic strain. In defining this dependence at finite strains, “true” (Cauchy) stress and logarithmic strain values should be given. Both models predict similar behavior for compression-dominated loading. However, for hydrostatic tension loading the volumetric hardening model assumes a perfectly plastic behavior, while the isotropic hardening model predicts the same behavior in both hydrostatic tension and hydrostatic compression.

Crushable Foam Model with Volumetric Hardening

The crushable foam model with volumetric hardening uses a yield surface with an elliptical dependence of deviatoric stress on pressure stress. It assumes that the evolution of the yield surface is controlled by the volumetric compacting plastic strain experienced by the material.

Yield Surface

The yield surface for the volumetric hardening model is defined as

$$F = \sqrt{q^2 + \alpha^2 (p - p_0)^2} - B = 0,$$

where

$p = -\frac{1}{3}\,\mathrm{trace}\,\boldsymbol{\sigma}$

is the pressure stress,

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$

is the Mises stress,

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$

is the deviatoric stress,

A

is the size of the (horizontal) p-axis of the yield ellipse,

$B = \alpha\ A = \alpha\ \frac{p_c + p_t}{2}$

is the size of the (vertical) q-axis of the yield ellipse,

$\alpha = B/A$

is the shape factor of the yield ellipse that defines the relative magnitude of the axes,

$p_0 = \frac{p_c - p_t}{2}$

is the center of the yield ellipse on the p-axis,

p_t

is the strength of the material in hydrostatic tension, and

p_c

is the yield stress in hydrostatic compression (p_c is always positive).

The yield surface represents the Mises circle in the deviatoric stress plane and is an ellipse on the meridional stress plane, as depicted in *Figure 1*.

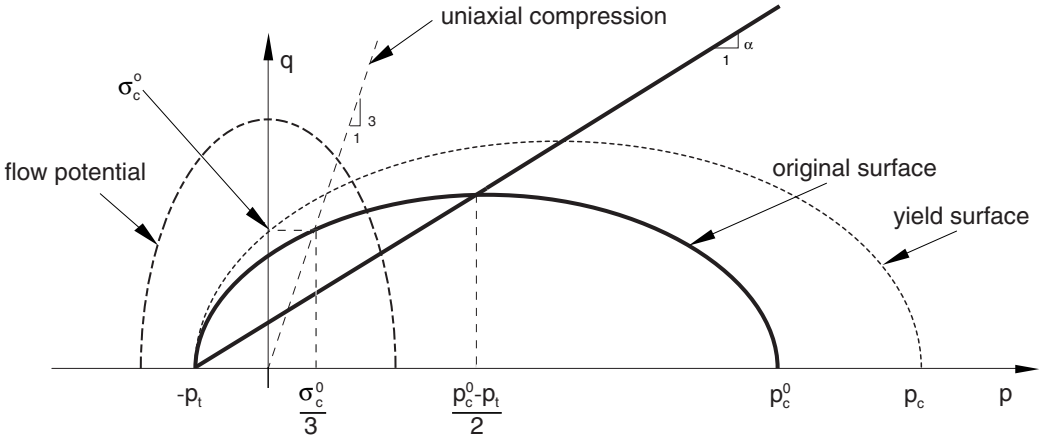


Figure 1: Crushable foam model with volumetric hardening: yield surface and flow potential in the *p–q* stress plane.

The yield surface evolves in a self-similar fashion (constant α); and the shape factor can be computed using the initial yield stress in uniaxial compression, σ_c^0, the initial yield stress in hydrostatic compression, p_c^0 (the initial value of p_c), and the yield strength in hydrostatic tension, p_t:

$$\alpha = \frac{3k}{\sqrt{(3k_t+k)(3-k)}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0} \quad \text{and} \quad k_t = \frac{p_t}{p_c^0}.$$

For a valid yield surface the choice of strength ratios must be such that $0 < k < 3$ and $k_t \geq 0$. If this is not the case, Abaqus issues an error message and terminates execution.

To define the shape of the yield surface, you provide the values of k and k_t. If desired, these variables can be defined as a tabular function of temperature and other predefined field variables. In this case the model requires that the hardening curve (described below) be also specified for the same values of temperature and predefined field variables.

Input File Usage: **CRUSHABLE FOAM*, HARDENING=VOLUMETRIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Hardening: Volumetric**

Calibration

To use this model, one needs to know the initial yield stress in uniaxial compression, σ_c^0; the initial yield stress in hydrostatic compression, p_c^0; and the yield strength in hydrostatic tension, p_t. Since foam materials are rarely tested in tension, it is usually necessary to guess the magnitude of the strength of the foam in hydrostatic tension, p_t. The choice of tensile strength should not have a strong effect on the numerical results unless the foam is

stressed in hydrostatic tension. A common approximation is to set p_t equal to 5% to 10% of the initial yield stress in hydrostatic compression p_c^0; thus, k_t= 0.05 to 0.10.

Flow Potential

The plastic strain rate for the volumetric hardening model is assumed to be

$$\dot{\boldsymbol{\varepsilon}}^{pl} = \dot{\bar{\varepsilon}}^{pl} \quad \frac{\partial G}{\partial \boldsymbol{\sigma}},$$

where G is the flow potential, chosen in this model as

$$G = \sqrt{q^2 + \tfrac{9}{2}p^2},$$

and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate defined as

$$\dot{\bar{\varepsilon}}^{pl} = \frac{\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{pl}}{G}.$$

The equivalent plastic strain rate is related to the rate of axial plastic strain, $\dot{\varepsilon}^{pl}_{\text{axial}}$, in uniaxial compression by

$$\dot{\bar{\varepsilon}}^{pl} = \sqrt{\tfrac{2}{3}} \quad \dot{\varepsilon}^{pl}_{\text{axial}}.$$

A geometrical representation of the flow potential in the p–q stress plane is shown in *Figure 1*. This potential gives a direction of flow that is identical to the stress direction for radial paths. This is motivated by simple laboratory experiments that suggest that loading in any principal direction causes insignificant deformation in the other directions. As a result, the plastic flow is nonassociative for the volumetric hardening model. For more details regarding plastic flow, see *About plasticity models*.

Nonassociated Flow

The nonassociated plastic flow rule makes the material stiffness matrix unsymmetric; therefore, the unsymmetric matrix storage and solution scheme should be used in Abaqus/Standard (see *Defining an Analysis*). Usage of this scheme is especially important when large regions of the model are expected to flow plastically.

Hardening

The yield surface intersects the p-axis at $-p_t$ and p_c. We assume that p_t remains fixed throughout any plastic deformation process. By contrast, the compressive strength, p_c, evolves as a result of compaction (increase in density) or dilation (reduction in density) of the material. The evolution of the yield surface can be expressed through the evolution of the yield surface size on the hydrostatic stress axis, $p_c + p_t$, as a function of the value of volumetric compacting plastic strain, $-\varepsilon^{pl}_{\text{vol}}$. With p_t constant, this relation can be obtained from user-provided uniaxial compression test data using

$$p_c\left(\varepsilon^{pl}_{\text{vol}}\right) = \frac{\sigma_c\left(\varepsilon^{pl}_{\text{axial}}\right)\left[\sigma_c\left(\varepsilon^{pl}_{\text{axial}}\right)\left(\frac{1}{\alpha^2}+\frac{1}{9}\right)+\frac{p_t}{3}\right]}{p_t+\frac{\sigma_c\left(\varepsilon^{pl}_{\text{axial}}\right)}{3}}$$

along with the fact that $\varepsilon^{pl}_{\text{axial}} = \varepsilon^{pl}_{\text{vol}}$ in uniaxial compression for the volumetric hardening model. Thus, you provide input to the hardening law by specifying, in the usual tabular form, only the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The table must start with a zero plastic strain (corresponding to the virgin state of the material), and the tabular entries must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. If desired, the yield stress can also be a function of temperature and other predefined

field variables. In this case the model requires that the values of the strength ratios k and k_t be also specified for the same values of temperature and predefined field variables.

Input File Usage: **CRUSHABLE FOAM HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**

Rate Dependence

As strain rates increase, many materials show an increase in the yield stress. For many crushable foam materials this increase in yield stress becomes important when the strain rates are in the range of 0.1–1 per second and can be very important if the strain rates are in the range of 10–100 per second, as commonly occurs in high-energy dynamic events.

Two methods for specifying strain-rate-dependent material behavior are available in Abaqus as discussed below. Both methods assume that the shapes of the hardening curves at different strain rates are similar, and either can be used in static or dynamic procedures. When rate dependence is included, the static stress-strain hardening behavior must be specified for the crushable foam as described above.

Overstress Power Law

You can specify a Cowper-Symonds overstress power law that defines strain rate dependence. This law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D(R-1)^n \qquad \text{for} \qquad R \geq 1,$$

with

$$R \equiv \frac{\overline{B}}{B},$$

where B is the size of the static yield surface and $\overline{B}$ is the size of the yield surface at a nonzero strain rate. The ratio R can be written as

$$R - 1 = (r-1) \quad \frac{3k_t + r[k + k_t(3-k)]}{(1+k_t)(3k_t + r \quad k)},$$

where r is the uniaxial compression yield stress ratio defined by

$$r \equiv \frac{\overline{\sigma}_c}{\sigma_c}.$$

σ_c, specified as part of the crushable foam hardening definition, is the uniaxial compression yield stress at a given value of $\varepsilon^{pl}_{\text{axial}}$ for the experiment with the lowest strain rate and can depend on temperature and predefined field variables; D and n are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

The power-law rate dependency can be rewritten in the following form

$$\ln(R-1) = \frac{1}{n} \quad \ln \dot{\bar{\varepsilon}}^{pl} - \frac{1}{n} \quad \ln D.$$

The procedure outlined below can be followed to obtain the material parameters D and n based on the uniaxial compression test data.

1. Compute R using the uniaxial compression yield stress ratio, r.
2. Convert the rate of the axial plastic strain, $\dot{\varepsilon}^{pl}_{\text{axial}}$, to the corresponding equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$.

3. Plot $\ln(R-1)$ versus $\ln\left(\dot{\bar{\varepsilon}}^{pl}\right)$. If the curve can be approximated by a straight line such as that shown in *Figure 2*, the overstress power law is suitable. The slope of the line is $1/n$, and the intercept of the $\ln(R-1)$ axis is $-(1/n)\ln D$.

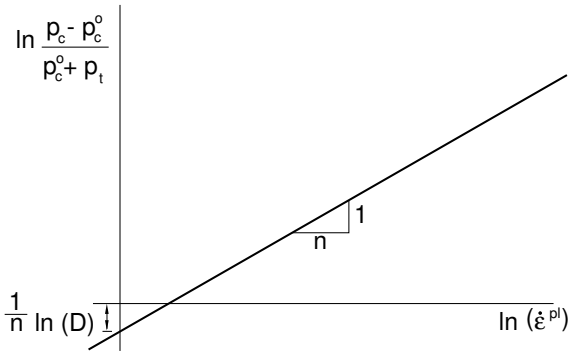


Figure 2: Calibration of overstress power law data.

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**; **Suboptions->Rate Dependent**: **Hardening: Power Law**

Chaboche Rate Dependence

Chaboche rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left\langle \frac{\bar{B}-B}{K} \right\rangle^n,$$

where $\dot{\varepsilon}_0(\theta, f_i)$, $K(\theta, f_i)$, and $n(\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

The above relation can be rewritten as

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left(\frac{B}{K}\right)^n \left(R-1\right)^n.$$

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=CHABOCHE

Abaqus/CAE Usage: Chaboche rate dependence is not supported in Abaqus/CAE.

Tabular Input of Yield Ratio

Rate-dependent behavior can alternatively be specified by giving a table of the ratio $r = \overline{\sigma}_c/\sigma_c$ as a function of the absolute value of the rate of the axial plastic strain and, optionally, temperature and predefined field variables.

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**; **Suboptions->Rate Dependent**: **Hardening: Yield Ratio**

Initial Conditions

When we need to study the behavior of a material that has already been subjected to some hardening, Abaqus allows you to prescribe initial conditions for the volumetric compacting plastic strain, $-\varepsilon_{\text{vol}}^{pl}$ (see *Defining Initial Values of State Variables for Plastic Hardening*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Crushable Foam Model with Isotropic Hardening

The isotropic hardening model uses a yield surface that is an ellipse centered at the origin in the *p*–*q* stress plane. The yield surface evolves in a self-similar manner, and the evolution is governed by the equivalent plastic strain (to be defined later).

Yield Surface

The yield surface for the isotropic hardening model is defined as

$$F = \sqrt{q^2 + \alpha^2 p^2} - B = 0,$$

where

$p = -\frac{1}{3}\,\text{trace}\,\boldsymbol{\sigma}$

is the pressure stress,

$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$

is the Mises stress,

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$

is the deviatoric stress,

$B = \alpha p_c = \sigma_c \sqrt{1 + \left(\frac{\alpha}{3}\right)^2}$

is the size of the (vertical) q-axis of the yield ellipse,

α

is the shape factor of the yield ellipse that defines the relative magnitude of the axes,

p_c

is the yield stress in hydrostatic compression, and

σ_c

is the absolute value of the yield stress in uniaxial compression.

The yield surface represents the Mises circle in the deviatoric stress plane. The shape of the yield surface in the meridional stress plane is depicted in *Figure 3*. The shape factor, α, can be computed using the initial yield stress in uniaxial compression, σ_c^0, and the initial yield stress in hydrostatic compression, p_c^0 (the initial value of p_c), using the relation:

$$\alpha = \frac{3k}{\sqrt{9-k^2}} \qquad \text{with} \qquad k = \frac{\sigma_c^0}{p_c^0}.$$

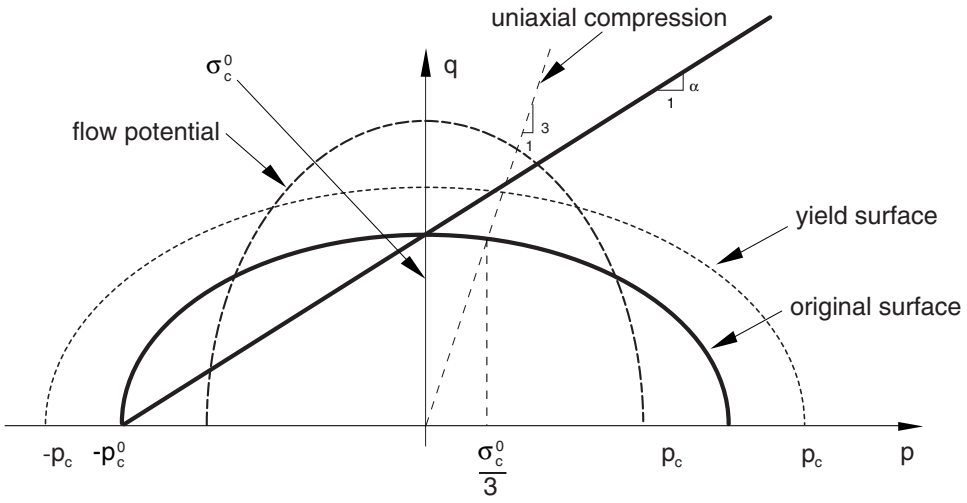


Figure 3: Crushable foam model with isotropic hardening: yield surface and flow potential in the p–q stress plane.

To define the shape of the yield ellipse, you provide the value of k. For a valid yield surface the strength ratio must be such that $0 \le k < 3$. The particular case of $k = 0$ corresponds to the Mises plasticity. In general, the initial yield strengths in uniaxial compression and in hydrostatic compression, σ_c^0 and p_c^0, can be used to calculate the value of k. However, in many practical cases the stress versus strain response curves of crushable foam materials do not show clear yielding points, and the initial yield stress values cannot be determined exactly. Many of these response curves have a horizontal plateau—the yield stress is nearly a constant for a significantly

large range of plastic strain values. If you have data from both uniaxial compression and hydrostatic compression, the plateau values of the two experimental curves can be used to calculate the ratio of *k*.

Input File Usage: **CRUSHABLE FOAM*, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Hardening: Isotropic**

Flow Potential

The flow potential for the isotropic hardening model is chosen as

$$G = \sqrt{q^2 + \beta^2 p^2},$$

where β represents the shape of the flow potential ellipse on the *p*–*q* stress plane. It is related to the plastic Poisson's ratio, ν_p, via

$$\beta = \frac{3}{\sqrt{2}} \sqrt{\frac{1-2\nu_p}{1+\nu_p}}.$$

The plastic Poisson's ratio, which is the ratio of the transverse to the longitudinal plastic strain under uniaxial compression, must be in the range of −1 and 0.5; and the upper limit ($\nu_p = 0.5$) corresponds to the case of incompressible plastic flow ($\beta = 0$). For many low-density foams the plastic Poisson's ratio is nearly zero, which corresponds to a value of $\beta \approx 2.12$.

The plastic flow is associated when the value of α is the same as that of β. By default, the plastic flow is nonassociated to allow for the independent calibrations of the shape of the yield surface and the plastic Poisson's ratio. If you have information only about the plastic Poisson's ratio and choose to use associated plastic flow, the yield stress ratio *k* can be calculated from

$$k = \sqrt{3\,(1 - 2\quad \nu_p)}.$$

Alternatively, if only the shape of the yield surface is known and you choose to use associated plastic flow, the plastic Poisson's ratio can be obtained from

$$\nu_p = \frac{3-k^2}{6}.$$

You provide the value of ν_p.

Input File Usage: **CRUSHABLE FOAM*, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Hardening: Isotropic**

Hardening

A simple uniaxial compression test is sufficient to define the evolution of the yield surface. The hardening law defines the value of the yield stress in uniaxial compression as a function of the absolute value of the axial plastic strain. The piecewise linear relationship is entered in tabular form. The table must start with a zero plastic strain (corresponding to the virgin state of the materials), and the tabular entries must be given in ascending magnitude of $\varepsilon^{pl}_{\text{axial}}$. For values of plastic strain greater than the last user-specified value, the stress-strain relationship is

extrapolated based on the last slope computed from the data. If desired, the yield stress can also be a function of temperature and other predefined field variables.

Input File Usage: **CRUSHABLE FOAM HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**

Rate Dependence

As strain rates increase, many materials show an increase in the yield stress. For many crushable foam materials this increase in yield stress becomes important when the strain rates are in the range of 0.1–1 per second and can be very important if the strain rates are in the range of 10–100 per second, as commonly occurs in high-energy dynamic events.

Two methods for specifying strain-rate-dependent material behavior are available in Abaqus as discussed below. Both methods assume that the shapes of the hardening curves at different strain rates are similar, and either can be used in static or dynamic procedures. When rate dependence is included, the static stress-strain hardening behavior must be specified for the crushable foam as described above.

Overstress Power Law

You can specify a Cowper-Symonds overstress power law that defines strain rate dependence. This law has the form

$$\dot{\bar{\varepsilon}}^{pl} = D(R-1)^n \qquad \text{for} \qquad R \geq 1,$$

with

$$R \equiv \frac{\bar{\sigma}_c}{\sigma_c},$$

where σ_c, specified as part of the crushable foam hardening definition, is the static uniaxial compression yield stress at a given value of $\varepsilon^{pl}_{\text{axial}}$ for the experiment with the lowest strain rate, and $\bar{\sigma}_c$ is the yield stress at a nonzero strain rate. $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate, which is equal to the rate of the axial plastic strain in uniaxial compression for the isotropic hardening model.

The power-law rate dependency can be rewritten in the following form

$$\ln(R-1) = \frac{1}{n} \ln \dot{\bar{\varepsilon}}^{pl} - \frac{1}{n} \ln D.$$

Plot $\ln(R-1)$ versus $\ln\left(\dot{\varepsilon}^{pl}_{\text{axial}}\right)$. If the curve can be approximated by a straight line such as that shown in *Figure 2*, the overstress power law is suitable. The slope of the line is $1/n$, and the intercept of the $\ln(R-1)$ axis is $-(1/n)\ln D$. The material parameters D and n can be functions of temperature and, possibly, of other predefined field variables.

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**; **Suboptions->Rate Dependent**: **Hardening: Power Law**

Chaboche Rate Dependence

Chaboche rate dependence has the form

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left\langle \frac{\overline{\sigma_c} - \sigma_c}{K} \right\rangle^n,$$

where $\dot{\varepsilon}_0 (\theta, f_i)$, $K (\theta, f_i)$, and $n (\theta, f_i)$ are material parameters that can be functions of temperature and, possibly, of other predefined field variables.

The above relation can be rewritten as

$$\dot{\bar{\varepsilon}}^{pl} = \dot{\varepsilon}_0 \left(\frac{\sigma_c}{K} \right)^n \left(R - 1 \right)^n.$$

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=CHABOCHE

Abaqus/CAE Usage: Chaboche rate dependence is not supported in Abaqus/CAE.

Tabular Input of Yield Ratio

Rate-dependent behavior can alternatively be specified by giving a table of the ratio R as a function of the absolute value of the rate of the axial plastic strain and, optionally, temperature and predefined field variables.

Input File Usage: Use both of the following options:

**CRUSHABLE FOAM HARDENING*
**RATE DEPENDENT*, TYPE=YIELD RATIO

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Crushable Foam**: **Suboptions->Foam Hardening**; **Suboptions->Rate Dependent**: **Hardening: Yield Ratio**

Elements

The crushable foam plasticity model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements. This model cannot be used with elements for which the stress state is assumed to be planar (plane stress, shell, and membrane elements) or with beam, pipe, or truss elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variable has special meaning for the crushable foam plasticity model:

PEEQ For the volumetric hardening model, PEEQ is the volumetric compacting plastic strain defined as $-\varepsilon_{\text{vol}}^{pl}$. For the isotropic hardening model, PEEQ is the equivalent plastic strain defined as $\bar{\varepsilon}^{pl} \equiv \int \frac{\boldsymbol{\sigma}:d\boldsymbol{\varepsilon}^{pl}}{\sigma_c}$, where σ_c is the uniaxial compression yield stress.

The volumetric plastic strain, $\varepsilon_{\text{vol}}^{pl}$, is the trace of the plastic strain tensor; you can also calculate it as the sum of direct plastic strain components.

For the volumetric hardening model, the initial values of the volumetric compacting plastic strain can be specified for elements that use the crushable foam material model, as described above. The volumetric compacting plastic strain (output variable PEEQ) provided by Abaqus then contains the initial value of the volumetric compacting plastic strain plus any additional volumetric compacting plastic strain due to plastic straining during the analysis. However, the plastic strain tensor (output variable PE) contains only the amount of straining due to deformation during the analysis.

References

- Deshpande, V. S., and N. A. Fleck, "Isotropic Constitutive Model for Metallic Foams," *Journal of the Mechanics and Physics of Solids*, vol. 48, pp. 1253–1276, 2000.

Fabric Material Behavior

Products: Abaqus/Explicit

References:

- *About the Material Library*
- *Elastic Behavior*
- *VFABRIC*
- **FABRIC*
- **UNIAXIAL*
- **LOADING DATA*
- **UNLOADING DATA*
- **EXPANSION*
- **DENSITY*
- **INITIAL CONDITIONS*

Overview

The fabric material model:

- is anisotropic and nonlinear;
- is a phenomenological model that captures the mechanical response of a woven fabric made of yarns in the fill and the warp directions;
- is valid for materials that exhibit two “structural” directions that may not be orthogonal to each other with deformation;
- defines the local fabric stresses as a function of change in angle between the fibers (shear strain) and the nominal strains along the yarn directions;
- allows for the computation of local fabric stresses based on test data or through user subroutine *VFABRIC*, which can be used to define a complex constitutive model; and
- requires that geometric nonlinearity be accounted for during the analysis step (*General and Perturbation Procedures*), since it is intended for finite-strain applications.

The fabric material model defined based on test data:

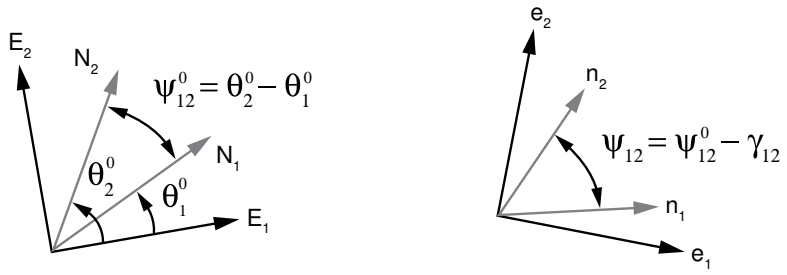
- assumes that the responses along the fill and the warp directions are independent of each other and that the shear response is independent of the direct response along the yarns;
- can include separate loading and unloading responses;
- can exhibit nonlinear elastic behavior, damaged elastic behavior, or elastic-plastic type behavior with permanent deformation upon complete unloading;
- can deform elastically to large tensile and shear strains; and
- can have properties that depend on temperature and/or other field variables.

Fabric Material Behavior

Woven fabrics are used in a number of engineering applications across various industries, including such products as automobile airbags; flexible structures like boat sails and parachutes; reinforcement in composites; architectural expressions in building roof structures; protective vests for military, police, and other security circles; and protective layers around the fuselage in planes.

Woven fabrics consist of yarns woven in the fill and the warp directions. The yarn is crimped, or curved, as it is woven up and down over the cross yarns. The nonlinear mechanical behavior of the fabric arises from different sources: the nonlinear response of the individual yarns, the exchange of crimp between the fill and the warp yarns as they are stretched, and the contact and friction between the yarns in cross directions and between the yarns in the same direction. In general, the fabric exhibits a significant stiffness only along the yarn directions under tension. The tensile response in the fill and warp directions may be coupled due to the crimp exchange mentioned above. Under in-plane shear deformation, the fill and warp direction yarns rotate with respect to each other. The resistance increases with shear deformation as lateral contact is formed between the yarns in each direction. The fabrics typically have negligible stiffness in bending and in-plane compression.

The behavior of woven fabrics is modeled phenomenologically in Abaqus/Explicit to capture the nonlinear anisotropic behavior of the fabric. The planar kinematic state of a given fabric is described in terms of the nominal direct strains in the fabric plane along the fill and the warp directions and the angle between the two yarn directions. The material orthogonal basis and the yarn local directions are illustrated in *Figure 1* showing the reference and the deformed configurations.



(a) Reference configuration

(b) Deformed configuration

Figure 1: Fabric kinematics

The engineering nominal shear strain, γ_{12}, is defined as the change in angle, ψ_{12} between the two yarn directions going from the reference to the deformed configuration. The nominal strains along the yarn directions $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ in the deformed configuration are computed from the respective yarn stretch values, $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$. The corresponding nominal stress components $\boldsymbol{T_{11}}, \boldsymbol{T_{22}}$, and $\boldsymbol{T_{12}}$ are defined as the work conjugate of the above nominal strains. The fabric nominal stress, $\mathbf{T}$, is converted by Abaqus to the Cauchy stress, $\boldsymbol{\sigma}$; and the subsequent internal forces arising from the fabric deformation are computed. You can obtain output of the fabric nominal strains, the fabric nominal stresses, and the regular Cauchy stresses. The relationship between the Cauchy stress, $\boldsymbol{\sigma}$, and the nominal stress, $\mathbf{T}$, is

$$J\sigma = \lambda_1 T_{11}\mathbf{n_1}\mathbf{n_1} + \lambda_2 T_{22}\mathbf{n_2}\mathbf{n_2} + T_{12}\csc(\psi_{12})\left(\mathbf{n_1}\mathbf{n_2} + \mathbf{n_2}\mathbf{n_1}\right) - T_{12}\cot(\psi_{12})\left(\mathbf{n_1}\mathbf{n_1} + \mathbf{n_2}\mathbf{n_2}\right),$$

where $\boldsymbol{J}$ is the volumetric Jacobian.

Either experimental data or a user subroutine, *VFABRIC*, can be used to characterize the Abaqus/Explicit fabric material model, providing the nominal fabric stress as a function of the nominal fabric strains. The user subroutine allows for building a complex material model taking into account both the fabric structural parameters such as yarn spacing, yarn cross-section shape, etc. and the yarn material properties. The test data–based fabric model

makes some simplifying assumptions but allows for nonlinear response including energy loss. The two models are presented below in detail. Both models capture the wrinkling of fabric under compression only in a smeared fashion.

The application of fabric material in a crash simulation is illustrated in *Side curtain airbag impactor test*.

Test Data–Based Fabric Materials

The fabric material model based on test data assumes that the responses along the fill and the warp directions are independent of each other and that the shear response is independent of the direct response along the yarn. Hence, each component-wise fabric stress response depends only on the fabric strain in that component. Thus, the overall behavior of the fabric consists of three independent component-wise responses: namely, the direct response along the fill yarn to the nominal strain in the fill yarn, the direct response along the warp yarn to the nominal strain in the warp yarn, and the shear response to the change in angle between the two yarns.

Within each component you must provide test data defining the response of the fabric. To fully define the fabric response, the test data must cover all of the following attributes:

- Within a component, separate test data can be defined for the fabric response in the tensile direction and in the compressive direction.
- Within a deformation direction (tension or compression), both loading and unloading test data can be provided.
- The loading and unloading test data can be classified according to three available behavior types: nonlinear elastic behavior, damaged elastic behavior, or elastic-plastic type behavior with permanent deformation. The behavior type determines how the fabric transitions from its loading response to its unloading response.

When elastic, the test data in a particular component can also be rate dependent. When separate loading and unloading paths are required, the test data for the two deformation directions (tension and compression) must be given separately. Otherwise, the data for both tension and compression may be given in a single table.

Input File Usage: Use the following options to define a fabric material using test data:

**FABRIC*
**UNIAXIAL*, COMPONENT=*component*
**LOADING DATA*, DIRECTION=*deformation direction*,
TYPE=*behavior type*
data lines to define loading data
**UNLOADING DATA*
data lines to define unloading data

Repeat all of the options underneath **FABRIC* as often as necessary to account for each component and deformation direction.

Specifying Uniaxial Behavior in a Component Direction

Independent loading and unloading test data can be provided in each of the three component directions. The components correspond to the response along the fill yarn, the response along the warp yarn, and the shear response.

Input File Usage: Use the following option to define the response along the fill yarn direction:

**UNIAXIAL*, COMPONENT=1

Use the following option to define the response along the warp yarn direction:

**UNIAXIAL*, COMPONENT=2

Use the following option to define the shear response:

**UNIAXIAL*, COMPONENT=SHEAR

Defining the Deformation Direction

The test data can be defined separately for tension and compression by specifying the deformation direction. If the deformation direction is defined (tension or compression), the tabular values defining tensile or compressive behavior should be specified with positive values of the stress and strain in the specified component and the loading data must start at the origin. If the behavior is not defined in a loading direction, the stress response will be zero in that direction (the fabric has no resistance in that direction).

If the deformation direction is not defined, the data apply to both tension and compression. However, the behavior is then considered to be nonlinear elastic and no unloading response can be specified. The test data will be considered to be symmetric about the origin if either tensile or compressive data are omitted.

Input File Usage: Use the following option to define tensile behavior:

**LOADING DATA*, DIRECTION=TENSION

Use the following option to define compressive behavior:

**LOADING DATA*, DIRECTION=COMPRESSION

Use the following option to define both tensile and compressive behavior in a single table:

**LOADING DATA*

Compressive Behavior

In general, a fabric material does not have significant stiffness under compression. To prevent the collapse of wrinkled elements under compressive loading, the specified stress-strain curve should have nonzero compressive stiffness, particularly at larger compressive strains.

Defining Loading/Unloading Component-Wise Response for a Fabric Material

To define the loading response, you specify the fabric stress as nonlinear functions of the fabric strain. This function can also depend on temperature and field variables. See *Input Syntax Rules* for further information about defining data as functions of temperature and field variables.

The unloading response can be defined in the following different ways:

- You can specify several unloading curves that express the fabric stress as nonlinear functions of the fabric strain; Abaqus interpolates these curves to create an unloading curve that passes through the point of unloading in an analysis.
- You can specify an energy dissipation factor (and a permanent deformation factor for models with permanent deformation), from which Abaqus calculates a quadratic unloading function.
- You can specify an energy dissipation factor (and a permanent deformation factor for models with permanent deformation), from which Abaqus calculates an exponential unloading function.
- You can specify the fabric stress as a nonlinear function of the fabric strain, as well as a transition slope; the fabric unloads along the specified transition slope until it intersects the specified unloading function, at which point it unloads according to the function. (This unloading definition is referred to as combined unloading.)

- You can specify the fabric stress as a nonlinear function of the fabric strain; Abaqus shifts the specified unloading function along the strain axis so that it passes through the point of unloading in an analysis.

The behavior type that is specified for the fabric dictates the type of unloading you can define, as summarized in *Table 1*. The different behavior types, as well as the associated loading and unloading curves, are discussed in more detail in the sections that follow.

Table 1: Available unloading definitions for the fabric behavior types.

Material behavior type	Unloading definition				
	Interpolated	Quadratic	Exponential	Combined	Shifted
Nonlinear elastic (rate-dependent only)	√				
Damaged elastic	√	√	√	√	
Permanent deformation	√	√	√		√

Defining Nonlinear Elastic Behavior

The elastic behavior can be nonlinear and, optionally, rate dependent. When the loading response is rate dependent, a separate unloading curve can also be specified. However, the unloading response need not be rate dependent.

Defining Rate-Independent Elasticity

When the loading response is rate independent, the unloading response is also rate independent and occurs along the same user-specified loading curve as illustrated in *Figure 2*. An unloading curve does not need to be specified.

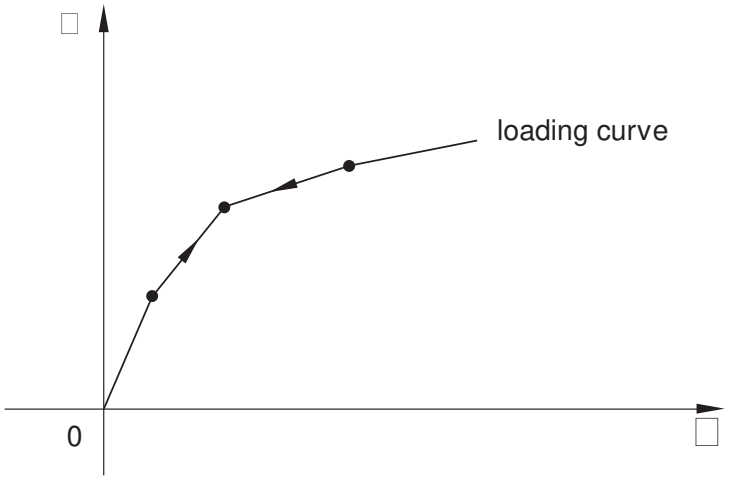


Figure 2: Nonlinear elastic rate-independent loading.

Input File Usage: **LOADING DATA*, TYPE=ELASTIC

Defining Rate-Dependent Elasticity

When the elastic response is rate dependent, both the loading and the unloading curves can be specified. The unloading data can be rate independent or rate dependent and must be specified in a single table; that is, only a single unloading data definition is allowed. If the unloading data are not specified, the unloading occurs along the loading curve specified with the smallest rate of deformation.

Unphysical jumps in the stress due to sudden changes in the rate of deformation are prevented using a technique based on viscoplastic regularization. This technique also helps model relaxation effects in a very simplistic manner, with the relaxation time given as $\tau = \mu_0 + \mu_1|\lambda - 1|^{\alpha}$ where μ_0, μ_1, and α are material parameters and λ is the stretch. μ_0 is a linear viscosity parameter that controls the relaxation time when $\lambda \approx 1$. Small values of this parameter should be used; a suggested value is 0.0001s. μ_1 is a nonlinear viscosity parameter that controls the relaxation time at higher values of λ. The smaller this value, the shorter the relaxation time. The suggested value for this parameter is 0.005s. α controls the sensitivity of the relaxation speed to the fabric strain component.

Figure 3 illustrates the loading/unloading behavior as the component is loaded at a rate $\dot{\varepsilon}_2^l$ and then unloaded at a rate $\dot{\varepsilon}_2^u$.

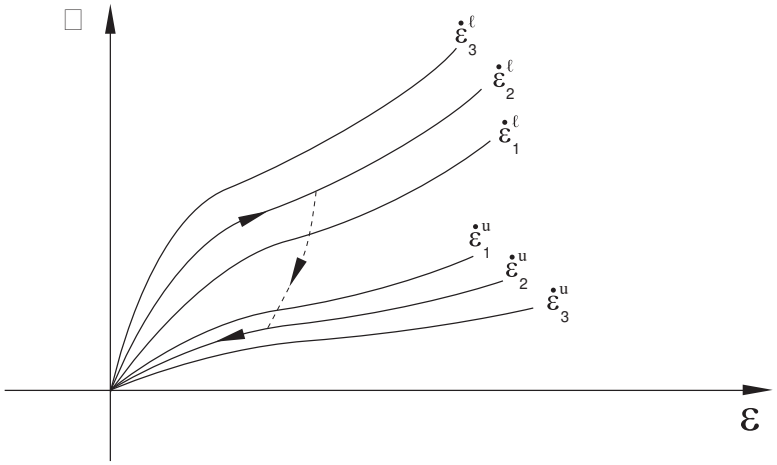


Figure 3: Rate-dependent loading/unloading.

The unloading path is determined by interpolating the specified unloading curves. The unloading need not be rate dependent, even though the loading response is rate dependent. When the unloading is rate dependent, the unloading path at any given component strain and strain rate is determined by interpolating the specified unloading curves.

Input File Usage: Use the following options when the unloading is also rate dependent:

**LOADING DATA*, TYPE=ELASTIC, RATE DEPENDENT, DIRECTION
**UNLOADING DATA*, DEFINITION=INTERPOLATED CURVE, RATE DEPENDENT

Use the following options when the unloading is rate independent:

**LOADING DATA*, TYPE=ELASTIC, RATE DEPENDENT, DIRECTION
**UNLOADING DATA*, DEFINITION=INTERPOLATED CURVE

Defining Models with Damage

The damage models dissipate energy upon unloading, and there is no permanent deformation upon complete unloading. You can specify the onset of damage by defining the strain above which the material response in unloading does not retrace the loading curve.

The unloading behavior controls the amount of energy dissipated by damage mechanisms and can be specified in one of the following ways:

- an analytical unloading curve (exponential/quadratic);
- an unloading curve interpolated from multiple user-specified unloading curves; or
- unloading along a transition unloading curve (constant slope specified by user) to the user-specified unloading curve (combined unloading).

Input File Usage: Use the following options to define damage with quadratic unloading behavior:

**LOADING DATA*, TYPE=DAMAGE, DIRECTION
**UNLOADING DATA*, DEFINITION=QUADRATIC

Use the following options to define damage with exponential unloading behavior:

**LOADING DATA*, TYPE=DAMAGE, DIRECTION
**UNLOADING DATA*, DEFINITION=EXPONENTIAL

Use the following options to define damage with an interpolated unloading curve:

**LOADING DATA*, TYPE=DAMAGE, DIRECTION
**UNLOADING DATA*, DEFINITION=INTERPOLATED CURVE

Use the following options to specify damage with combined unloading behavior:

**LOADING DATA*, TYPE=DAMAGE, DIRECTION
**UNLOADING DATA*, DEFINITION=COMBINED

Defining Onset of Damage

You can specify the onset of damage by defining the strain above which the material response in unloading does not retrace the loading curve.

Input File Usage: **LOADING DATA*, TYPE=DAMAGE, DAMAGE ONSET=*value*

Specifying Exponential/Quadratic Unloading

The damage model in *Figure 4* is based on an analytical unloading curve that is derived from an energy dissipation factor, H (fraction of energy that is dissipated at any strain level). As the fabric component is loaded, the stress follows the path given by the loading curve. If the fabric component is unloaded (for example, at point B), the stress follows the unloading curve BCO. Reloading after unloading follows the unloading curve OCB until the loading is such that the strain becomes greater than ε_B^{max} , after which the loading path follows the loading curve. The arrows shown in *Figure 4* illustrate the loading/unloading paths of this model.

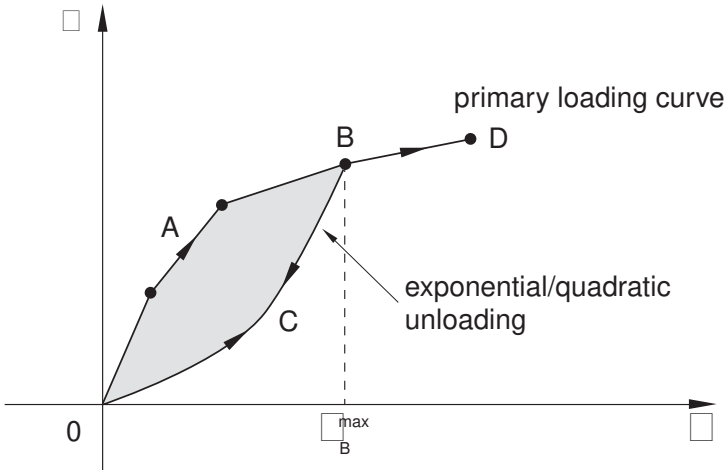


Figure 4: Exponential/quadratic unloading.

The unloading response follows the loading curve when the calculated unloading curve lies above the loading curve to prevent energy generation and follows a zero stress response when the unloading curve yields a negative response. In such cases the dissipated energy will be less than the value specified by the energy dissipation factor.

Specifying Interpolated Curve Unloading

The damage model in *Figure 5* illustrates an interpolated unloading response based on multiple unloading curves that intersect the primary loading curve at increasing values of stress/strains. You can specify as many unloading curves as are necessary to define the unloading response. Each unloading curve always starts at point *O*, the point of zero stress and zero strains, since the damage models do not allow any permanent deformation. The unloading curves are stored in normalized form so that they intersect the loading curve at a unit stress for a unit strain, and the interpolation occurs between these normalized curves. If unloading occurs from a maximum strain for which an unloading curve is not specified, the unloading is interpolated from neighboring unloading curves. As the fabric component is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example, at point *B*), the stress follows the unloading curve BCO. Reloading after unloading follows the unloading path OCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

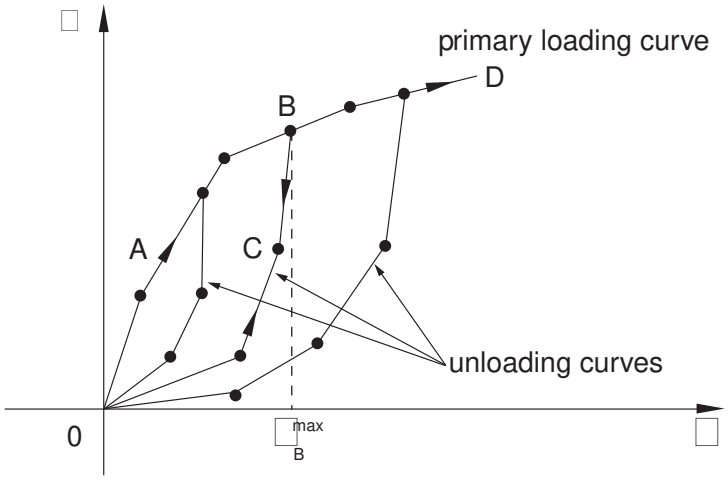


Figure 5: Interpolated curve unloading.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Specifying Combined Unloading

As illustrated in *Figure 6*, you can specify an unloading curve OCE in addition to the loading curve $OABD$ as well as a constant transition slope that connects the loading curve to the unloading curve. As the fabric is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example at point B) the stress follows the unloading curve BCO. The path BC is defined by the constant transition slope, and CO lies on the specified unloading curve. Reloading after unloading follows the unloading path OCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

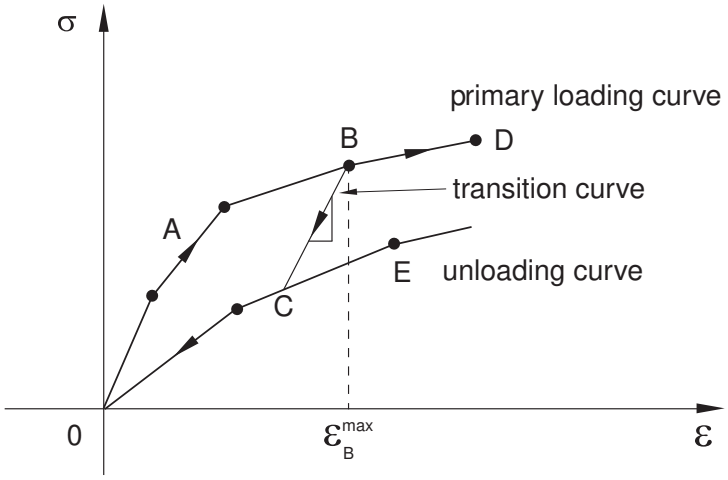


Figure 6: Combined unloading.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Defining Models with Permanent Deformation

These models dissipate energy upon unloading and exhibit permanent deformation upon complete unloading. You can specify the onset of permanent deformation by defining the strain below which unloading occurs along the loading curve.

The unloading behavior controls the amount of energy dissipated as well as the amount of permanent deformation. The unloading behavior can be specified in one of the following ways:

- an analytical unloading curve (exponential/quadratic);
- an unloading curve interpolated from multiple user-specified unloading curves; or
- an unloading curve obtained by shifting the user-specified unloading curve to the point of unloading.

Input File Usage: Use the following options to define permanent deformation with quadratic unloading behavior:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION,
DIRECTION
**UNLOADING DATA*, DEFINITION=QUADRATIC

Use the following options to define permanent damage with exponential unloading behavior:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION,
DIRECTION
**UNLOADING DATA*, DEFINITION=EXPONENTIAL

Use the following options to define permanent damage with an interpolated unloading curve:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION,
DIRECTION
**UNLOADING DATA*, DEFINITION=INTERPOLATED CURVE

Use the following options to specify permanent damage with a shifted unloading curve:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION,
DIRECTION
**UNLOADING DATA*, DEFINITION=SHIFTED CURVE

Defining the Onset of Permanent Deformation

By default, the onset of yield will be obtained as soon as the slope of the loading curve decreases by 10% from the maximum slope recorded up to that point while traversing along the loading curve. To override the default method of determining the onset of yield, you can specify either a value for the decrease in slope of the loading curve other than the default value of 10% (slope drop = 0.1) or by defining the strain below which unloading occurs along the loading curve. If a slope drop is specified, the onset of yield will be obtained as soon as the slope of the loading curve decreases by the specified factor from the maximum slope recorded up to that point.

Input File Usage:

Use the following options to specify the onset of yield by defining the strain below which unloading occurs along the loading curve:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION, YIELD ONSET=*value*

Use the following options to specify the onset of yield by defining a slope drop for the loading curve:

**LOADING DATA*, TYPE=PERMANENT DEFORMATION, SLOPE DROP=*value*

Specifying Exponential/Quadratic Unloading

The model in *Figure 7* illustrates an analytical unloading curve that is derived from an energy dissipation factor, H (fraction of energy that is dissipated at any strain level), and a permanent deformation factor, D_p. As the fabric component is loaded, the fabric stress follows the path given by the loading curve. If the component is unloaded (for example, at point *B*), the stress follows the unloading curve BCD. The point *D* corresponds to the permanent deformation, $D_p \varepsilon_B^{max}$. Reloading after unloading follows the unloading curve DCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve. The arrows shown in *Figure 7* illustrate the loading/unloading paths of this model.

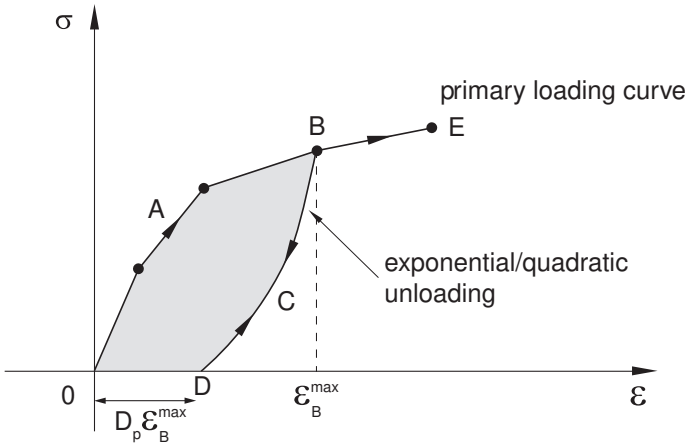


Figure 7: Exponential/quadratic unloading.

The unloading response follows the loading curve when the calculated unloading curve lies above the loading curve to prevent energy generation and follows a zero stress response when the unloading curve yields a negative response. In such cases the dissipated energy will be less than the value specified by the energy dissipation factor.

Specifying Interpolated Curve Unloading

The model in *Figure 8* illustrates an interpolated unloading response based on multiple unloading curves that intersect the primary loading curve at increasing values of stresses/strains.

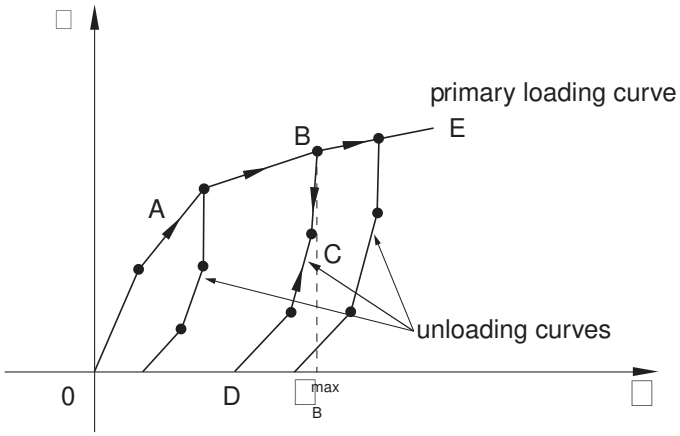


Figure 8: Interpolated curve unloading.

You can specify as many unloading curves as are necessary to define the unloading response. The first point of each unloading curve defines the permanent deformation if the fabric component is completely unloaded. The unloading curves are stored in normalized form so that they intersect the loading curve at a unit stress for a unit strain, and the interpolation occurs between these normalized curves. If unloading occurs from a maximum strain for which an unloading curve is not specified, the unloading is interpolated from neighboring unloading curves. As the fabric is loaded, the stress follows the path given by the loading curve. If the fabric is unloaded (for example, at point *B*), the stress follows the unloading curve BCD. Reloading after unloading follows the unloading path DCB until the loading is such that the strain becomes greater than ε_B^{max}, after which the loading path follows the loading curve.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Specifying Shifted Curve Unloading

You can specify an unloading curve passing through the origin in addition to the loading curve. The actual unloading curve is obtained by horizontally shifting the user-specified unloading curve to pass through the point of unloading as shown in *Figure 9*. The permanent deformation upon complete unloading is the horizontal shift applied to the unloading curve.

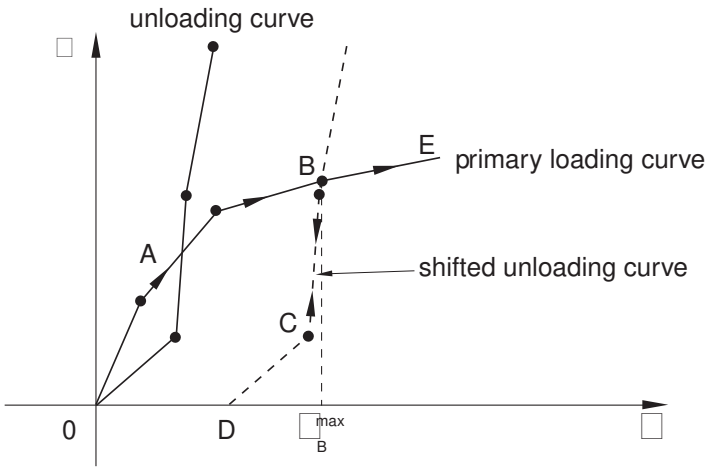


Figure 9: Shifted curve unloading.

The unloading curve also has the same temperature and field variable dependencies as the loading curve.

Using Different Uniaxial Models in Tension and Compression

When appropriate, different uniaxial behavior models can be used in tension and compression. For example, response under tension can be plastic with exponential unloading, while the response in compression can be nonlinear elastic (see *Figure 10*).

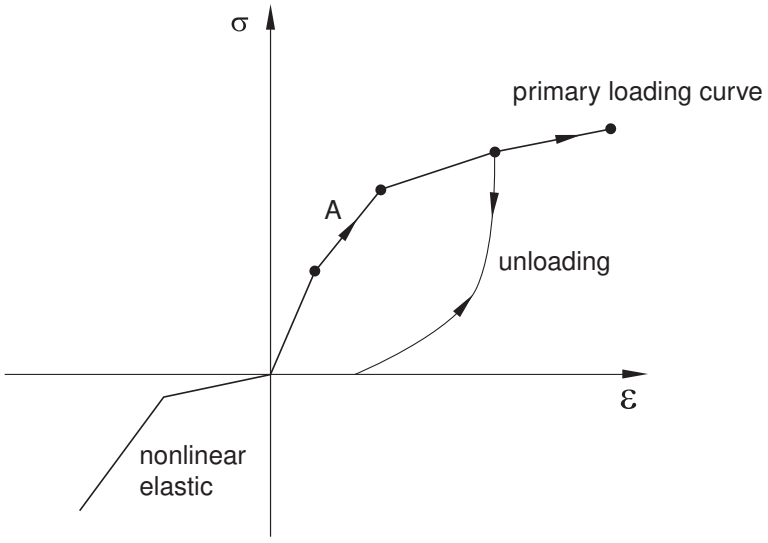


Figure 10: Different uniaxial models in tension and compression.

User-Defined Fabric Materials

The mechanical response of a fabric material depends on a number of micro and meso-scale parameters covering the fabric construction and that of the individual yarns as a bundle of fibers. Often a multiscale model becomes necessary to track the state of the fabric and its response to loading. Abaqus provides a specialized user subroutine, *VFABRIC*, to capture the complex fabric response given the deformed yarn directions and the strains along these directions.

The density (*Density*) is required when using a fabric material.

Input File Usage: Use the following options to define a fabric material through user subroutine *VFABRIC*:

**MATERIAL*, NAME=*name*
**FABRIC*, USER
**DENSITY*

Properties for a User-Defined Fabric Material

Any material constants that are needed in user subroutine *VFABRIC* must be specified as part of a user-defined fabric material definition. Abaqus can be used to compute the isotropic thermal expansion response under thermal loading, even as the remaining mechanical response is defined by the user subroutine. Alternatively, you can include the thermal expansion within the definition of the mechanical response in user subroutine *VFABRIC*.

Input File Usage: Use the following option to define properties for a user-defined fabric material behavior:

**FABRIC*, USER, PROPERTIES=*number_of_constants*

Material State

Many mechanical constitutive models require the storage of solution-dependent state variables (plastic strains, "back stress," saturation values, etc. in rate constitutive forms or historical data for theories written in integral form). You should allocate storage for these variables in the associated material definition (see *Allocating Space for Solution-Dependent State Variables*). There is no restriction on the number of state variables associated with a user-defined fabric material.

State variables associated with *VFABRIC* can be output to the output database (`.odb`) file and results (`.fil`) file using output identifiers SDV and SDV*n* (see *Abaqus/Explicit Output Variable Identifiers*).

User subroutine *VFABRIC* is called for blocks of material points at each increment. When the subroutine is called, it is provided with the state at the start of the increment (fabric stress in the local system, solution-dependent state variables). It is also provided with the nominal fabric strains at the end of the increment and the incremental nominal fabric strains over the increment, both in the local system. The *VFABRIC* user material interface passes a block of material points to the subroutine on each call, which allows vectorization of the material subroutine.

The temperature is provided to user subroutine *VFABRIC* at the start and the end of the increment. The temperature is passed in as information only and cannot be modified, even in a fully coupled thermal-stress analysis. However, if the inelastic heat fraction is defined in conjunction with the specific heat and conductivity in a fully coupled thermal-stress analysis, the heat flux due to inelastic energy dissipation is calculated automatically. If user subroutine *VFABRIC* is used to define an adiabatic material behavior (conversion of plastic work to heat) in an explicit dynamic procedure, the temperatures must be stored and integrated as user-defined state variables. Most often the temperatures are provided by specifying initial conditions (*Initial Conditions*) and are constant throughout the analysis.

Deleting Elements from an Abaqus/Explicit Mesh Using State Variables

Element deletion in a mesh can be controlled during the course of an Abaqus/Explicit analysis through user subroutine *VFABRIC*. Deleted elements have no ability to carry stresses and, therefore, have no contribution to the stiffness of the model. You specify the state variable number controlling the element deletion flag. For example, specifying a state variable number of 4 indicates that the fourth state variable is the deletion flag in *VFABRIC*. The deletion state variable should be set to a value of one or zero in *VFABRIC*. A value of one indicates that the material point is active, while a value of zero indicates that Abaqus/Explicit should delete the material point from the model by setting the stresses to zero. The structure of the block of material points passed to user subroutine *VFABRIC* remains unchanged during the analysis; deleted material points are not removed from the block. Abaqus/Explicit passes zero stresses and strain increments for all deleted material points. Once a material point is flagged as deleted, it cannot be reactivated. An element is deleted from a mesh based on the material point status (active or deleted). Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Input File Usage: **DEPVAR*, DELETE=*variable number*

Thermal Expansion

You can define isotropic thermal expansion to specify the same coefficient of thermal expansion for the membrane and thickness-direction behaviors.

The membrane thermal strains, $\boldsymbol{\varepsilon^{th}}$, are obtained as explained in *Thermal Expansion*.

The elastic stretch in a given direction, $\boldsymbol{\lambda^{el}}$, relates the total stretch, $\boldsymbol{\lambda}$, and the thermal stretch, $\boldsymbol{\lambda^{th}}$:

$$\boldsymbol{\lambda^{el}} = \frac{\boldsymbol{\lambda}}{\boldsymbol{\lambda^{th}}}.$$

$\boldsymbol{\lambda^{th}}$ is given by

$$\boldsymbol{\lambda^{th}} = 1 + \boldsymbol{\varepsilon^{th}},$$

where $\boldsymbol{\varepsilon^{th}}$ is the linear thermal expansion strain in that direction.

Fabric Thickness

The thickness of a fabric is difficult to measure experimentally. Fortunately, an accurate value for thickness is not always required due to the fact that a nominal stress measure, defined as force per unit area in the reference configuration, is used to characterize the in-plane response. An initial thickness can be specified on the section definition. Accurate tracking of the thickness with deformation is necessary only if the material is used with shell elements and the bending response needs to be captured accurately. You can compute the thickness direction strain increment when the fabric is defined through user subroutine *VFABRIC*. For test data–based fabric materials the thickness is assumed to remain constant with deformation. For a test data–based fabric definition, you must use the thickness value specified on the section definition for converting the experimental load data (which are typically available as force applied per unit width of the fabric) to stress quantities.

Defining a Reference Mesh (Initial Metric)

Abaqus/Explicit allows the specification of a reference mesh (initial metric) for fabrics modeled with membrane elements. For example, this is useful in airbag simulations to model wrinkles and changes in yarn orientations that arise from the airbag folding process. A flat mesh may be suitable for the unstressed reference configuration, but the initial state may require a corresponding folded mesh defining the folded state. The angular orientation of the yarn in the reference configuration is updated to obtain the new orientation in the initial configuration.

Input File Usage: Use the following option to define the reference configuration giving the element number and its nodal coordinates in the reference configuration:

**INITIAL CONDITIONS*, TYPE=REF COORDINATE

Use the following option to define the reference configuration giving the node number and its coordinates in the reference configuration:

**INITIAL CONDITIONS*, TYPE=NODE REF COORDINATE

Yarn Behavior under Initial Compressive Strains

Defining a reference configuration that is different from the initial configuration generally results in nonzero stresses and strains in the initial configuration based on the material definition. By default, compressive initial strains in the yarn directions generate zero stresses. The stress remains zero as the strain is continuously recovered from the initial compressive values toward the strain-free state. Once the initial slack is recovered, any subsequent compressive/tensile strains generate stresses as per the material definition.

Input File Usage: Use the following option to specify that initial compressive strains are recovered stress free (default):

**FABRIC*, STRESS FREE INITIAL SLACK=YES

Use the following option to specify that initial compressive strains generate nonzero initial stresses:

**FABRIC*, STRESS FREE INITIAL SLACK=NO

Defining Yarn Directions in the Reference Configuration

In general, the yarn directions may not be orthogonal to each other in the reference configuration. You can specify these local directions with respect to the in-plane axes of an orthogonal orientation system at a material point. Both the local directions and the orthogonal system are defined together as a single orientation definition. See *Orientations* for more information.

If the local directions are not specified, these directions are assumed to match the in-plane axes of the orthogonal system defined. The local direction may not remain orthogonal with deformation. Abaqus updates the local directions with deformation and computes the nominal strains along these directions and the angle between them (fabric shear strain). The constitutive behavior for the fabric defines the nominal stresses in the local system in terms of the fabric strain.

Local yarn directions can be output to the output database as described in *Output* below.

Picture-Frame Shear Fabric Test

The shear response of the fabric is typically studied using a picture-frame shear test. The reference and the deformed configuration for a picture-frame shear test under force $\boldsymbol{F}$ is illustrated in *Figure 11*, where $\boldsymbol{L_0}$ is the size of the picture-frame, and $\boldsymbol{\psi_{12}^0}$ is the initial angle between the yarn directions. The four sides of the specimen are constrained not to change in their length even as the frame elongates and the angle between the yarn directions $\boldsymbol{\psi_{12}}$ decreases with deformation.

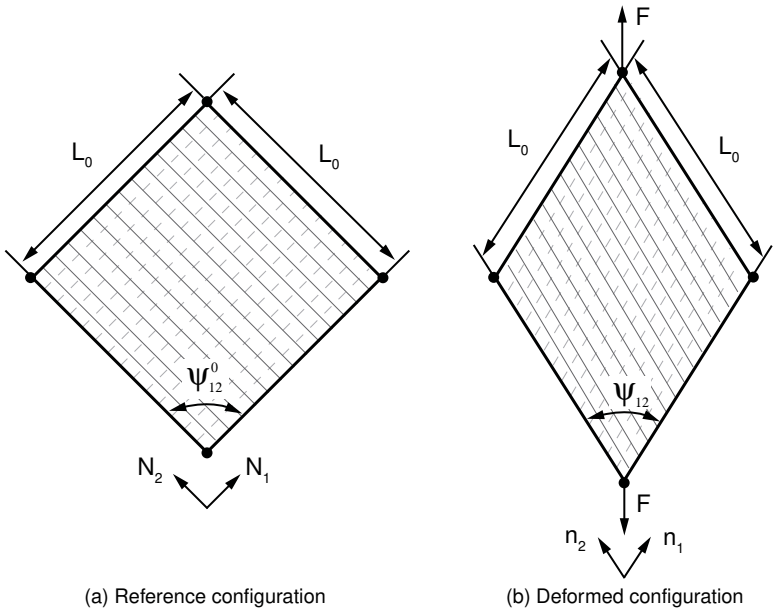


Figure 11: Picture-frame shear test on a fabric.

The relationship between the nominal shear stress, T_{12}, and the applied force, F, is

$$T_{12} = \left(\frac{FL_0}{v_0}\right) \sin\left(\frac{\psi_{12}}{2}\right),$$

where v_0 is the initial volume of the specimen. The fabric engineering shear strain, γ_{12}, is related to the change in the angle between the yarn directions as

$$\gamma_{12} = \psi_{12}^0 - \psi_{12}.$$

Material Options

The fabric material model can be used by itself, or it can be combined with isotropic thermal expansion to introduce thermal volume changes (*Thermal Expansion*). See *Combining Material Behaviors* for more details. Thermal expansion can alternatively be an integral part of the constitutive model implemented in *VFABRIC* for user-defined fabric materials.

For a test-data based fabric material, both the mass proportional and the stiffness proportional damping can be specified (see *Material Damping*). If stiffness proportional damping is specified, Abaqus calculates the damping stress based on the current elastic stiffness of the material and the resulting damping stress is included in the reported stress output at the integration points.

For a fabric material defined by user subroutine *VFABRIC*, mass proportional damping can be specified, but stiffness proportional damping must be defined within the user subroutine.

Elements

The fabric material model can be used with plane stress elements (plane stress solid elements, finite-strain shells, and membranes). It is recommended that the fabric material model be used with fully integrated or triangular membrane elements. When the fabric material model is used with shell elements, Abaqus does not compute a transverse shear stiffness by default. You must either specify the material transverse shear modulus (see *Defining the Elastic Transverse Shear Modulus*) based on which Abaqus can calculate the transverse shear stiffness or

define the transverse shear stiffness for the section directly (see *Transverse Shear Stiffness* and *Defining the Transverse Shear Stiffness*).

Procedures

Fabric materials must always be used with geometrically nonlinear analyses (*General and Perturbation Procedures*).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the fabric material models:

EFABRIC Nominal fabric strain with components similar to that of LE, but with the direct components measuring the nominal strain along the yarn directions and the engineering shear component measuring the change in angle between the two yarn directions.

SFABRIC Nominal fabric stress with components similar to that of the regular Cauchy stress, S, but with the direct components measuring the nominal stress along the yarn directions and the shear component measuring response to the change in angle between the two yarn directions.

By default, Abaqus outputs local material directions whenever element field output is requested to the output database for fabric models. The local directions are output as field variables (LOCALDIR1 and LOCALDIR2) representing the yarn direction cosines with respect to the orthogonal system at the material point resulting from the current orientation definition; these variables can be visualized as vector plots in the Visualization module of Abaqus/CAE (Abaqus/Viewer).

Output of local material directions is suppressed if no element field output is requested or if you specify not to have element material directions written to the output database (see *Specifying the Directions for Element Output*).

Jointed Material Model

Products: Abaqus/Standard

References:

- *Orientations*
- *About the Material Library*
- *Inelastic Behavior*
- **JOINTED MATERIAL*

Overview

The jointed material model:

- is intended to provide a simple continuum model for a material containing a high density of parallel joint surfaces where each system of parallel joints is associated with a particular orientation, such as sedimentary rock;
- assumes that the spacing of the joints of a particular orientation is sufficiently close compared to characteristic dimensions in the domain of the model such that the joints can be smeared into a continuum of slip systems;
- provides for opening or frictional sliding of the joints in each of these systems (a “system” in this context is a joint orientation in a particular direction at a material calculation point); and
- assumes that the elastic behavior of the material is isotropic and linear when all joints at a point are closed (isotropic linear elastic behavior must be included in the material definition; see *Defining Isotropic Elasticity*).

Joint Opening/Closing

The jointed material model is intended primarily for applications where the stresses are mainly compressive. The model provides a joint opening capability when the stress normal to the joint tries to become tensile. In this case the stiffness of the material normal to the joint plane becomes zero instantaneously. Abaqus/Standard uses a stress-based joint opening criterion, whereas joint closing is monitored based on strain. Joint system a opens when the estimated pressure stress across the joint (normal to the joint surface) is no longer positive:

$$p_a \leq 0.$$

In this case the material is assumed to have no elastic stiffness with respect to direct strain across the joint system. Open joints thus create anisotropic elastic response at a point. The joint system remains open so long as

$$\varepsilon^{el}_{an(ps)} \leq \varepsilon^{el}_{an},$$

where ε^{el}_{an} is the component of direct elastic strain across the joint and $\varepsilon^{el}_{an(ps)}$ is the component of direct elastic strain across the joint calculated in plane stress as

$$\varepsilon^{el}_{an(ps)} = -\frac{\nu}{E}\left(\sigma_{a1} + \sigma_{a2}\right),$$

where E is the Young's modulus of the material, ν is the Poisson's ratio, and σ_{a1}, σ_{a2} are the direct stresses in the plane of the joint.

The shear response of open joints is governed by the shear retention parameter, f_{sr}, which represents the fraction of the elastic shear modulus retained when the joints are open (f_{sr}=0 means no shear stiffness associated with open joints, while f_{sr}=1 corresponds to elastic shear stiffness in open joints; any value between these two extremes can be used). When a joint opens, the shear behavior might be brittle, depending on the shear retention factor used for open joints. In addition, the stiffness of the material normal to the joint plane suddenly goes to zero. For these reasons, in situations where the confining stresses are low or significant regions experience tensile behavior, the joint systems might experience a sequence of alternate opening and closing states from iteration to iteration. Typically such behavior manifests itself as oscillating global residual forces. The convergence rate associated with such discontinuous behavior might be very slow and, thus, prohibit obtaining a solution. This type of failure is more probable in cases where more than one joint system is modeled.

Improving Convergence When Joints Open and Close Repeatedly

When the repeated opening and closing of joints makes convergence difficult, you can improve convergence by preventing a joint from opening. In this case an elastic stiffness is always associated with the joint. It is most useful when the opening and closing of joints is limited to small regions of the model. You can prevent a joint from opening only when the joint direction is specified, as described below.

Input File Usage: **JOINTED MATERIAL*, NO SEPARATION, JOINT DIRECTION

Specifying Nonzero Shear Retention in Open Joints

You must specify nonzero shear retention in open joints directly. The parameter f_{sr} can be defined as a tabular function of temperature and predefined field variables.

Input File Usage: **JOINTED MATERIAL*, SHEAR RETENTION

Compressive Joint Sliding

The failure surface for sliding on joint system a is defined by

$$f_a = \tau_a - p_a \tan \beta_a - d_a = 0,$$

where τ_a is the magnitude of the shear stress resolved onto the joint surface, p_a is the normal pressure stress acting across the joint, β_a is the friction angle for system a, and d_a is the cohesion for system a. So long as $f_a < 0$, joint system a does not slip. When $f_a = 0$, joint system a slips. The inelastic (“plastic”) strain on the system is given by

$$\begin{aligned} d\gamma_{a\alpha}^{pl} &= d\bar{\varepsilon}_a^{pl} \frac{\tau_{a\alpha}}{\tau_a} \cos \psi_a \\ d\varepsilon_{an}^{pl} &= d\bar{\varepsilon}_a^{pl} \sin \psi_a, \end{aligned}$$

where

$d\gamma_{a\alpha}^{pl}$

is the rate of inelastic shear strain in direction α on the joint surface ($\alpha = 1, 2$ are orthogonal directions on the joint surface),

$d\bar{\varepsilon}_a^{pl}$

is the magnitude of the inelastic strain rate,

$\tau_{a\alpha}$

is a component of the shear stress on the joint surface,

ψ_a

is the dilation angle for this joint system (choosing $\psi_a = 0$ provides pure shear flow on the joint, while $\psi_a > 0$ causes dilation of the joint as it slips), and

$d\varepsilon_{an}^{pl}$

is the inelastic strain normal to the joint surface.

The sliding of the different joint systems at a point is independent, in the sense that sliding on one system does not change the failure criterion or the dilation angle for any other joint system at the same point.

Up to three joint directions can be included in the material description. The orientations of the joint directions are given by referring to the names of user-defined local orientations (*Orientations*) that define the joint orientations in the original configuration. Output of stress and strain components is in the global directions unless a local orientation is also used in the material's section definition.

The parameters β_a, ψ_a, and d_a can be specified as tabular functions of temperature and/or predefined field variables for each joint direction.

Input File Usage: Use both of the following options:

**ORIENTATION*, NAME=*name*
**JOINTED MATERIAL*, JOINT DIRECTION=*name*

Repeat the **JOINTED MATERIAL* option for each direction to be specified, up to three times.

Joint Directions and Finite Rotations

In geometrically nonlinear analysis steps the joint directions always remain fixed in space.

Bulk Failure

In addition to the joint systems, the jointed material model includes a bulk material failure mechanism, which is based on the Drucker-Prager failure criterion:

$$q - p\tan\beta_b - d_b = 0,$$

where

$q \overset{\text{def}}{=} \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}}$

is the Mises equivalent deviatoric stress,

$\mathbf{S} \overset{\text{def}}{=} \boldsymbol{\sigma} + p\mathbf{I}$

is the deviatoric stress,

$p \overset{\text{def}}{=} -\frac{1}{3}\mathbf{I} : \boldsymbol{\sigma}$

is the equivalent pressure stress,

β_b

is the friction angle for the bulk material, and

d_b

is the cohesion for the bulk material.

If this failure criterion is reached, the bulk inelastic flow is defined by

$$d\boldsymbol{\varepsilon}_b^{pl} = d\bar{\varepsilon}_b^{pl} \frac{1}{1 - \frac{1}{3}\tan\psi_b} \frac{\partial g_b}{\partial \boldsymbol{\sigma}},$$

where

$$g_b = q - p\tan\psi_b$$

is the flow potential. Here $d\bar{\varepsilon}_b^{pl}$ is the magnitude of the inelastic flow rate (chosen so that $d\bar{\varepsilon}_b^{pl} = \left|\left(d\boldsymbol{\varepsilon}_b^{pl}\right)_{11}\right|$ in uniaxial compression in the 1-direction), and ψ_b is the dilation angle for the bulk material. This bulk failure model is a simplified version of the extended Drucker-Prager model (*Extended Drucker-Prager Models*). This bulk failure system is independent of the joint systems in that bulk inelastic flow does not change the behavior of any joint system.

If bulk material failure is to be modeled, a jointed material behavior must be specified to define the parameters associated with bulk material failure behavior. Thus, up to five jointed material behaviors can appear in the same material definition: three joint directions, shear retention in open joints, and bulk material failure.

The parameters β_b, ψ_b, and d_b can be specified as a tabular function of temperature and/or predefined field variables.

Input File Usage: **JOINTED MATERIAL* (the JOINT DIRECTION parameter must be omitted)

Nonassociated Flow

If $\psi \neq \beta$ in any joint system, whether it be associated with the joint surfaces or the bulk material, the flow in that system is "nonassociated." The implication is that the material stiffness matrix is not symmetric. Therefore, the unsymmetric matrix solution scheme should be used for the analysis step (*Defining an Analysis*), especially

when large regions of the model are expected to flow plastically and when the difference between ψ and β is large. If the difference between ψ and β is not large, a symmetric approximation to the matrix can provide an acceptable rate of convergence of the equilibrium equations and, hence, a lower overall solution cost. Therefore, the unsymmetric matrix solution scheme is not invoked automatically when jointed material behavior is defined.

Elements

The jointed material model can be used with plane strain, generalized plane strain, axisymmetric, and three-dimensional solid (continuum) elements in Abaqus/Standard. This model cannot be used with elements for which the assumed stress state is plane stress (plane stress, shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning in the jointed material model:

PE
Sum of the plastic strains for all four possible systems.

PEEQ
Sum of the equivalent plastic strains for all four possible systems.

PEQC
Equivalent plastic strains for all four possible systems (three joints - PEQC1, PEQC2, and PEQC3, and bulk material - PEQC4). This identifier also provides yes/no flags (1/0 on the output database) AC YIELD1, AC YIELD2, AC YIELD3, and AC YIELD4 indicating if each individual system is currently yielding, that is, whether or not the plastic strain changed during the increment.

Concrete

In this section:

- *Concrete Smeared Cracking*
- *Cracking Model for Concrete*
- *Concrete Damaged Plasticity*

Concrete Smeared Cracking

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **CONCRETE*
- **TENSION STIFFENING*
- **SHEAR RETENTION*
- **FAILURE RATIOS*
- *Defining concrete smeared cracking*

Overview

The smeared crack concrete model in Abaqus/Standard:

- provides a general capability for modeling concrete in all types of structures, including beams, trusses, shells, and solids;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- can be used with rebar to model concrete reinforcement;
- is designed for applications in which the concrete is subjected to essentially monotonic straining at low confining pressures;
- consists of an isotropically hardening yield surface that is active when the stress is dominantly compressive and an independent "crack detection surface" that determines if a point fails by cracking;
- uses oriented damaged elasticity concepts (smeared cracking) to describe the reversible part of the material's response after cracking failure;
- requires that the linear elastic material model (see *Linear Elastic Behavior*) be used to define elastic properties; and
- cannot be used with local orientations (see *Orientations*).

See *Inelastic Behavior* for a discussion of the concrete models available in Abaqus.

Reinforcement

Reinforcement in concrete structures is typically provided by means of rebars, which are one-dimensional strain theory elements (rods) that can be defined singly or embedded in oriented surfaces. Rebars are typically used with metal plasticity models to describe the behavior of the rebar material and are superposed on a mesh of standard element types used to model the concrete.

With this modeling approach, the concrete behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some "tension stiffening" into the concrete modeling to simulate load transfer across cracks through the rebar. Details regarding tension stiffening are provided below.

Defining the rebar can be tedious in complex problems, but it is important that this be done accurately since it may cause an analysis to fail due to lack of reinforcement in key regions of a model. See *Defining Reinforcement* for more information regarding rebars.

Cracking

The model is intended as a model of concrete behavior for relatively monotonic loadings under fairly low confining pressures (less than four to five times the magnitude of the largest stress that can be carried by the concrete in uniaxial compression).

Crack Detection

Cracking is assumed to be the most important aspect of the behavior, and representation of cracking and of postcracking behavior dominates the modeling. Cracking is assumed to occur when the stress reaches a failure surface that is called the "crack detection surface." This failure surface is a linear relationship between the equivalent pressure stress, p, and the Mises equivalent deviatoric stress, q, and is illustrated in *Figure 5*. When a crack has been detected, its orientation is stored for subsequent calculations. Subsequent cracking at the same point is restricted to being orthogonal to this direction since stress components associated with an open crack are not included in the definition of the failure surface used for detecting the additional cracks.

Cracks are irrecoverable: they remain for the rest of the calculation (but may open and close). No more than three cracks can occur at any point (two in a plane stress case, one in a uniaxial stress case). Following crack detection, the crack affects the calculations because a damaged elasticity model is used. Oriented, damaged elasticity is discussed in more detail in *An inelastic constitutive model for concrete*.

Smeared Cracking

The concrete model is a smeared crack model in the sense that it does not track individual "macro" cracks. Constitutive calculations are performed independently at each integration point of the finite element model. The presence of cracks enters into these calculations by the way in which the cracks affect the stress and material stiffness associated with the integration point.

Tension Stiffening

The postfailure behavior for direct straining across cracks is modeled with tension stiffening, which allows you to define the strain-softening behavior for cracked concrete. This behavior also allows for the effects of the reinforcement interaction with concrete to be simulated in a simple manner. Tension stiffening is required in the concrete smeared cracking model. You can specify tension stiffening by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure Stress-Strain Relation

Specification of strain softening behavior in reinforced concrete generally means specifying the postfailure stress as a function of strain across the crack. In cases with little or no reinforcement this specification often introduces mesh sensitivity in the analysis results in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands. This problem typically occurs if only a few discrete cracks form in the structure, and mesh refinement does not result in formation of additional cracks. If cracks are evenly distributed (either due to the effect of rebar or due to the presence of stabilizing elastic material, as in the case of plate bending), mesh sensitivity is less of a concern.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. The interaction between the rebars and the concrete tends to reduce the mesh sensitivity, provided that a reasonable amount of tension stiffening is introduced in the concrete model to simulate this interaction (*Figure 1*).

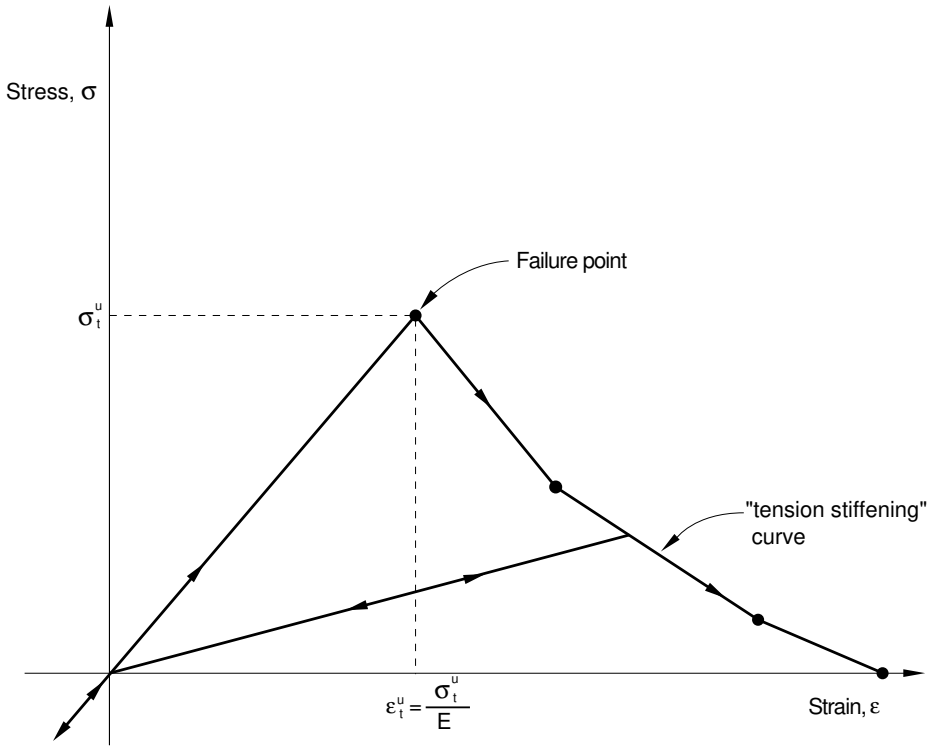


Figure 1: "Tension stiffening" model.

The tension stiffening effect must be estimated; it depends on such factors as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. The strain at failure in standard concretes is typically 10^{-4}, which suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

The choice of tension stiffening parameters is important in Abaqus/Standard since, generally, more tension stiffening makes it easier to obtain numerical solutions. Too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: Use both of the following options:

**CONCRETE*
**TENSION STIFFENING*, TYPE=STRAIN (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Smeared Cracking**: **Suboptions->Tension Stiffening**: **Type: Strain**

Fracture Energy Cracking Criterion

As discussed earlier, when there is no reinforcement in significant regions of a concrete model, the strain softening approach for defining tension stiffening may introduce unreasonable mesh sensitivity into the results. Crisfield (1986) discusses this issue and concludes that Hillerborg's (1976) proposal is adequate to allay the concern for

many practical purposes. Hillerborg defines the energy required to open a unit area of crack as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-*displacement* response rather than a stress-*strain* response. Under tension a concrete specimen will crack across some section. After it has been pulled apart sufficiently for most of the stress to be removed (so that the elastic strain is small), its length will be determined primarily by the opening at the crack. The opening does not depend on the specimen's length (*Figure 2*).

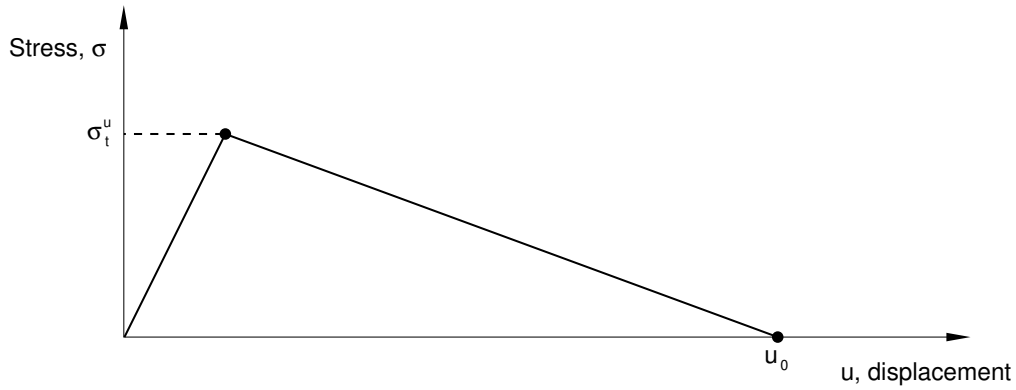


Figure 2: Fracture energy cracking model.

Implementation

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length associated with an integration point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic crack length is used because the direction in which cracks will occur is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect, and elements that are as close to square as possible are recommended.

This approach to modeling the concrete's brittle response requires the specification of the displacement u_0 at which a linear approximation to the postfailure strain softening gives zero stress (see *Figure 2*).

The failure stress, σ_t^u, occurs at a failure *strain* (defined by the failure stress divided by the Young's modulus); however, the stress goes to zero at an ultimate *displacement*, u_0, that is independent of the specimen length. The implication is that a displacement-loaded specimen can remain in static equilibrium after failure only if the specimen is short enough so that the strain at failure, ε_t^u, is less than the strain at this value of the displacement:

$$\varepsilon_t^u < u_0/L,$$

where L is the length of the specimen.

Input File Usage: Use both of the following options:

**CONCRETE*
**TENSION STIFFENING*, TYPE=DISPLACEMENT

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Smeared Cracking**: **Suboptions->Tension Stiffening**: **Type: Displacement**

Obtaining the Ultimate Displacement

The ultimate displacement, u_0, can be estimated from the fracture energy per unit area, G_f, as $u_0 = 2G_f/\sigma_t^u$, where σ_t^u is the maximum tensile stress that the concrete can carry. Typical values for u_0 are 0.05 mm (2×10^{-3} in) for a normal concrete to 0.08 mm (3×10^{-3} in) for a high strength concrete. A typical value for ε_t^u is about 10^{-4}, so that the requirement is that $L < 500$ mm (20 in).

Critical Length

If the specimen is longer than the critical length, L, more strain energy is stored in the specimen than can be dissipated by the cracking process when it cracks under fixed displacement. Some of the strain energy must, therefore, be converted into kinetic energy, and the failure event must be dynamic even under prescribed displacement loading. This implies that, when this approach is used in finite elements, characteristic element dimensions must be less than this critical length, or additional (dynamic) considerations must be included. The analysis input file processor checks the characteristic length of each element using this concrete model and will not allow any element to have a characteristic length that exceeds u_0/ε_t^u. You must remesh with smaller elements where necessary or use the stress-strain definition of tension stiffening. Since the fracture energy approach is generally used only for plain concrete, this rarely places any limit on the meshing.

Cracked Shear Retention

As the concrete cracks, its shear stiffness is diminished. This effect is defined by specifying the reduction in the shear modulus as a function of the opening strain across the crack. You can also specify a reduced shear modulus for closed cracks. This reduced shear modulus will also have an effect when the normal stress across a crack becomes compressive. The new shear stiffness will have been degraded by the presence of the crack.

The modulus for shearing of cracks is defined as ϱG, where G is the elastic shear modulus of the uncracked concrete and ϱ is a multiplying factor. The shear retention model assumes that the shear stiffness of open cracks reduces linearly to zero as the crack opening increases:

$$\varrho = (1 - \varepsilon/\varepsilon^{\max}) \text{ for } \varepsilon < \varepsilon^{\max}, \qquad \varrho = 0 \text{ for } \varepsilon \geq \varepsilon^{\max},$$

where ε is the direct strain across the crack and $\varepsilon^{\max}$ is a user-specified value. The model also assumes that cracks that subsequently close have a reduced shear modulus:

$$\varrho = \varrho^{\text{close}} \text{ for } \varepsilon < 0,$$

where you specify ϱ^{close}.

ϱ^{close} and $\varepsilon^{\max}$ can be defined with an optional dependency on temperature and/or predefined field variables. If shear retention is not included in the material definition for the concrete smeared cracking model, Abaqus/Standard will automatically invoke the default behavior for shear retention such that the shear response is unaffected by cracking (full shear retention). This assumption is often reasonable: in many cases, the overall response is not strongly dependent on the amount of shear retention.

Input File Usage: Use both of the following options:

**CONCRETE*
**SHEAR RETENTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Smeared Cracking**: **Suboptions->Shear Retention**

Compressive Behavior

When the principal stress components are dominantly compressive, the response of the concrete is modeled by an elastic-plastic theory using a simple form of yield surface written in terms of the equivalent pressure stress, p, and the Mises equivalent deviatoric stress, q; this surface is illustrated in *Figure 5*. Associated flow and isotropic hardening are used. This model significantly simplifies the actual behavior. The associated flow assumption generally over-predicts the inelastic volume strain. The yield surface cannot be matched accurately to data in triaxial tension and triaxial compression tests because of the omission of third stress invariant dependence. When the concrete is strained beyond the ultimate stress point, the assumption that the elastic response is not affected by the inelastic deformation is not realistic. In addition, when concrete is subjected to very high pressure stress, it exhibits inelastic response: no attempt has been made to build this behavior into the model.

The simplifications associated with compressive behavior are introduced for the sake of computational efficiency. In particular, while the assumption of associated flow is not justified by experimental data, it can provide results that are acceptably close to measurements, provided that the range of pressure stress in the problem is not large. From a computational viewpoint, the associated flow assumption leads to enough symmetry in the Jacobian matrix of the integrated constitutive model (the "material stiffness matrix") such that the overall equilibrium equation solution usually does not require unsymmetric equation solution. All of these limitations could be removed at some sacrifice in computational cost.

You can define the stress-strain behavior of plain concrete in uniaxial compression outside the elastic range. Compressive stress data are provided as a tabular function of plastic strain and, if desired, temperature and field variables. Positive (absolute) values should be given for the compressive stress and strain. The stress-strain curve can be defined beyond the ultimate stress, into the strain-softening regime.

Input File Usage: **CONCRETE*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Smeared Cracking**

Uniaxial and Multiaxial Behavior

The cracking and compressive responses of concrete that are incorporated in the concrete model are illustrated by the uniaxial response of a specimen shown in *Figure 3*.

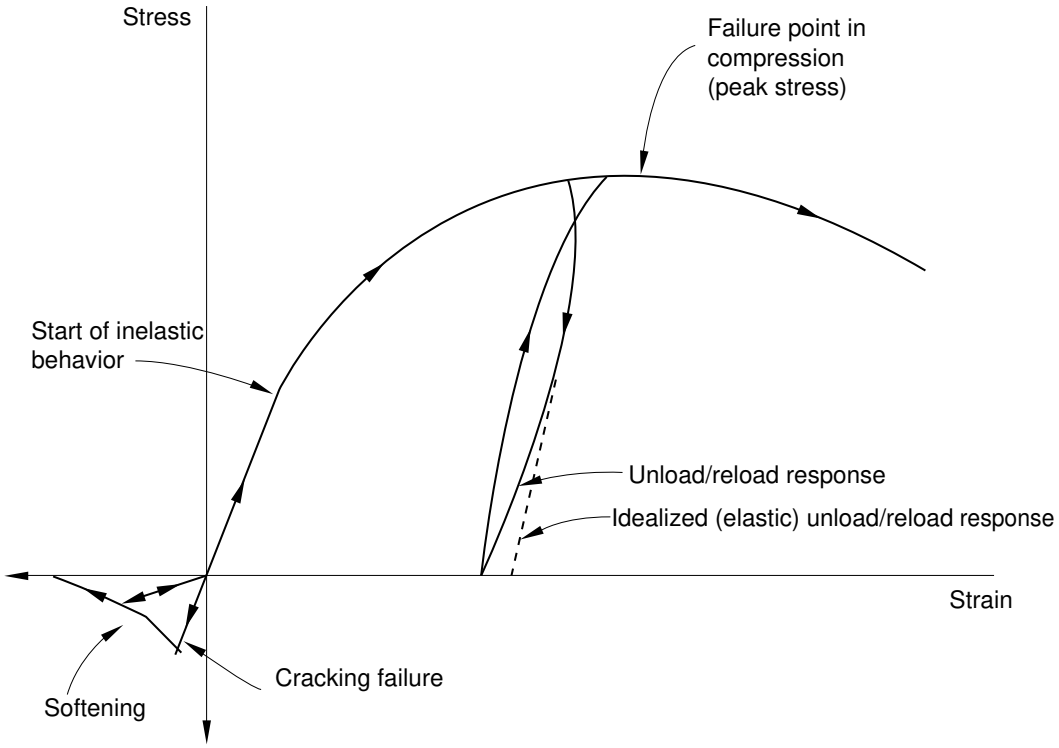


Figure 3: Uniaxial behavior of plain concrete.

When concrete is loaded in compression, it initially exhibits elastic response. As the stress is increased, some nonrecoverable (inelastic) straining occurs and the response of the material softens. An ultimate stress is reached, after which the material loses strength until it can no longer carry any stress. If the load is removed at some point after inelastic straining has occurred, the unloading response is softer than the initial elastic response: the elasticity has been damaged. This effect is ignored in the model, since we assume that the applications involve primarily monotonic straining, with only occasional, minor unloadings. When a uniaxial concrete specimen is loaded in tension, it responds elastically until, at a stress that is typically 7%–10% of the ultimate compressive stress, cracks form. Cracks form so quickly that, even in the stiffest testing machines available, it is very difficult to observe the actual behavior. The model assumes that cracking causes damage, in the sense that open cracks can be represented by a loss of elastic stiffness. It is also assumed that there is no permanent strain associated with cracking. This will allow cracks to close completely if the stress across them becomes compressive.

In multiaxial stress states these observations are generalized through the concept of surfaces of failure and flow in stress space. These surfaces are fitted to experimental data. The surfaces used are shown in *Figure 4* and *Figure 5*.

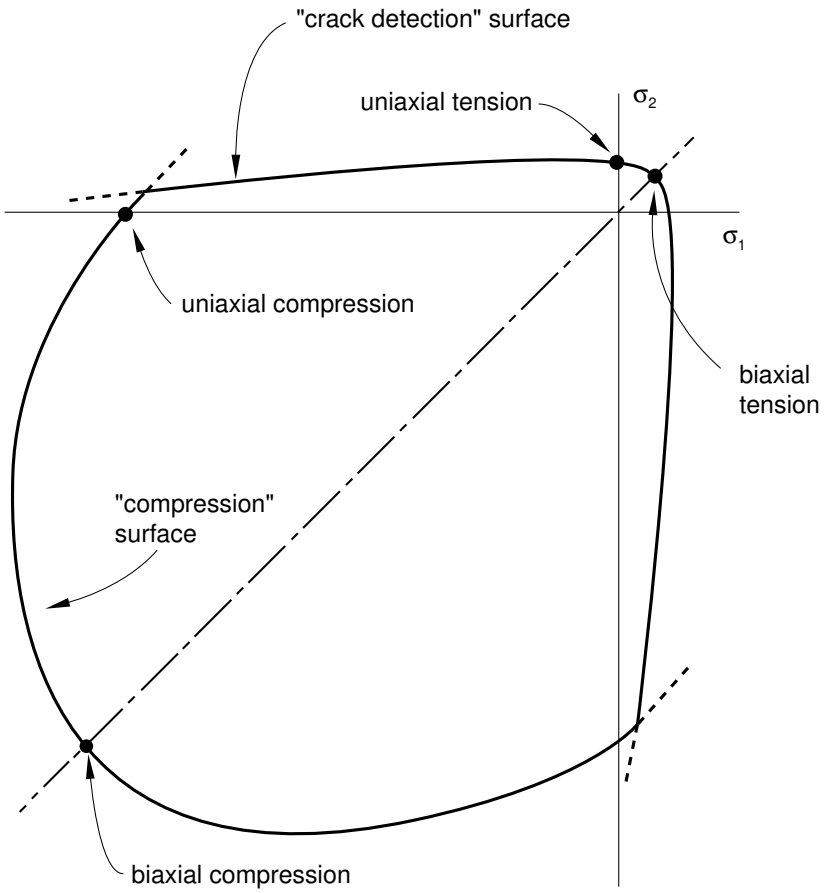


Figure 4: Yield and failure surfaces in plane stress.

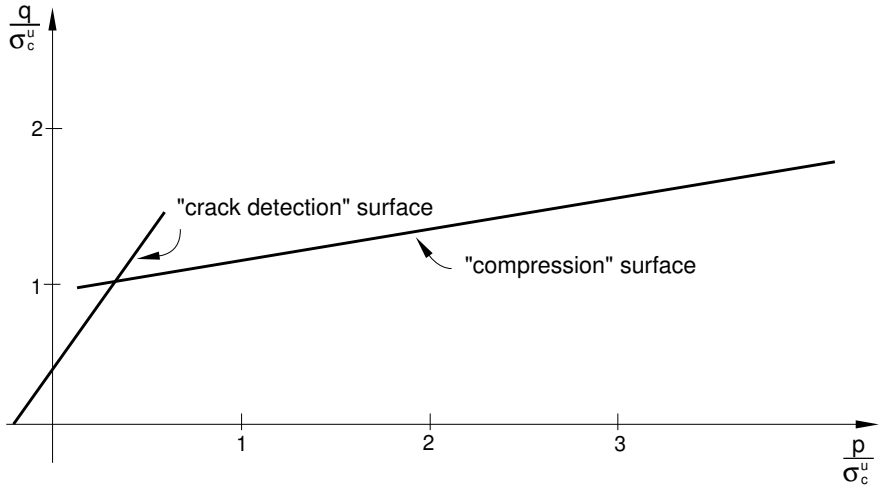


Figure 5: Yield and failure surfaces in the (*p*–*q*) plane.

Failure Surface

You can specify failure ratios to define the shape of the failure surface (possibly as a function of temperature and predefined field variables). Four failure ratios can be specified:

- The ratio of the ultimate biaxial compressive stress to the ultimate uniaxial compressive stress.
- The absolute value of the ratio of the uniaxial tensile stress at failure to the ultimate uniaxial compressive stress.
- The ratio of the magnitude of a principal component of plastic strain at ultimate stress in biaxial compression to the plastic strain at ultimate stress in uniaxial compression.
- The ratio of the tensile principal stress at cracking, in plane stress, when the other principal stress is at the ultimate compressive value, to the tensile cracking stress under uniaxial tension.

Default values of the above ratios are used if you do not specify them.

Input File Usage: **FAILURE RATIOS*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Smeared Cracking**: **Suboptions->Failure Ratios**

Response to Strain Reversals

Because the model is intended for application to problems involving relatively monotonic straining, no attempt is made to include prediction of cyclic response or of the reduction in the elastic stiffness caused by inelastic straining under predominantly compressive stress. Nevertheless, it is likely that, even in those applications for which the model is designed, the strain trajectories will not be entirely radial, so that the model should predict the response to occasional strain reversals and strain trajectory direction changes in a reasonable way. Isotropic hardening of the “compressive” yield surface forms the basis of this aspect of the model's inelastic response prediction when the principal stresses are dominantly compressive.

Calibration

A minimum of two experiments, uniaxial compression and uniaxial tension, is required to calibrate the simplest version of the concrete model (using all possible defaults and assuming temperature and field variable independence). Other experiments may be required to gain accuracy in postfailure behavior.

Uniaxial Compression and Tension Tests

The uniaxial compression test involves compressing the sample between two rigid platens. The load and displacement in the direction of loading are recorded. From this, you can extract the stress-strain curve required for the concrete model directly. The uniaxial tension test is much more difficult to perform in the sense that it is necessary to have a stiff testing machine to be able to record the postfailure response. Quite often this test is not available, and you make an assumption about the tensile failure strength of the concrete (usually about 7%–10% of the compressive strength). The choice of tensile cracking stress is important; numerical problems may arise if very low cracking stresses are used (less than 1/100 or 1/1000 of the compressive strength).

Postcracking Tensile Behavior

The calibration of the postfailure response depends on the reinforcement present in the concrete. For plain concrete simulations the stress-displacement tension stiffening model should be used. Typical values for u_0 are 0.05 mm (2×10^{-3} in) for a normal concrete to 0.08 mm (3×10^{-3} in) for a high-strength concrete. For reinforced concrete simulations the stress-strain tension stiffening model should be used. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

Postcracking Shear Behavior

Combined tension and shear experiments are used to calibrate the postcracking shear behavior in Abaqus/Standard. These experiments are quite difficult to perform. If the test data are not available, a reasonable starting point is to assume that the shear retention factor, ρ, goes linearly to zero at the same crack opening strain used for the tension stiffening model.

Biaxial Yield and Flow Parameters

Biaxial experiments are required to calibrate the biaxial yield and flow parameters used to specify the failure ratios. If these are not available, the defaults can be used.

Temperature Dependence

The calibration of temperature dependence requires the repetition of all the above experiments over the range of interest.

Comparison with Experimental Results

With proper calibration, the concrete model should produce reasonable results for mostly monotonic loadings. Comparison of the predictions of the model with the experimental results of Kupfer and Gerstle (1973) are shown in *Figure 6* and *Figure 7*.

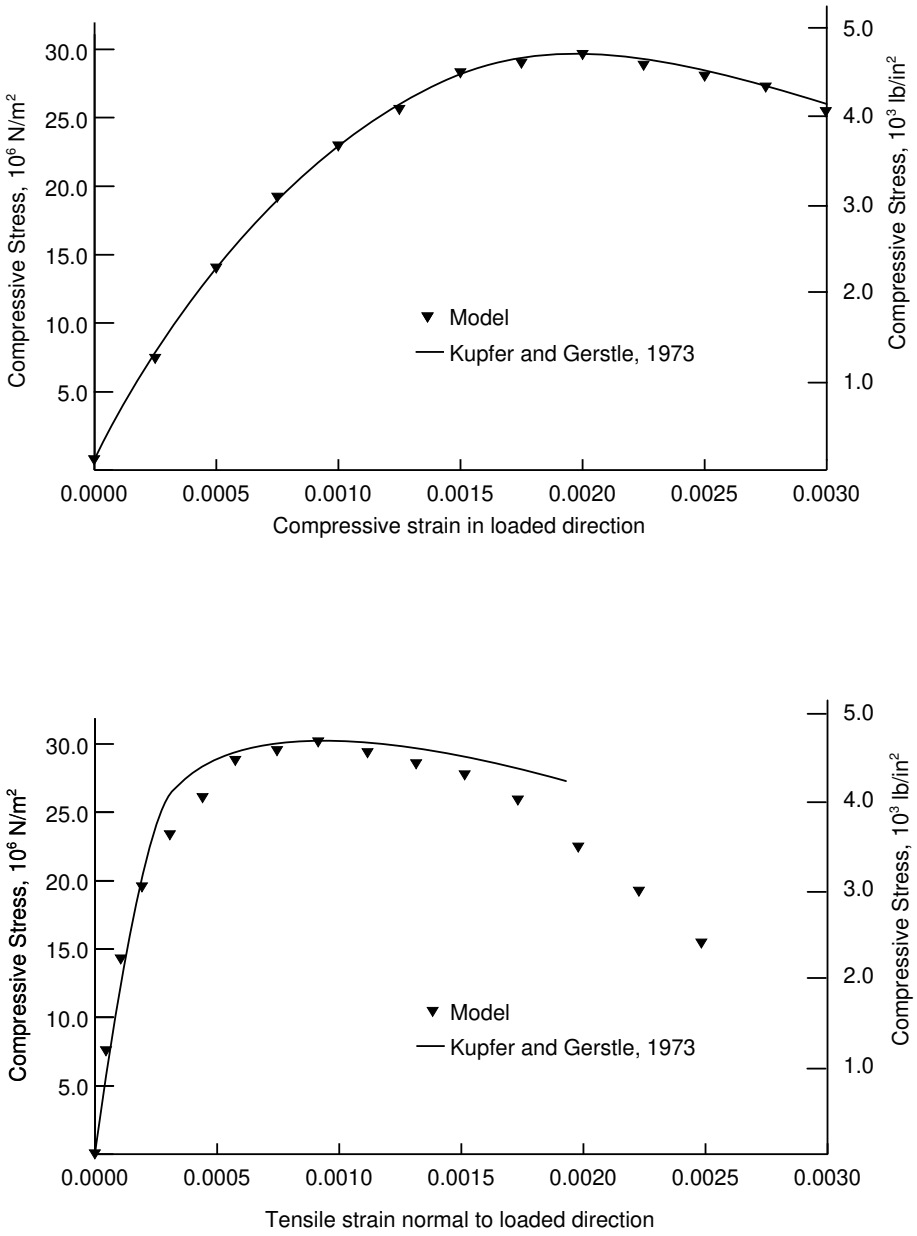


Figure 6: Comparison of model prediction and Kupfer and Gerstle's data for a uniaxial compression test.

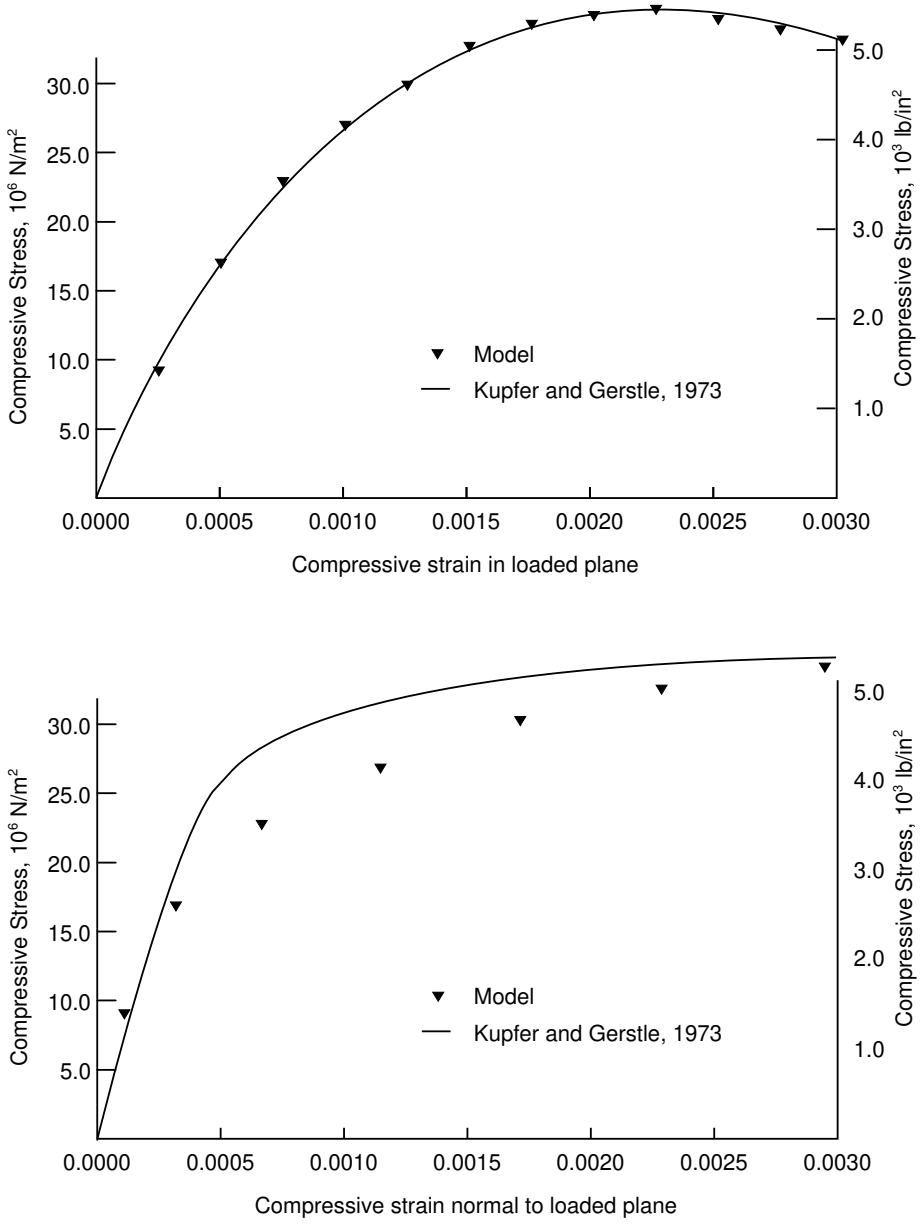


Figure 7: Comparison of model prediction and Kupfer and Gerstle's data for a biaxial compression test.

Elements

Abaqus/Standard offers a variety of elements for use with the smeared crack concrete model: beam, shell, plane stress, plane strain, generalized plane strain, axisymmetric, and three-dimensional elements.

For general shell analysis more than the default number of five integration points through the thickness of the shell should be used; nine thickness integration points are commonly used to model progressive failure of the concrete through the thickness with acceptable accuracy.

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables relate specifically to material points in the smeared crack concrete model:

CRACK	Unit normal to cracks in concrete.
CONF	Number of cracks at a concrete material point.

References

- Crisfield, M. A., "Snap-Through and Snap-Back Response in Concrete Structures and the Dangers of Under-Integration," *International Journal for Numerical Methods in Engineering*, vol. 22, pp. 751–767, 1986.
- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," *Cement and Concrete Research*, vol. 6, pp. 773–782, 1976.
- Kupfer, H. B., , and K. H. Gerstle, "Behavior of Concrete under Biaxial Stresses," *Journal of Engineering Mechanics Division, ASCE*, vol. 99 853, 1973.

Cracking Model for Concrete

Products: Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **BRITTLE CRACKING*
- **BRITTLE FAILURE*
- **BRITTLE SHEAR*
- *Defining brittle cracking*

Overview

The brittle cracking model in Abaqus/Explicit:

- provides a capability for modeling concrete in all types of structures: beams, trusses, shells and solids;
- can also be useful for modeling other materials such as ceramics or brittle rocks;
- is designed for applications in which the behavior is dominated by tensile cracking;
- assumes that the compressive behavior is always linear elastic;
- must be used with the linear elastic material model (*Linear Elastic Behavior*), which also defines the material behavior completely prior to cracking;
- is most accurate in applications where the brittle behavior dominates such that the assumption that the material is linear elastic in compression is adequate;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- allows removal of elements based on a brittle failure criterion; and
- is defined in detail in *A cracking model for concrete and other brittle materials*.

See *Inelastic Behavior* for a discussion of the concrete models available in Abaqus.

Reinforcement

Reinforcement in concrete structures is typically provided by means of rebars. Rebars are one-dimensional strain theory elements (rods) that can be defined singly or embedded in oriented surfaces. Rebars are discussed in *Defining Rebar as an Element Property*. They are typically used with elastic-plastic material behavior and are superposed on a mesh of standard element types used to model the plain concrete. With this modeling approach, the concrete cracking behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some “tension stiffening” into the concrete cracking model to simulate load transfer across cracks through the rebar.

Cracking

Abaqus/Explicit uses a smeared crack model to represent the discontinuous brittle behavior in concrete. It does not track individual “macro” cracks: instead, constitutive calculations are performed independently at each material point of the finite element model. The presence of cracks enters into these calculations by the way in which the cracks affect the stress and material stiffness associated with the material point.

For simplicity of discussion in this section, the term “crack” is used to mean a direction in which cracking has been detected at the single material calculation point in question: the closest physical concept is that there exists a continuum of micro-cracks in the neighborhood of the point, oriented as determined by the model. The anisotropy introduced by cracking is assumed to be important in the simulations for which the model is intended.

Crack Directions

The Abaqus/Explicit cracking model assumes fixed, orthogonal cracks, with the maximum number of cracks at a material point limited by the number of direct stress components present at that material point of the finite element model (a maximum of three cracks in three-dimensional, plane strain, and axisymmetric problems; two cracks in plane stress and shell problems; and one crack in beam or truss problems). Internally, once cracks exist at a point, the component forms of all vector- and tensor-valued quantities are rotated so that they lie in the local system defined by the crack orientation vectors (the normals to the crack faces). The model ensures that these crack face normal vectors will be orthogonal, so that this local crack system is rectangular Cartesian. For output purposes you are offered results of stresses and strains in the global and/or local crack systems.

Crack Detection

A simple Rankine criterion is used to detect crack initiation. This criterion states that a crack forms when the maximum principal tensile stress exceeds the tensile strength of the brittle material. Although crack detection is based purely on Mode I fracture considerations, ensuing cracked behavior includes both Mode I (tension softening/stiffening) and Mode II (shear softening/retention) behavior, as described later.

As soon as the Rankine criterion for crack formation has been met, we assume that a first crack has formed. The crack surface is taken to be normal to the direction of the maximum tensile principal stress. Subsequent cracks may form with crack surface normals in the direction of maximum principal tensile stress that is orthogonal to the directions of any existing crack surface normals at the same point.

Cracking is irrecoverable in the sense that, once a crack has occurred at a point, it remains throughout the rest of the calculation. However, crack closing and reopening may take place along the directions of the crack surface normals. The model neglects any permanent strain associated with cracking; that is, it is assumed that the cracks can close completely when the stress across them becomes compressive.

Tension Stiffening

You can specify the postfailure behavior for direct straining across cracks by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure Stress-Strain Relation

In reinforced concrete the specification of postfailure behavior generally means giving the postfailure stress as a function of strain across the crack (*Figure 1*). In cases with little or no reinforcement, this introduces mesh sensitivity in the results, in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands.

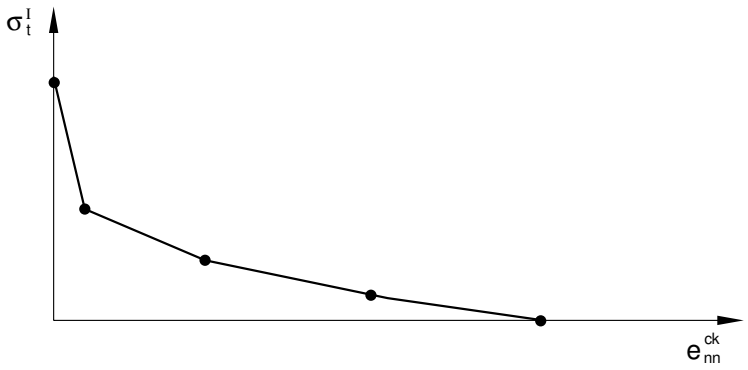


Figure 1: Postfailure stress-strain curve.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. In this case the interaction between the rebars and the concrete tends to mitigate this effect, provided that a reasonable amount of “tension stiffening” is introduced in the cracking model to simulate this interaction. This requires an estimate of the tension stiffening effect, which depends on factors such as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain about ten times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to each particular case. In static applications too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: **BRITTLE CRACKING*, TYPE=STRAIN

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Brittle Cracking: **Type: Strain**

Fracture Energy Cracking Criterion

When there is no reinforcement in significant regions of the model, the tension stiffening approach described above will introduce unreasonable mesh sensitivity into the results. However, it is generally accepted that Hillerborg's (1976) fracture energy proposal is adequate to allay the concern for many practical purposes.

Hillerborg defines the energy required to open a unit area of crack in Mode I (G_f^I) as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-*displacement* response rather than a stress-*strain* response. Under tension a concrete specimen will crack across some section; and its length, after it has been pulled apart sufficiently for most of the stress to be removed (so that the elastic strain is small), will be determined primarily by the opening at the crack, which does not depend on the specimen's length.

Implementation

In Abaqus/Explicit this fracture energy cracking model can be invoked by specifying the postfailure stress as a tabular function of displacement across the crack, as illustrated in *Figure 2*.

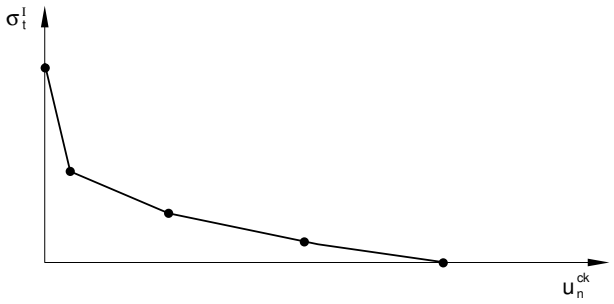


Figure 2: Postfailure stress-displacement curve.

Alternatively, the Mode I fracture energy, G_f^I, can be specified directly as a material property; in this case, define the failure stress, σ_{tu}^I, as a tabular function of the associated Mode I fracture energy. This model assumes a linear loss of strength after cracking (*Figure 3*).

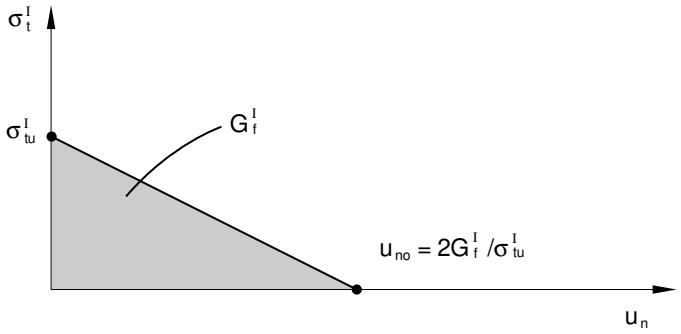


Figure 3: Postfailure stress-fracture energy curve.

The crack normal displacement at which complete loss of strength takes place is, therefore, $u_{n0} = 2G_f^I/\sigma_{tu}^I$.

Typical values of G_f^I range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2).

Input File Usage: Use the following option to specify the postfailure stress as a tabular function of displacement:

**BRITTLE CRACKING*, TYPE=DISPLACEMENT

Use the following option to specify the postfailure stress as a tabular function of the fracture energy:

**BRITTLE CRACKING*, TYPE=GFI

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Brittle Cracking: **Type: Displacement** or **GFI**

Characteristic Crack Length

The implementation of the stress-displacement concept in a finite element model requires the definition of a characteristic length associated with a material point. The characteristic crack length is based on the element

geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. We use this definition of the characteristic crack length because the direction in which cracks will occur is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect. Elements that are as close to square as possible are, therefore, recommended unless you can predict the direction in which cracks will form. Alternatively, this mesh dependency could be reduced by directly specifying the characteristic length as a function of the element topology and material orientation in user subroutine *VUCHARLENGTH* (see *Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit*).

Shear Retention Model

An important feature of the cracking model is that, whereas crack initiation is based on Mode I fracture only, postcracked behavior includes Mode II as well as Mode I. The Mode II shear behavior is based on the common observation that the shear behavior depends on the amount of crack opening. More specifically, the cracked shear modulus is reduced as the crack opens. Therefore, Abaqus/Explicit offers a shear retention model in which the postcracked shear stiffness is defined as a function of the opening strain across the crack; the shear retention model must be defined in the cracking model, and zero shear retention should not be used.

In these models the dependence is defined by expressing the postcracking shear modulus, G_c, as a fraction of the uncracked shear modulus:

$$G_c = \rho\left(\varepsilon_{nn}^{ck}\right) \quad G,$$

where G is the shear modulus of the uncracked material and the shear retention factor, $\rho\left(\varepsilon_{nn}^{ck}\right)$, depends on the crack opening strain, ε_{nn}^{ck}. You can specify this dependence in piecewise linear form, as shown in *Figure 4*.

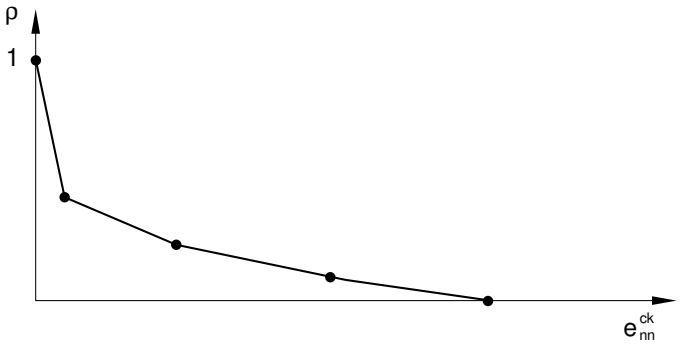


Figure 4: Piecewise linear form of the shear retention model.

Alternatively, shear retention can be defined in the power law form:

$$\rho\left(\varepsilon_{nn}^{ck}\right) = \left(1 - \frac{\varepsilon_{nn}^{ck}}{\varepsilon_{max}^{ck}}\right)^{p},$$

where p and ε_{max}^{ck} are material parameters. This form, shown in *Figure 5*, satisfies the requirements that $\rho \to 1$ as $\varepsilon_{nn}^{ck} \to 0$ (corresponding to the state before crack initiation) and $\rho \to 0$ as $\varepsilon_{nn}^{ck} \to \varepsilon_{max}^{ck}$ (corresponding to complete loss of aggregate interlock). See *A cracking model for concrete and other brittle materials* for a discussion of how shear retention is calculated in the case of two or more cracks.

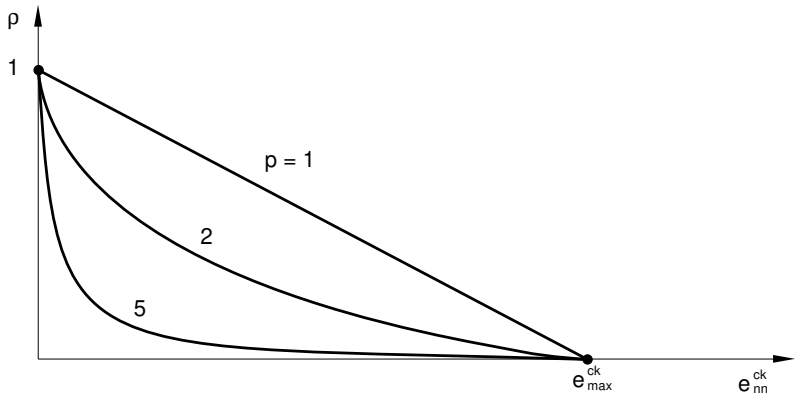


Figure 5: Power law form of the shear retention model.

Input File Usage: Use the following option to specify the piecewise linear form of the shear retention model:

**BRITTLE SHEAR*, TYPE=RETENTION FACTOR

Use the following option to specify the power law form of the shear retention model:

**BRITTLE SHEAR*, TYPE=POWER LAW

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Brittle Cracking: **Suboptions->Brittle Shear Type: Retention Factor** or **Power Law**

Calibration

One experiment, a uniaxial tension test, is required to calibrate the simplest version of the brittle cracking model. Other experiments may be required to gain accuracy in postfailure behavior.

Uniaxial Tension Test

This test is difficult to perform because it is necessary to have a very stiff testing machine to record the postcracking response. Quite often such equipment is not available; in this situation you must make an assumption about the tensile failure strength of the material and the postcracking response. For concrete the assumption usually made is that the tensile strength is 7–10% of the compressive strength. Uniaxial compression tests can be performed much more easily, so the compressive strength of concrete is usually known.

Postcracking Tensile Behavior

The values given for tension stiffening are a very important aspect of simulations using the Abaqus/Explicit brittle cracking model. The postcracking tensile response is highly dependent on the reinforcement present in the concrete. In simulations of unreinforced concrete, the tension stiffening models that are based on fracture energy concepts should be utilized. If reliable experimental data are not available, typical values that can be used

were discussed before: common values of G_f^I range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2). In simulations of reinforced concrete the stress-strain tension stiffening model should be used; the amount of tension stiffening depends on the reinforcement present, as discussed before. A reasonable starting point for relatively heavily reinforced

concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain about ten times the strain at failure. Since the strain at failure in standard concretes is typically 10^{-4}, this suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to each particular case.

Postcracking Shear Behavior

Calibration of the postcracking shear behavior requires combined tension and shear experiments, which are difficult to perform. If such test data are not available, a reasonable starting point is to assume that the shear retention factor, ρ, goes linearly to zero at the same crack opening strain used for the tension stiffening model.

Brittle Failure Criterion

You can define brittle failure of the material. When one, two, or all three local direct cracking strain (displacement) components at a material point reach the value defined as the failure strain (displacement), the material point fails and all the stress components are set to zero. An element is removed from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

If the postfailure relation is defined in terms of stress versus strain, the failure strain must be given as the failure criterion. If the postfailure relation is defined in terms of stress versus displacement or stress versus fracture energy, the failure displacement must be given as the failure criterion. The failure strain (displacement) can be specified as a function of temperature and/or predefined field variables.

You can control how many cracks at a material point must fail before the material point is considered to have failed; the default is one crack. The number of cracks that must fail can only be one for beam and truss elements; it cannot be greater than two for plane stress and shell elements; and it cannot be greater than three otherwise.

Input File Usage: **BRITTLE FAILURE*, CRACKS=*n*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Brittle Cracking: **Suboptions->Brittle Failure** and select **Failure Criteria**: **Unidirectional**, **Bidirectional**, or **Tridirectional** to indicate the number of cracks that must fail for the material point to fail.

Determining When to Use the Brittle Failure Criterion

The brittle failure criterion is a crude way of modeling failure in Abaqus/Explicit and should be used with care. The main motivation for including this capability is to help in computations where not removing an element that can no longer carry stress may lead to excessive distortion of that element and subsequent premature termination of the simulation. For example, in a monotonically loaded structure whose failure mechanism is expected to be dominated by a single tensile macrofracture (Mode I cracking), it may be reasonable to use the brittle failure criterion to remove elements. On the other hand, the fact that the brittle material loses its ability to carry tensile stress does not preclude it from withstanding compressive stress; therefore, it may not be appropriate to remove elements if the material is expected to carry compressive loads after it has failed in tension. An example may be a shear wall subjected to cyclic loading as a result of some earthquake excitation; in this case cracks that develop completely under tensile stress will be able to carry compressive stress when load reversal takes place.

Thus, the effective use of the brittle failure criterion relies on you having some knowledge of the structural behavior and potential failure mechanism. The use of the brittle failure criterion based on an incorrect user assumption of the failure mechanism will generally result in an incorrect simulation.

Selecting the Number of Cracks That Must Fail before the Material Point Is Considered to Have Failed

When you define brittle failure, you can control how many cracks must open to beyond the failure value before a material point is considered to have failed. The default number of cracks (one) should be used for most structural applications where failure is dominated by Mode I type cracking. However, there are cases in which you should specify a higher number because multiple cracks need to form to develop the eventual failure mechanism. One example may be an unreinforced, deep concrete beam where the failure mechanism is dominated by shear; in this case it is possible that two cracks need to form at each material point for the shear failure mechanism to develop.

Again, the appropriate choice of the number of cracks that must fail relies on your knowledge of the structural and failure behaviors.

Using Brittle Failure with Rebar

It is possible to use the brittle failure criterion in brittle cracking elements for which rebar are also defined; the obvious application is the modeling of reinforced concrete. When such elements fail according to the brittle failure criterion, the brittle cracking contribution to the element stress carrying capacity is removed but the rebar contribution to the element stress carrying capacity is not removed. However, if you also include shear failure in the rebar material definition, the rebar contribution to the element stress carrying capacity will also be removed if the shear failure criterion specified for the rebar is satisfied. This allows the modeling of progressive failure of an under-reinforced concrete structure where the concrete fails first followed by ductile failure of the reinforcement.

Elements

Abaqus/Explicit offers a variety of elements for use with the cracking model: truss; shell; two-dimensional beam; and plane stress, plane strain, axisymmetric, and three-dimensional continuum elements. The model cannot be used with pipe and three-dimensional beam elements. Plane triangular, triangular prism, and tetrahedral elements are not recommended for use in reinforced concrete analysis since these elements do not support the use of rebar.

Output

In addition to the standard output identifiers available in Abaqus/Explicit (see *Abaqus/Explicit Output Variable Identifiers*), the following output variables relate directly to material points that use the brittle cracking model:

CKE	All cracking strain components.
CKLE	All cracking strain components in local crack axes.
CKEMAG	Cracking strain magnitude.
CKLS	All stress components in local crack axes.
CRACK	Crack orientations.
CKSTAT	Crack status of each crack.
STATUS	Status of element (brittle failure model). The status of an element is 1.0 if the element is active and 0.0 if the element is not.

STATUSMP Status of each material point in the element (brittle failure model). The status of a material point is 1.0 if the material point is active and 0.0 if it is not.

References

- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," *Cement and Concrete Research*, vol. 6, pp. 773–782, 1976.

Concrete Damaged Plasticity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *Inelastic Behavior*
- **CONCRETE DAMAGED PLASTICITY*
- **CONCRETE TENSION STIFFENING*
- **CONCRETE COMPRESSION HARDENING*
- **CONCRETE TENSION DAMAGE*
- **CONCRETE COMPRESSION DAMAGE*
- **CONCRETE FAILURE*
- *Defining concrete damaged plasticity*

Overview

The concrete damaged plasticity model in Abaqus:

- provides a general capability for modeling concrete and other quasi-brittle materials in all types of structures (beams, trusses, shells, and solids);
- uses concepts of isotropic damaged elasticity in combination with isotropic tensile and compressive plasticity to represent the inelastic behavior of concrete;
- can be used for plain concrete, even though it is intended primarily for the analysis of reinforced concrete structures;
- can be used with rebar to model concrete reinforcement;
- is designed for applications in which concrete is subjected to monotonic, cyclic, and/or dynamic loading under low confining pressures;
- consists of the combination of nonassociated multi-hardening plasticity and scalar (isotropic) damaged elasticity to describe the irreversible damage that occurs during the fracturing process;
- allows user control of stiffness recovery effects during cyclic load reversals;
- allows removal of elements based on material failure criteria;
- can be defined to be sensitive to the rate of straining;
- can be used in conjunction with a viscoplastic regularization of the constitutive equations in Abaqus/Standard to improve the convergence rate in the softening regime;
- requires that the elastic behavior of the material be isotropic and linear (see *Defining Isotropic Elasticity*); and
- is defined in detail in *Damaged plasticity model for concrete and other quasi-brittle materials*.

See *Inelastic Behavior* for a discussion of the concrete models available in Abaqus.

Mechanical Behavior

The model is a continuum, plasticity-based, damage model for concrete. It assumes that the main two failure mechanisms are tensile cracking and compressive crushing of the concrete material. The evolution of the yield (or failure) surface is controlled by two hardening variables, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$, linked to failure mechanisms under

tension and compression loading, respectively. We refer to $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ as tensile and compressive equivalent plastic strains, respectively. The following sections discuss the main assumptions about the mechanical behavior of concrete.

Uniaxial Tension and Compression Stress Behavior

The model assumes that the uniaxial tensile and compressive response of concrete is characterized by damaged plasticity, as shown in *Figure 1*.

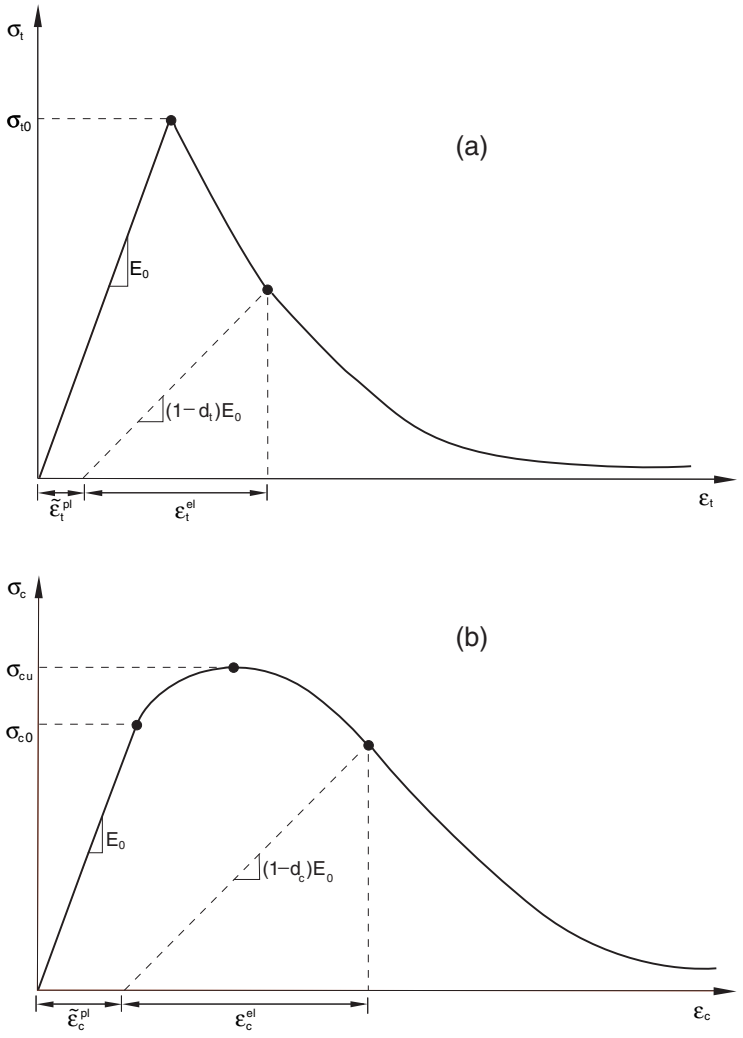


Figure 1: Response of concrete to uniaxial loading in tension (a) and compression (b).

Under uniaxial tension the stress-strain response follows a linear elastic relationship until the value of the failure stress, σ_{t0}, is reached. The failure stress corresponds to the onset of micro-cracking in the concrete material. Beyond the failure stress the formation of micro-cracks is represented macroscopically with a softening stress-strain response, which induces strain localization in the concrete structure. Under uniaxial compression the response is linear until the value of initial yield, σ_{c0}. In the plastic regime the response is typically characterized by stress hardening followed by strain softening beyond the ultimate stress, σ_{cu}. This representation, although somewhat simplified, captures the main features of the response of concrete.

It is assumed that the uniaxial stress-strain curves can be converted into stress versus plastic-strain curves. (This conversion is performed automatically by Abaqus from the user-provided stress versus "inelastic" strain data, as explained below.) Thus,

$$\sigma_t = \sigma_t\left(\tilde{\varepsilon}_t^{pl}, \dot{\tilde{\varepsilon}}_t^{pl}, \theta, f_i\right),$$
$$\sigma_c = \sigma_c\left(\tilde{\varepsilon}_c^{pl}, \dot{\tilde{\varepsilon}}_c^{pl}, \theta, f_i\right),$$

where the subscripts t and c refer to tension and compression, respectively; $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are the equivalent plastic strains, $\dot{\tilde{\varepsilon}}_t^{pl}$ and $\dot{\tilde{\varepsilon}}_c^{pl}$ are the equivalent plastic strain rates, θ is the temperature, and $f_i, (i = 1, 2, \ldots)$ are other predefined field variables.

As shown in *Figure 1*, when the concrete specimen is unloaded from any point on the strain softening branch of the stress-strain curves, the unloading response is weakened: the elastic stiffness of the material appears to be damaged (or degraded). The degradation of the elastic stiffness is characterized by two damage variables, d_t and d_c, which are assumed to be functions of the plastic strains, temperature, and field variables:

$$d_t = d_t\left(\tilde{\varepsilon}_t^{pl}, \theta, f_i\right); \quad 0 \le d_t \le 1,$$
$$d_c = d_c\left(\tilde{\varepsilon}_c^{pl}, \theta, f_i\right); \quad 0 \le d_c \le 1.$$

The damage variables can take values from zero, representing the undamaged material, to one, which represents total loss of strength.

If E_0 is the initial (undamaged) elastic stiffness of the material, the stress-strain relations under uniaxial tension and compression loading are, respectively:

$$\sigma_t = (1 - d_t) E_0\left(\varepsilon_t - \tilde{\varepsilon}_t^{pl}\right),$$
$$\sigma_c = (1 - d_c) E_0\left(\varepsilon_c - \tilde{\varepsilon}_c^{pl}\right).$$

We define the "effective" tensile and compressive cohesion stresses as

$$\overline{\sigma}_t = \frac{\sigma_t}{(1 - d_t)} = E_0\left(\varepsilon_t - \tilde{\varepsilon}_t^{pl}\right),$$
$$\overline{\sigma}_c = \frac{\sigma_c}{(1 - d_c)} = E_0\left(\varepsilon_c - \tilde{\varepsilon}_c^{pl}\right).$$

The effective cohesion stresses determine the size of the yield (or failure) surface.

Uniaxial Cyclic Behavior

Under uniaxial cyclic loading conditions the degradation mechanisms are quite complex, involving the opening and closing of previously formed micro-cracks, as well as their interaction. Experimentally, it is observed that there is some recovery of the elastic stiffness as the load changes sign during a uniaxial cyclic test. The stiffness recovery effect, also known as the "unilateral effect," is an important aspect of the concrete behavior under cyclic loading. The effect is usually more pronounced as the load changes from tension to compression, causing tensile cracks to close, which results in the recovery of the compressive stiffness.

The concrete damaged plasticity model assumes that the reduction of the elastic modulus is given in terms of a scalar degradation variable d as

$$E = (1 - d) E_0,$$

where E_0 is the initial (undamaged) modulus of the material.

This expression holds both in the tensile ($\sigma_{11} > 0$) and the compressive ($\sigma_{11} < 0$) sides of the cycle. The stiffness degradation variable, d, is a function of the stress state and the uniaxial damage variables, d_t and d_c. For the uniaxial cyclic conditions Abaqus assumes that

$$(1-d) = (1-s_t d_c)\,(1-s_c d_t)\,,$$

where s_t and s_c are functions of the stress state that are introduced to model stiffness recovery effects associated with stress reversals. They are defined according to

$$\begin{aligned} s_t &= 1 - w_t r^*(\sigma_{11}); \quad 0 \le w_t \le 1, \\ s_c &= 1 - w_c\,(1 - r^*(\sigma_{11})); \quad 0 \le w_c \le 1, \end{aligned}$$

where

$$r^*(\sigma_{11}) = H(\sigma_{11}) = \begin{cases} 1 \text{ if } \sigma_{11} > 0 \\ 0 \text{ if } \sigma_{11} < 0 \end{cases}$$

The weight factors w_t and w_c, which are assumed to be material properties, control the recovery of the tensile and compressive stiffness upon load reversal. To illustrate this, consider the example in *Figure 2*, where the load changes from tension to compression.

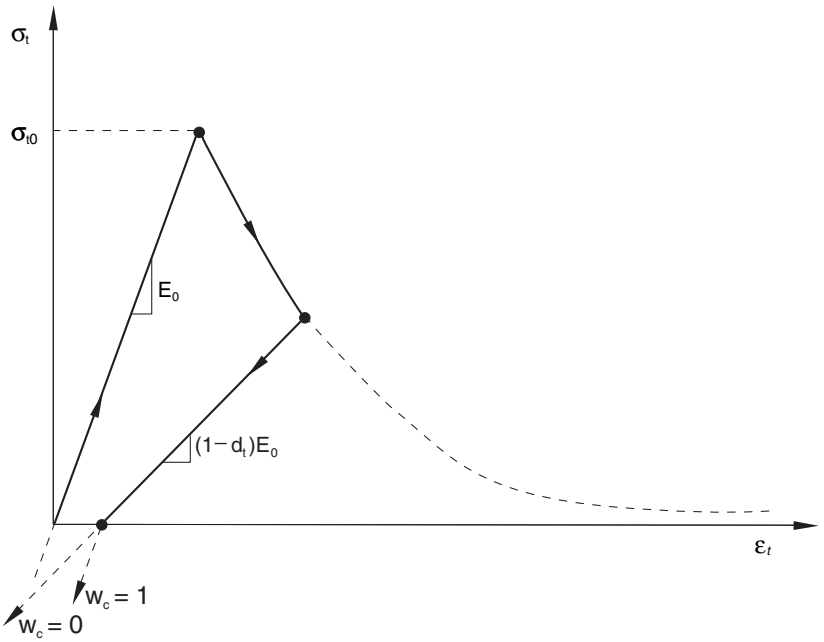


Figure 2: Illustration of the effect of the compression stiffness recovery parameter w_c.

Assume that there was no previous compressive damage (crushing) in the material; that is, $\tilde{\varepsilon}_c^{pl} = 0$ and $d_c = 0$. Then

$$(1-d) = (1-s_c d_t) = (1-(1-w_c\,(1-r^*))\,d_t)\,.$$

- In tension ($\sigma_{11} > 0$), $r^* = 1$; therefore, $d = d_t$ as expected.
- In compression ($\sigma_{11} < 0$), $r^* = 0$, and $d = (1-w_c)\,d_t$. If $w_c = 1$, then $d = 0$; therefore, the material fully recovers the compressive stiffness (which in this case is the initial undamaged stiffness, $E = E_0$). If, on the

other hand, $w_c = 0$, then $d = d_t$ and there is no stiffness recovery. Intermediate values of w_c result in partial recovery of the stiffness.

Multiaxial Behavior

The stress-strain relations for the general three-dimensional multiaxial condition are given by the scalar damage elasticity equation:

$$\boldsymbol{\sigma} = (1-d)\,\mathbf{D}_0^{el} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}\right),$$

where $\mathbf{D}_0^{el}$ is the initial (undamaged) elasticity matrix.

The previous expression for the scalar stiffness degradation variable, d, is generalized to the multiaxial stress case by replacing the unit step function $r^*(\sigma_{11})$ with a multiaxial stress weight factor, $r(\hat{\boldsymbol{\sigma}})$, defined as

$$r(\hat{\boldsymbol{\sigma}}) = \frac{\sum_{i=1}^{3}\langle\hat{\sigma}_i\rangle}{\sum_{i=1}^{3}|\hat{\sigma}_i|}; \quad 0 \le r(\hat{\boldsymbol{\sigma}}) \le 1,$$

where $\hat{\sigma}_i \quad (i = 1,2,3)$ are the principal stress components. The Macauley bracket $\langle\cdot\rangle$ is defined by $\langle x\rangle = \frac{1}{2}(|x| + x)$.

See *Damaged plasticity model for concrete and other quasi-brittle materials* for further details of the constitutive model.

Reinforcement

In Abaqus reinforcement in concrete structures is typically provided by means of rebars, which are one-dimensional rods that can be defined singly or embedded in oriented surfaces. Rebars are typically used with metal plasticity models to describe the behavior of the rebar material and are superposed on a mesh of standard element types used to model the concrete.

With this modeling approach, the concrete behavior is considered independently of the rebar. Effects associated with the rebar/concrete interface, such as bond slip and dowel action, are modeled approximately by introducing some "tension stiffening" into the concrete modeling to simulate load transfer across cracks through the rebar. Details regarding tension stiffening are provided below.

Defining the rebar can be tedious in complex problems, but it is important that this be done accurately since it may cause an analysis to fail due to lack of reinforcement in key regions of a model. See *Defining Rebar as an Element Property* for more information regarding rebars.

Defining Tension Stiffening

The postfailure behavior for direct straining is modeled with tension stiffening, which allows you to define the strain-softening behavior for cracked concrete. This behavior also allows for the effects of the reinforcement interaction with concrete to be simulated in a simple manner. Tension stiffening is required in the concrete damaged plasticity model. You can specify tension stiffening by means of a postfailure stress-strain relation or by applying a fracture energy cracking criterion.

Postfailure Stress-Strain Relation

In reinforced concrete the specification of postfailure behavior generally means giving the postfailure stress as a function of cracking strain, $\tilde{\varepsilon}_t^{ck}$. The cracking strain is defined as the total strain minus the elastic strain corresponding to the undamaged material; that is, $\tilde{\varepsilon}_t^{ck} = \varepsilon_t - \varepsilon_{0t}^{el}$, where $\varepsilon_{0t}^{el} = \sigma_t / E_0$, as illustrated in *Figure*

3. To avoid potential numerical problems, Abaqus enforces a lower limit on the postfailure stress equal to one-hundreth of the initial failure stress: $\sigma_t \geq \sigma_{t0}/100$.

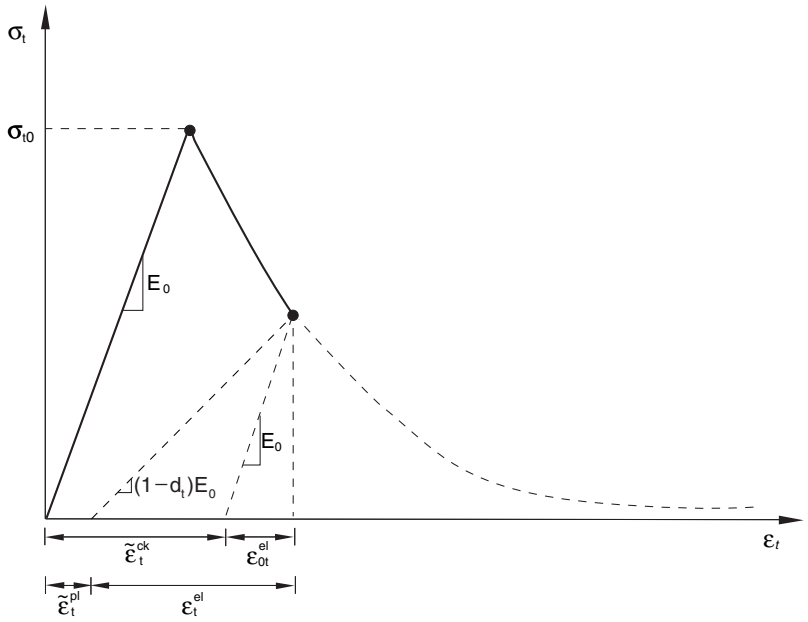


Figure 3: Illustration of the definition of the cracking strain $\tilde{\varepsilon}_t^{ck}$ used for the definition of tension stiffening data.

Tension stiffening data are given in terms of the cracking strain, $\tilde{\varepsilon}_t^{ck}$. When unloading data are available, the data are provided to Abaqus in terms of tensile damage curves, $d_t - \tilde{\varepsilon}_t^{ck}$, as discussed below. Abaqus automatically converts the cracking strain values to plastic strain values using the relationship

$$\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck} - \frac{d_t}{(1-d_t)}\frac{\sigma_t}{E_0}.$$

Abaqus will issue an error message if the calculated plastic strain values are negative and/or decreasing with increasing cracking strain, which typically indicates that the tensile damage curves are incorrect. In the absence of tensile damage $\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck}$.

In cases with little or no reinforcement, the specification of a postfailure stress-strain relation introduces mesh sensitivity in the results, in the sense that the finite element predictions do not converge to a unique solution as the mesh is refined because mesh refinement leads to narrower crack bands. This problem typically occurs if cracking failure occurs only at localized regions in the structure and mesh refinement does not result in the formation of additional cracks. If cracking failure is distributed evenly (either due to the effect of rebar or due to the presence of stabilizing elastic material, as in the case of plate bending), mesh sensitivity is less of a concern.

In practical calculations for reinforced concrete, the mesh is usually such that each element contains rebars. The interaction between the rebars and the concrete tends to reduce the mesh sensitivity, provided that a reasonable amount of tension stiffening is introduced in the concrete model to simulate this interaction. This requires an estimate of the tension stiffening effect, which depends on such factors as the density of reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the mesh. A reasonable starting point for relatively heavily reinforced concrete modeled with a fairly detailed mesh is to assume that the strain softening after failure reduces the stress linearly to zero at a total strain of about 10 times the strain at failure. The strain at failure in standard concretes is typically 10^{-4},

which suggests that tension stiffening that reduces the stress to zero at a total strain of about 10^{-3} is reasonable. This parameter should be calibrated to a particular case.

The choice of tension stiffening parameters is important since, generally, more tension stiffening makes it easier to obtain numerical solutions. Too little tension stiffening will cause the local cracking failure in the concrete to introduce temporarily unstable behavior in the overall response of the model. Few practical designs exhibit such behavior, so that the presence of this type of response in the analysis model usually indicates that the tension stiffening is unreasonably low.

Input File Usage: **CONCRETE TENSION STIFFENING*, TYPE=STRAIN (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Tensile Behavior**: **Type: Strain**

Fracture Energy Cracking Criterion

When there is no reinforcement in significant regions of the model, the tension stiffening approach described above will introduce unreasonable mesh sensitivity into the results. However, it is generally accepted that Hillerborg's (1976) fracture energy proposal is adequate to allay the concern for many practical purposes.

Hillerborg defines the energy required to open a unit area of crack, G_f, as a material parameter, using brittle fracture concepts. With this approach the concrete's brittle behavior is characterized by a stress-displacement response rather than a stress-strain response. Under tension a concrete specimen will crack across some section. After it has been pulled apart sufficiently for most of the stress to be removed (so that the undamaged elastic strain is small), its length will be determined primarily by the opening at the crack. The opening does not depend on the specimen's length.

This fracture energy cracking model can be invoked by specifying the postfailure stress as a tabular function of cracking displacement, as shown in *Figure 4*.

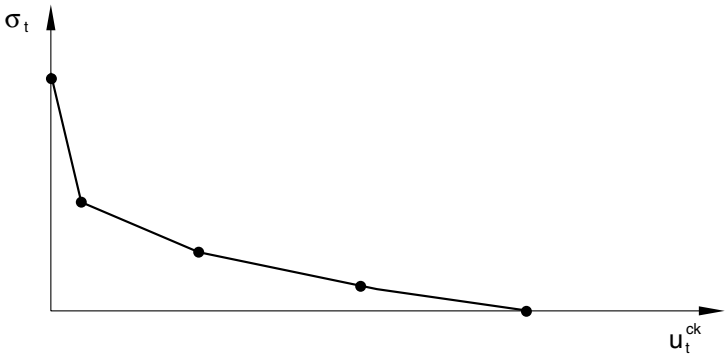


Figure 4: Postfailure stress-displacement curve.

Alternatively, the fracture energy, G_f, can be specified directly as a material property; in this case, define the failure stress, σ_{t0}, as a tabular function of the associated fracture energy. This model assumes a linear loss of strength after cracking, as shown in *Figure 5*.

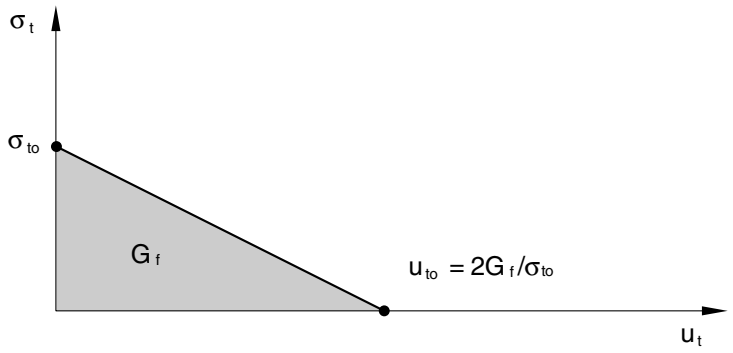


Figure 5: Postfailure stress-fracture energy curve.

The cracking displacement at which complete loss of strength takes place is, therefore, $u_{t0} = 2G_f/\sigma_{t0}$. Typical values of G_f range from 40 N/m (0.22 lb/in) for a typical construction concrete (with a compressive strength of approximately 20 MPa, 2850 lb/in^2) to 120 N/m (0.67 lb/in) for a high-strength concrete (with a compressive strength of approximately 40 MPa, 5700 lb/in^2).

If tensile damage, d_t, is specified, Abaqus automatically converts the cracking displacement values to "plastic" displacement values using the relationship

$$u_t^{pl} = u_t^{ck} - \frac{d_t}{(1-d_t)}\frac{\sigma_t l_0}{E_0},$$

where l_0 is the specimen length. By default, Abaqus assumes l_0 to be one unit length ($l_0 = 1$). However, it is recommended that you specify this value directly in the material definition to properly account for the length units.

Input File Usage: **CONCRETE DAMAGED PLASTICITY*, REF LENGTH=l_0

Abaqus/CAE Usage: Specifying the specimen length is not supported in Abaqus/CAE.

Implementation

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length associated with an integration point. The characteristic crack length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic crack length is used because the direction in which cracking occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect, and elements that have aspect ratios close to one are recommended. Alternatively, this mesh dependency could be reduced by directly specifying the characteristic length as a function of the element topology and material orientation in user subroutine *VUCHARLENGTH* (see *Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit*).

Input File Usage: Use the following option to specify the postfailure stress as a tabular function of displacement:

**CONCRETE TENSION STIFFENING*, TYPE=DISPLACEMENT

Use the following option to specify the postfailure stress as a tabular function of the fracture energy:

**CONCRETE TENSION STIFFENING*, TYPE=GFI

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Tensile Behavior**: **Type: Displacement** or **GFI**

Defining Compressive Behavior

You can define the stress-strain behavior of plain concrete in uniaxial compression outside the elastic range.

Compressive stress data are provided as a tabular function of inelastic (or crushing) strain, $\tilde{\varepsilon}_c^{in}$, and, if desired, strain rate, temperature, and field variables. Positive (absolute) values should be given for the compressive stress and strain. The stress-strain curve can be defined beyond the ultimate stress, into the strain-softening regime.

Hardening data are given in terms of an inelastic strain, $\tilde{\varepsilon}_c^{in}$, instead of plastic strain, $\tilde{\varepsilon}_c^{pl}$. The compressive inelastic strain is defined as the total strain minus the elastic strain corresponding to the undamaged material, $\tilde{\varepsilon}_c^{in} = \varepsilon_c - \varepsilon_{0c}^{el}$, where $\varepsilon_{0c}^{el} = \sigma_c / E_0$, as illustrated in *Figure 6*.

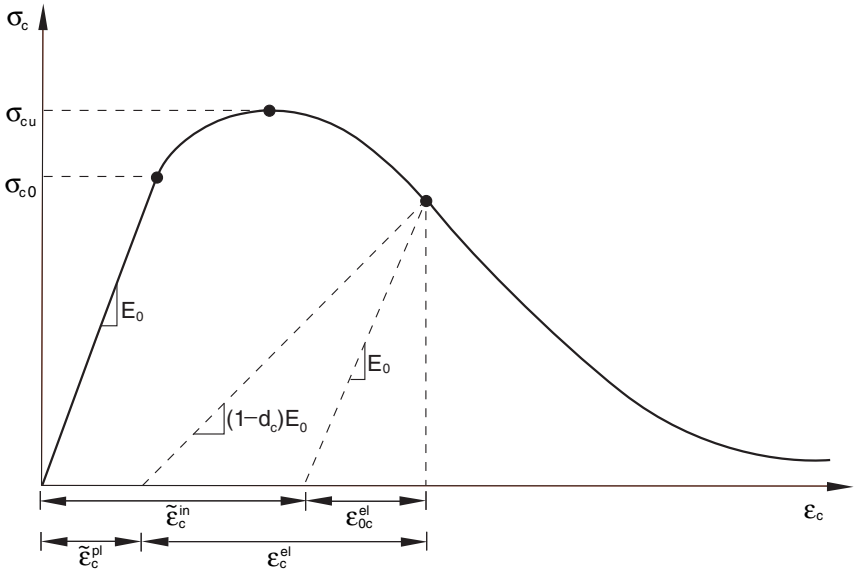


Figure 6: Definition of the compressive inelastic (or crushing) strain $\tilde{\varepsilon}_c^{in}$ used for the definition of compression hardening data.

Unloading data are provided to Abaqus in terms of compressive damage curves, $d_c - \tilde{\varepsilon}_c^{in}$, as discussed below. Abaqus automatically converts the inelastic strain values to plastic strain values using the relationship

$$\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in} - \frac{d_c}{(1-d_c)} \frac{\sigma_c}{E_0}.$$

Abaqus will issue an error message if the calculated plastic strain values are negative and/or decreasing with increasing inelastic strain, which typically indicates that the compressive damage curves are incorrect. In the absence of compressive damage $\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in}$.

Input File Usage: **CONCRETE COMPRESSION HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Compressive Behavior**

Defining Damage and Stiffness Recovery

Damage, d_t and/or d_c, can be specified in tabular form. (If damage is not specified, the model behaves as a plasticity model; consequently, $\tilde{\varepsilon}_t^{pl} = \tilde{\varepsilon}_t^{ck}$ and $\tilde{\varepsilon}_c^{pl} = \tilde{\varepsilon}_c^{in}$.)

In Abaqus the damage variables are treated as non-decreasing material point quantities. At any increment during the analysis, the new value of each damage variable is obtained as the maximum between the value at the end of the previous increment and the value corresponding to the current state (interpolated from the user-specified tabular data); that is,

$$d_t|_{t+\Delta t} = \max\left\{d_t|_t, d_t\left(\tilde{\varepsilon}_t^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t}\right)\right\},$$
$$d_c|_{t+\Delta t} = \max\left\{d_c|_t, d_c\left(\tilde{\varepsilon}_c^{pl}|_{t+\Delta t}, \theta|_{t+\Delta t}, f_i|_{t+\Delta t}\right)\right\}.$$

The choice of the damage properties is important since, generally, excessive damage may have a critical effect on the rate of convergence. It is recommended to avoid using values of the damage variables above 0.99, which corresponds to a 99% reduction of the stiffness.

It is strongly recommended that you specify the tabular definition of tensile and compressive damage variables for the same values of cracking and inelastic strains/displacements as those used in the tabular definitions of tension stiffening and compressive behavior, respectively. When the tensile stiffening response is defined directly in terms of failure stress and fracture energy (G_f), the definition of the tension damage variable should be such that it increases linearly as a function of the cracking displacement.

Tensile Damage

You can define the uniaxial tension damage variable, d_t, as a tabular function of either cracking strain or cracking displacement.

Input File Usage: Use the following option to specify tensile damage as a function of cracking strain:

**CONCRETE TENSION DAMAGE*, TYPE=STRAIN (default)

Use the following option to specify tensile damage as a function of cracking displacement:

**CONCRETE TENSION DAMAGE*, TYPE=DISPLACEMENT

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Tensile Behavior**: **Suboptions->Tension Damage**:
Type: Strain or **Displacement**

Compressive Damage

You can define the uniaxial compression damage variable, d_c, as a tabular function of inelastic (crushing) strain.

Input File Usage: **CONCRETE COMPRESSION DAMAGE*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Compressive Behavior**: **Suboptions->Compression Damage**

Stiffness Recovery

As discussed above, stiffness recovery is an important aspect of the mechanical response of concrete under cyclic loading. Abaqus allows direct user specification of the stiffness recovery factors w_t and w_c.

The experimental observation in most quasi-brittle materials, including concrete, is that the compressive stiffness is recovered upon crack closure as the load changes from tension to compression. On the other hand, the tensile stiffness is not recovered as the load changes from compression to tension once crushing micro-cracks have developed. This behavior, which corresponds to $w_t = 0$ and $w_c = 1$, is the default used by Abaqus. *Figure 7* illustrates a uniaxial load cycle assuming the default behavior.

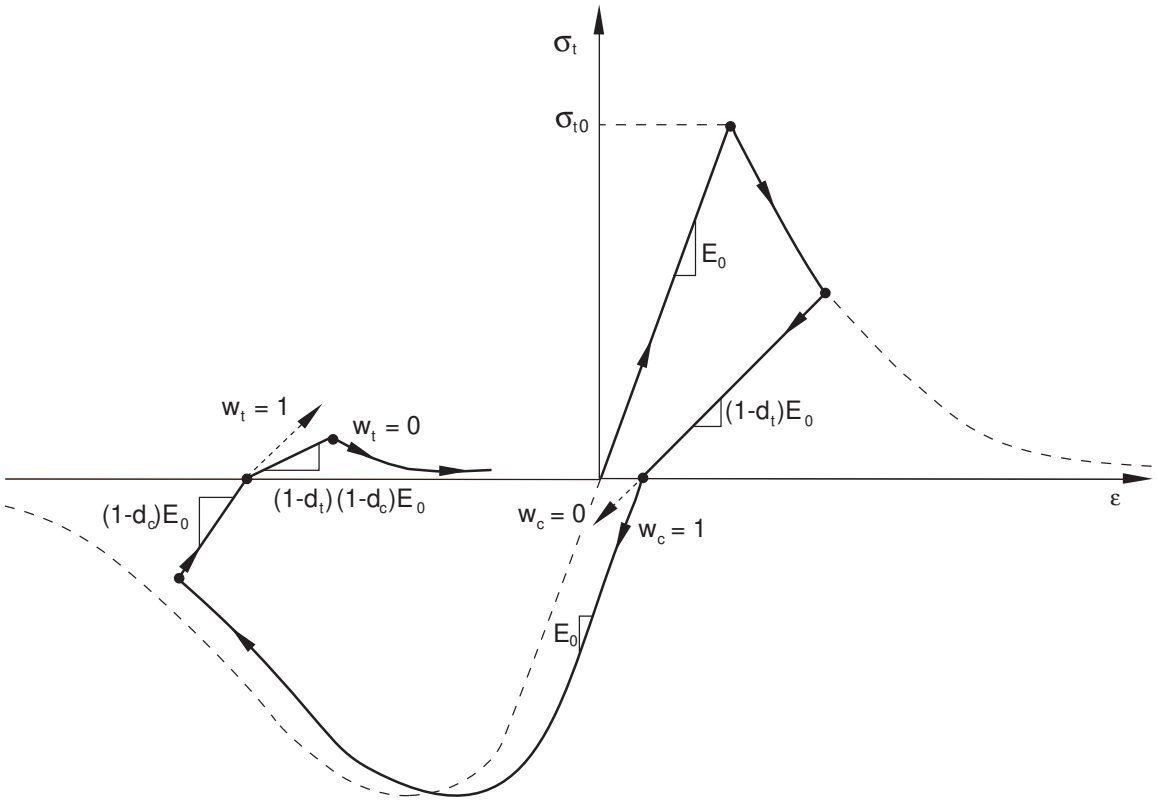


Figure 7: Uniaxial load cycle (tension-compression-tension) assuming default values for the stiffness recovery factors: $w_t = 0$ and $w_c = 1$.

Input File Usage: Use the following option to specify the compression stiffness recovery factor, w_c:

**CONCRETE TENSION DAMAGE*, COMPRESSION RECOVERY=w_c

Use the following option to specify the tension stiffness recovery factor, w_t:

**CONCRETE COMPRESSION DAMAGE*, TENSION RECOVERY=w_t

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**:
Tensile Behavior: **Suboptions->Tension Damage**: **Compression recovery:** w_c
Compressive Behavior: **Suboptions->Compression Damage**: **Tension recovery:** w_t

Defining Material Failure Criteria in Abaqus/Explicit

In Abaqus/Explicit you can define material failure for the concrete damaged plasticity model based on the following criteria:

- Tensile cracking strain criterion, $\tilde{\varepsilon}_t^{ck} \geq \left(\tilde{\varepsilon}_t^{ck}\right)_f$, or tensile cracking displacement criterion, $u_t^{ck} \geq \left(u_t^{ck}\right)_f$;
- Compressive inelastic strain criterion, $\tilde{\varepsilon}_c^{in} \geq \left(\tilde{\varepsilon}_c^{in}\right)_f$;
- Tensile damage variable criterion, $d_t \geq \left(d_t\right)_f$, if a tensile damage behavior is defined; and
- Compressive damage variable criterion, $d_c \geq \left(d_c\right)_f$, if a compressive damage behavior is defined.

If you specify a positive value at failure for the quantity of a criterion, the criterion is considered during the analysis. When any criterion at a material point is met, the material point fails and all stress components are set to zero. By default, an element is deleted from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Input File Usage: Use the following option to specify material failure criteria and define tensile failure based on a tensile cracking strain criterion:

**CONCRETE FAILURE*, TYPE=STRAIN (default)

Use the following option to specify material failure criteria and define tensile failure based on a tensile cracking displacement criterion:

**CONCRETE FAILURE*, TYPE=DISPLACEMENT

Abaqus/CAE Usage: Specifying material failure criteria is not supported in Abaqus/CAE.

Rate Dependence

The rate-sensitive behavior of quasi-brittle materials is mainly connected to the retardation effects that high strain rates have on the growth of micro-cracks. The effect is usually more pronounced under tensile loading. As the strain rate increases, the stress-strain curves exhibit decreasing nonlinearity as well as an increase in the peak strength. You can specify tension stiffening as a tabular function of cracking strain (or displacement) rate, and you can specify compression hardening data as a tabular function of inelastic strain rate.

Input File Usage: Use the following options:

**CONCRETE TENSION STIFFENING*
**CONCRETE COMPRESSION HARDENING*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**:
Tensile Behavior: **Use strain-rate-dependent data**
Compressive Behavior: **Use strain-rate-dependent data**

Concrete Plasticity

You can define flow potential, yield surface, and in Abaqus/Standard viscosity parameters for the concrete damaged plasticity material model.

Input File Usage: **CONCRETE DAMAGED PLASTICITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Concrete Damaged Plasticity**: **Plasticity**

Effective Stress Invariants

The effective stress is defined as

$$\overline{\boldsymbol{\sigma}} = \mathbf{D}_0^{el} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{pl}\right).$$

The plastic flow potential function and the yield surface make use of two stress invariants of the effective stress tensor, namely the hydrostatic pressure stress,

$$\bar{p} = -\tfrac{1}{3}\,\mathrm{trace}\,\left(\overline{\boldsymbol{\sigma}}\right),$$

and the Mises equivalent effective stress,

$$\bar{q} = \sqrt{\tfrac{3}{2}\left(\overline{\mathbf{S}} : \overline{\mathbf{S}}\right)},$$

where $\overline{\mathbf{S}}$ is the effective stress deviator, defined as

$$\overline{\mathbf{S}} = \overline{\boldsymbol{\sigma}} + \bar{p}\mathbf{I}.$$

Plastic Flow

The concrete damaged plasticity model assumes nonassociated potential plastic flow. The flow potential G used for this model is the Drucker-Prager hyperbolic function:

$$G = \sqrt{\left(\epsilon\sigma_{t0}\tan\psi\right)^2 + \bar{q}^2} - \bar{p}\tan\psi,$$

where

$\psi\left(\theta, f_i\right)$

is the dilation angle measured in the p–q plane at high confining pressure;

$\sigma_{t0}\left(\theta, f_i\right) = \sigma_t\big|_{\tilde{\varepsilon}_t^{pl}=0, \dot{\tilde{\varepsilon}}_t^{pl}=0}$

is the uniaxial tensile stress at failure, taken from the user-specified tension stiffening data; and

$\epsilon\left(\theta, f_i\right)$

is a parameter, referred to as the eccentricity, that defines the rate at which the function approaches the asymptote (the flow potential tends to a straight line as the eccentricity tends to zero).

This flow potential, which is continuous and smooth, ensures that the flow direction is always uniquely defined. The function approaches the linear Drucker-Prager flow potential asymptotically at high confining pressure stress and intersects the hydrostatic pressure axis at 90°. See *Models for granular or polymer behavior* for further discussion of this potential.

The default flow potential eccentricity is $\epsilon = 0.1$, which implies that the material has almost the same dilation angle over a wide range of confining pressure stress values. Increasing the value of ϵ provides more curvature to the flow potential, implying that the dilation angle increases more rapidly as the confining pressure decreases. Values of ϵ that are significantly less than the default value may lead to convergence problems if the material is subjected to low confining pressures because of the very tight curvature of the flow potential locally where it intersects the *p*-axis.

Yield Function

The model makes use of the yield function of Lubliner et. al. (1989), with the modifications proposed by Lee and Fenves (1998) to account for different evolution of strength under tension and compression. The evolution of the yield surface is controlled by the hardening variables, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$. In terms of effective stresses, the yield function takes the form

$$F = \frac{1}{1-\alpha}\left(\bar{q} - 3\alpha\bar{p} + \beta\left(\tilde{\varepsilon}^{pl}\right)\left\langle\hat{\bar{\sigma}}_{\max}\right\rangle - \gamma\left\langle-\hat{\bar{\sigma}}_{\max}\right\rangle\right) - \bar{\sigma}_c\left(\tilde{\varepsilon}_c^{pl}\right) = 0,$$

with

$$\alpha = \frac{\left(\sigma_{b0}/\sigma_{c0}\right) - 1}{2\left(\sigma_{b0}/\sigma_{c0}\right) - 1};\ 0 \le \alpha \le 0.5,$$

$$\beta = \frac{\bar{\sigma}_c\left(\tilde{\varepsilon}_c^{pl}\right)}{\bar{\sigma}_t\left(\tilde{\varepsilon}_t^{pl}\right)}(1-\alpha) - (1+\alpha),$$

$$\gamma = \frac{3\left(1-K_c\right)}{2K_c - 1}.$$

Here,

$\hat{\bar{\sigma}}_{\max}$

is the maximum principal effective stress;

σ_{b0}/σ_{c0}

is the ratio of initial equibiaxial compressive yield stress to initial uniaxial compressive yield stress (the default value is **1.16**);

K_c

is the ratio of the second stress invariant on the tensile meridian, $q_{(\mathrm{TM})}$, to that on the compressive meridian, $q_{(\mathrm{CM})}$, at initial yield for any given value of the pressure invariant *p* such that the maximum principal stress

is negative, $\hat{\bar{\sigma}}_{\max} < 0$ (see *Figure 8*); it must satisfy the condition $0.5 < K_c \leq 1.0$ (the default value is $2/3$);

$\bar{\sigma}_t\left(\tilde{\varepsilon}_t^{pl}\right)$

is the effective tensile cohesion stress; and

$\bar{\sigma}_c\left(\tilde{\varepsilon}_c^{pl}\right)$

is the effective compressive cohesion stress.

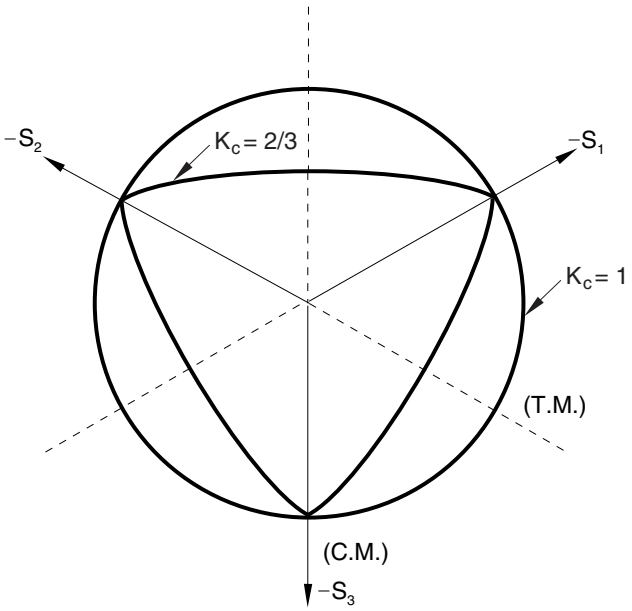


Figure 8: Yield surfaces in the deviatoric plane, corresponding to different values of K_c.

Typical yield surfaces are shown in *Figure 8* on the deviatoric plane and in *Figure 9* for plane stress conditions.

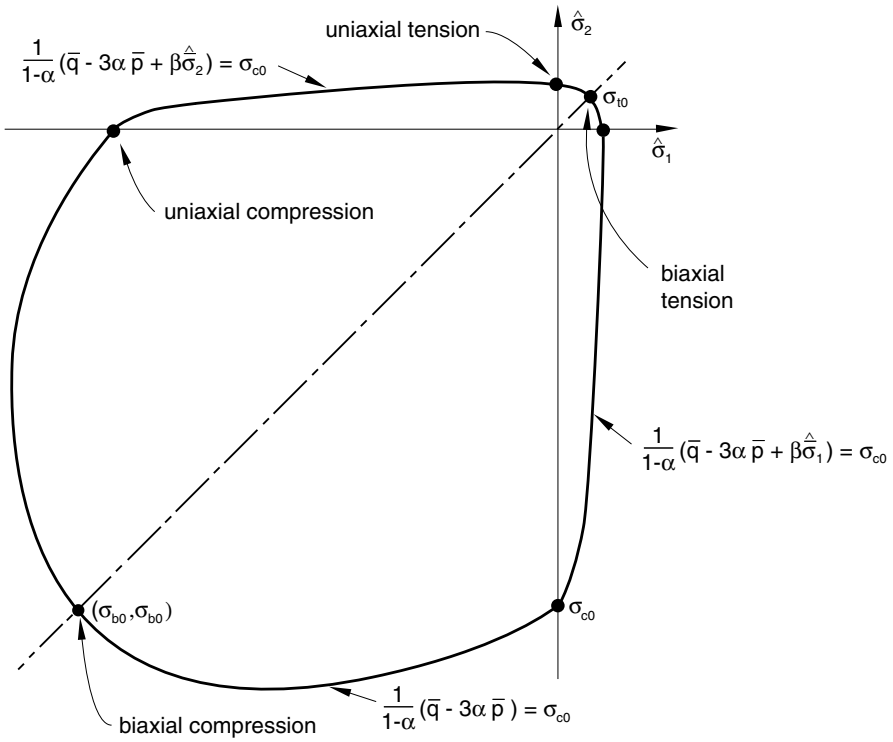


Figure 9: Yield surface in plane stress.

Nonassociated Flow

Because plastic flow is nonassociated, the use of concrete damaged plasticity results in a nonsymmetric material stiffness matrix. Therefore, to obtain an acceptable rate of convergence in Abaqus/Standard, the unsymmetric matrix storage and solution scheme should be used. Abaqus/Standard will automatically activate the unsymmetric solution scheme if concrete damaged plasticity is used in the analysis. If desired, you can turn off the unsymmetric solution scheme for a particular step (see *Defining an Analysis*).

Viscoplastic Regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. A common technique to overcome some of these convergence difficulties is the use of a viscoplastic regularization of the constitutive equations, which causes the consistent tangent stiffness of the softening material to become positive for sufficiently small time increments.

The concrete damaged plasticity model can be regularized in Abaqus/Standard using viscoplasticity by permitting stresses to be outside of the yield surface. We use a generalization of the Duvaut-Lions regularization, according

to which the viscoplastic strain rate tensor, $\dot{\boldsymbol{\varepsilon}}_v^{pl}$, is defined as

$$\dot{\boldsymbol{\varepsilon}}_v^{pl} = \frac{1}{\mu}\left(\boldsymbol{\varepsilon}^{pl} - \boldsymbol{\varepsilon}_v^{pl}\right).$$

Here μ is the viscosity parameter representing the relaxation time of the viscoplastic system, and $\boldsymbol{\varepsilon}^{pl}$ is the plastic strain evaluated in the inviscid backbone model.

Similarly, a viscous stiffness degradation variable, $\boldsymbol{d}_v$, for the viscoplastic system is defined as

$$\dot{d}_v = \frac{1}{\mu}\left(d - d_v\right),$$

where d is the degradation variable evaluated in the inviscid backbone model. The stress-strain relation of the viscoplastic model is given as

$$\boldsymbol{\sigma} = \left(1 - d_v\right)\mathbf{D}_0^{el} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_v^{pl}\right).$$

Using the viscoplastic regularization with a small value for the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results. The basic idea is that the solution of the viscoplastic system relaxes to that of the inviscid case as $t/\mu \to \infty$, where t represents time. You can specify the value of the viscosity parameter as part of the concrete damaged plasticity material behavior definition. If the viscosity parameter is different from zero, output results of the plastic strain and stiffness degradation refer to the viscoplastic values, $\boldsymbol{\varepsilon}_v^{pl}$ and d_v. In Abaqus/Standard the default value of the viscosity parameter is zero, so that no viscoplastic regularization is performed.

Material Damping

The concrete damaged plasticity model can be used in combination with material damping (see *Material Damping*). If stiffness proportional damping is specified, Abaqus calculates the damping stress based on the undamaged elastic stiffness. This may introduce large artificial damping forces on elements undergoing severe damage at high strain rates.

Visualization of "Crack Directions"

Unlike concrete models based on the smeared crack approach, the concrete damaged plasticity model does not have the notion of cracks developing at the material integration point. However, it is possible to introduce the concept of an effective crack direction with the purpose of obtaining a graphical visualization of the cracking patterns in the concrete structure. Different criteria can be adopted within the framework of scalar-damage plasticity for the definition of the direction of cracking. Following Lubliner et. al. (1989), we can assume that cracking initiates at points where the tensile equivalent plastic strain is greater than zero, $\tilde{\varepsilon}_t^{pl} > 0$, and the maximum principal plastic strain is positive. The direction of the vector normal to the crack plane is assumed to be parallel to the direction of the maximum principal plastic strain. This direction can be viewed in the Visualization module of Abaqus/CAE.

Abaqus/CAE Usage: Visualization module:
Result->Field Output: **PE**, **Max. Principal**
Plot->Symbols

Elements

Abaqus offers a variety of elements for use with the concrete damaged plasticity model: truss, beam, shell, plane stress, plane strain, generalized plane strain, axisymmetric, and three-dimensional elements. Thin-walled, open-section beam elements and PIPE elements can also be used with the concrete damaged plasticity model in Abaqus/Standard.

For general shell analysis more than the default number of five integration points through the thickness of the shell should be used; nine thickness integration points are commonly used to model progressive failure of the concrete through the thickness with acceptable accuracy.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables relate specifically to material points in the concrete damaged plasticity model:

DAMAGEC
Compressive damage variable, d_c.

DAMAGET
Tensile damage variable, d_t.

PEEQ
Compressive equivalent plastic strain, $\tilde{\varepsilon}_c^{pl}$.

PEEQT
Tensile equivalent plastic strain, $\tilde{\varepsilon}_t^{pl}$.

SDEG
Stiffness degradation variable, d.

DMENER
Energy dissipated per unit volume by damage.

ELDMD
Total energy dissipated in the element by damage.

ALLDMD
Energy dissipated in the whole (or partial) model by damage. The contribution from ALLDMD is included in the total strain energy ALLIE.

EDMDDEN
Energy dissipated per unit volume in the element by damage.

SENER
The recoverable part of the energy per unit volume.

ELSE
The recoverable part of the energy in the element.

ALLSE
The recoverable part of the energy in the whole (partial) model.

ESEDEN
The recoverable part of the energy per unit volume in the element.

STATUS
Status of element (available only for Abaqus/Explicit). The status of an element is 1.0 if the element is active and 0.0 if it is not.

STATUSMP
Status of each material point in the element (available only for Abaqus/Explicit). The status of a material point is 1.0 if the material point is active and 0.0 if it is not.

References

- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," *Cement and Concrete Research*, vol. 6, pp. 773–782, 1976.
- Lee, J., and G. L. Fenves, "Plastic-Damage Model for Cyclic Loading of Concrete Structures," *Journal of Engineering Mechanics*, vol. 124, no. 8, pp. 892–900, 1998.

- Lubliner, J., J. Oliver, S. Oller, and E. Oñate, "A Plastic-Damage Model for Concrete," *International Journal of Solids and Structures*, vol. 25, pp. 299–329, 1989.

Permanent Set in Rubberlike Materials

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Combining Material Behaviors*
- *Hyperelastic Behavior of Rubberlike Materials*
- *Classical Metal Plasticity*
- **HYPERELASTIC*
- **MULLINS EFFECT*
- **PLASTIC*

Overview

This feature:

- is intended for modeling permanent set observed in filled elastomers and thermoplastics;
- is based on multiplicative split of the deformation gradient;
- is based on the theory of incompressible isotropic hardening plasticity;
- can be used with any isotropic hyperelasticity model;
- can be combined with Mullins effects; and
- cannot be used to model viscoelastic or hysteresis effects or with the steady-state transport procedure.

Material Behavior

The real behavior of filled rubber elastomers under cyclic loading conditions is quite complex as shown in *Figure 1*.

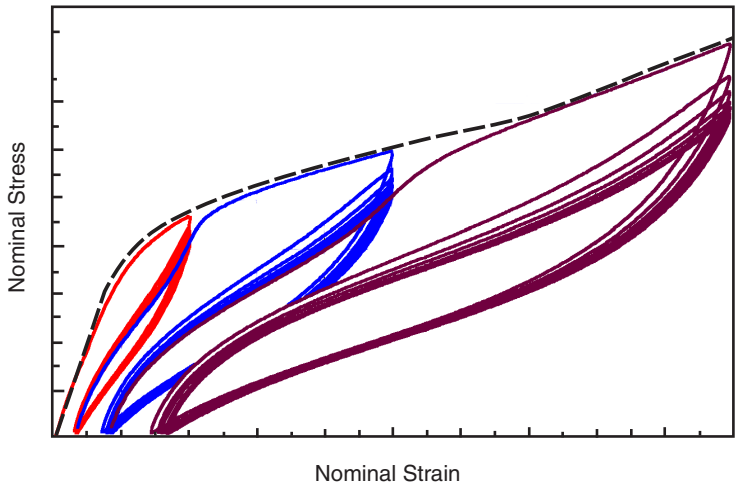


Figure 1: Typical behavior of a filled elastomer.

The observed mechanical behaviors are progressive damage resulting in a reduction of load-carrying capacity with each cycle, stress softening (also known as Mullins effect) upon reloading after the first unloading from a previously attained maximum strain level, hysteretic dissipation of energy, and permanent set. This section is concerned with modeling permanent set; therefore, the idealized representation of permanent set is described below.

Idealized Material Behavior

From *Figure 1* it is clear that the observed permanent set is different for each cycle, but the material has a tendency to stabilize after a number of cycles of loading between zero stress and a given level of strain. For a given load level along the primary loading path shown with the dashed line in *Figure 1*, the idealized representation of permanent set will be a single strain value after unloading has taken place. Since rate and time effects are ignored in this model, idealized loading and unloading take place along the same path, whether Mullins effect is included or not.

The permanent set behavior is captured by isotropic hardening Mises plasticity with an associated flow rule. In the context of finite elastic strains associated with the underlying rubberlike material, plasticity is modeled using a multiplicative split of the deformation gradient into elastic and plastic components:

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p,$$

where $\mathbf{F}^e$ is the elastic part of the deformation gradient (representing the hyperelastic behavior) and $\mathbf{F}^p$ is the plastic part of the deformation gradient (representing the stress-free intermediate configuration).

An example of modeling permanent set along with Mullins effect for a rubberlike material can be found in *Analysis of a solid disc with Mullins effect and permanent set*.

Specifying Permanent Set

The primary hyperelastic behavior can be defined by using any of the hyperelastic material models (see *Hyperelastic Behavior of Rubberlike Materials*). If test data input is used to define the hyperelastic response of the material, the data must be specified with respect to the stress-free intermediate configuration after unloading has taken place.

Permanent set can be defined through an isotropic hardening function in terms of the yield stress and the equivalent plastic strain. In this case the yield stress is the (effective) Kirchoff stress on the primary loading path from which unloading takes place, and the equivalent plastic strain is the corresponding logarithmic permanent set observed in the material. If $\boldsymbol{\sigma}$ is the true (Cauchy) stress, Kirchoff stress is defined as $J\boldsymbol{\sigma}$, where J is the determinant of $\mathbf{F}$.

Depending on what is being modeled, permanent set may be defined as the true permanent set seen in the material after recovery of viscoelastic strains or it may include viscoelastic strains. In either case, an initial yield stress is required, below which there will be no permanent set and the behavior of the material will be fully elastic. In the case of filled rubbers this initial yield stress may correspond to a small nonzero stress; whereas for the family of thermoplastic materials, there may be a more marked value of initial yield stress.

Input File Usage: **PLASTIC*, HARDENING=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Plasticity->Plastic**

Processing Test Data

If you have uniaxial and/or biaxial test data, as shown in *Figure 1*, you can use the **Calibration** tool in Abaqus/CAE to obtain the hyperelasticity, plasticity, and Mullins effect data. For more information about the **Calibration** tool, see *Creating material calibrations*.

Limitations

The model is intended to capture permanent set under multiaxial stress states and mild reverse loading conditions, as illustrated by Govindarajan, Hurtado, and Mars (2007). This model is not intended to capture deformation under complete reverse loading. Any rate effects apply only to the plastic part of the material definition.

Elements

Permanent set can be modeled with all element types that support the use of the hyperelastic material model.

Procedures

Permanent set modeling can be carried out in all procedures that support the use of the hyperelastic material model with the exception of the steady-state transport procedure. In linear perturbation steps in Abaqus/Standard, the current material tangent stiffness corresponding to the elastic part is used to determine the response, while ignoring any plasticity effects.

Output

The standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*) corresponding to other isotropic hardening plasticity models can be obtained for permanent set models.

References

- Govindarajan, S. M., J. A. Hurtado, and W. V. Mars, "Simulation of Mullins Effect in Filled Elastomers Using Multiplicative Decomposition," *European Conference for Constitutive Models for Rubber*, September 2007, Paris, France.
- Simo, J. C., "Algorithms for Static and Dynamic Multiplicative Plasticity that Preserve the Classical Return Mapping Schemes of the Infinitesimal Theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 99 61–112, 1992.
- Weber, G., and L. Anand, "Finite Deformation Constitutive Equations and Time Integration Procedure for Isotropic Hyperelastic-Viscoplastic Solids," *Computer Methods in Applied Mechanics and Engineering*, vol. 79 173–202, 1990.

Plasticity Model for Superelastic Materials

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- **SUPERELASTIC HARDENING*
- **SUPERELASTIC HARDENING MODIFICATIONS*
- *Superelasticity*
- *Defining superelastic hardening*
- *Defining superelastic hardening modifications*

Overview

The plasticity model for superelastic materials provided in Abaqus is intended for modeling the plastic response of the martensitic phase of Nitinol-type materials that undergo solid-solid phase transformations and exhibit superelastic-plastic response.

Material Behavior

The plasticity model for superelastic materials is based on the uniaxial stress-strain response, as shown in *Figure 1*. Such materials (e.g., Nitinol) are in the austenite phase under no loading conditions. Austenite is assumed to follow isotropic linear elasticity. On loading the material, the austenite phase starts transforming into martensite beyond a certain stress. Martensite is assumed to follow an elastoplastic response, with elasticity characterized by the linear elastic model and the plastic behavior represented by the Drucker-Prager model. Martensite exhibits plastic behavior after full transformation.

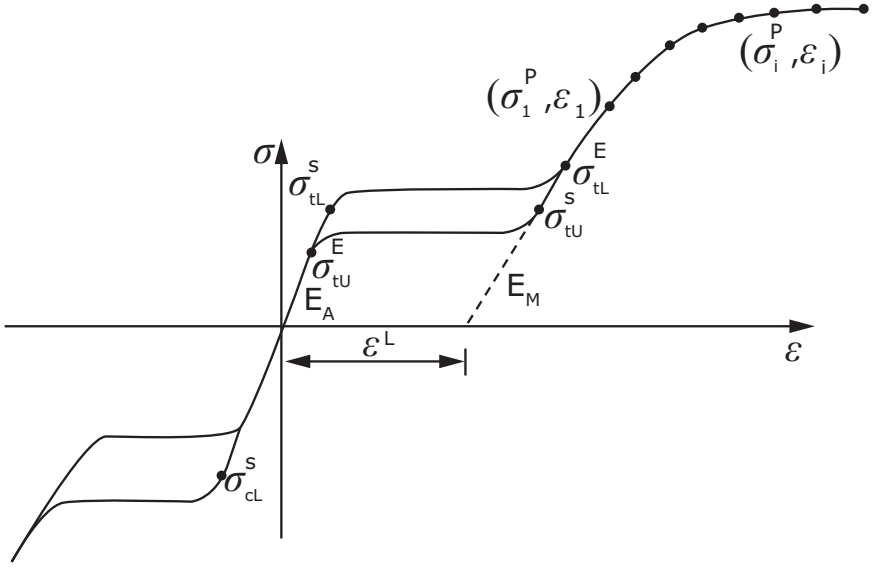


Figure 1: Uniaxial superelastic-plastic response of Nitinol type materials.

In this model the total strain increment, $\Delta\varepsilon$, is assumed to be the sum of the elastic strain increment, $\Delta\varepsilon^{el}$, the increment in transformation strain, $\Delta\varepsilon^{tr}$, and the increment in plastic strain, $\Delta\varepsilon^{pl}$:

$$\Delta\varepsilon = \Delta\varepsilon^{el} + \Delta\varepsilon^{tr} + \Delta\varepsilon^{pl}.$$

The increment in plastic strain is calculated using a nonassociated flow rule:

$$\Delta\varepsilon^{pl} = \Delta\bar{\varepsilon}^{pl}\frac{\partial G^{pl}}{\partial\sigma},$$

where G^{pl} is the plastic flow potential, G^{pl} is equal to the Mises equivalent stress, q,

$$G^{pl} = q.$$

The yield surface, F^{pl}, is assumed to follow the Drucker-Prager formulation with hardening driven by the equivalent plastic strain, $\bar{\varepsilon}^{pl}$:

$$F^{pl} = q - p\tan\beta - F^{Y}\left(\bar{\varepsilon}^{pl}\right),$$

where p is the pressure and β is the friction angle. The initial yield stress is the stress at which the transformation of austenite into martensite ends, σ^{E}_{tL}.

Input File Usage: Use the following option to define the yield stress as a function of total strain.

**SUPERELASTIC HARDENING*

Abaqus/CAE Usage: Use the following option to define the yield stress as a function of total strain.

Property module: material editor: **Mechanical**->**Superelasticity**: **Suboptions**->**Superelastic Hardening**

Superelastic Hardening Modifications

It is observed that the transformation stress levels decrease with an increase in the plastic strain. There are two ways to specify this variation in the transformation plateau with plastic strain in Abaqus. You can either specify the data describing the change in transformation stress levels as a function of the plastic strain or you can use a user subroutine to specify this dependency. *USUPERELASHARDMOD* should be used in Abaqus/Standard, and *VUSUPERELASHARDMOD* should be used in Abaqus/Explicit.

Input File Usage: Use the following option to specify the data to define the dependence of the transformation stress levels on plastic strain:

**SUPERELASTIC HARDENING MODIFICATIONS*

Use the following option to define the dependence of the transformation stress levels on plastic strain in a user subroutine:

**SUPERELASTIC HARDENING MODIFICATIONS*, USER

Abaqus/CAE Usage: Use the following option to specify the data to define the dependence of the transformation stress levels on plastic strain:

Property module: material editor: **Mechanical**->**Superelasticity**: **Suboptions**->**Superelastic Hardening Modifications**

Use the following option to define the dependence of the transformation stress levels on plastic strain in a user subroutine:

Property module: material editor: **Mechanical**->**Superelasticity**: **Suboptions**->**Superelastic Hardening Modifications**: toggle on **User**

Elements

The plasticity model for superelastic materials can be used with any stress/displacement element in Abaqus.

Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials

Products: Abaqus/Explicit

References:

- **PLY FABRIC HARDENING*
- *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*
- *Ply Fabric Criterion*
- *Damage Evolution for the Ply Fabric Model*

Overview

In Abaqus/Explicit the elastic behavior for bidirectional fabric-reinforced composites is assumed to be in the form of bilamina elasticity (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*). This shear plasticity model can be used with the ply fabric damage initiation criterion (see *Ply Fabric Criterion*) to simulate progressive damage and failure for bidirectional fabric-reinforced composites.

Shear Plasticity Behavior

The shear plasticity model of the matrix is formulated in the local directions of the bidirectional fabric-reinforced material and is assumed to be decoupled from the response along the local 1 and 2 fiber directions. The following elastic relation gives the in-plane shear stress in terms of the elastic shear strain, ε_{12}^{el}:

$$\sigma_{12} = 2G_{12}\varepsilon_{12}^{el} = 2G_{12}\left(\varepsilon_{12} - \varepsilon_{12}^{pl}\right),$$

where G_{12} is the shear modulus of the matrix, ε_{12} is the total shear strain, and ε_{12}^{pl} is the plastic shear strain.

The yield surface and flow rule are defined as:

$$F = \left|\sigma_{12}\right| - \overline{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}\right) = 0,$$

$$\dot{\varepsilon}^{pl} = \dot{\bar{\varepsilon}}^{pl}\frac{\partial F}{\partial \sigma_{12}},$$

where $\overline{\sigma}$ is the yield stress, $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain, and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate.

The hardening behavior and strain rate dependency are incorporated into the yield stress $\overline{\sigma}$ assuming this separable form:

$$\overline{\sigma}\left(\bar{\varepsilon}^{pl}, \dot{\bar{\varepsilon}}^{pl}\right) = \sigma^0\left(\bar{\varepsilon}^{pl}\right) R\left(\dot{\bar{\varepsilon}}^{pl}\right),$$

where σ^0 is the static yield stress, and R is a ratio, defined as $R = 1.0$ at $\dot{\bar{\varepsilon}}^{pl} = 0.0$.

The hardening behavior defines the evolution of the yield surface size, σ^0, as a function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$. This evolution can be introduced by specifying σ^0 directly as a function of $\bar{\varepsilon}^{pl}$ in tabular form or by using the Johnson-Cook law:

$$\sigma^0 = \sigma|_0 + C\left(\bar{\varepsilon}^{pl}\right)^n,$$

where $\sigma|_0$ is the yield stress at zero plastic strain and C and n are material parameters. A schematic figure of the hardening curve based on the Johnson-Cook law is shown in *Figure 1*.

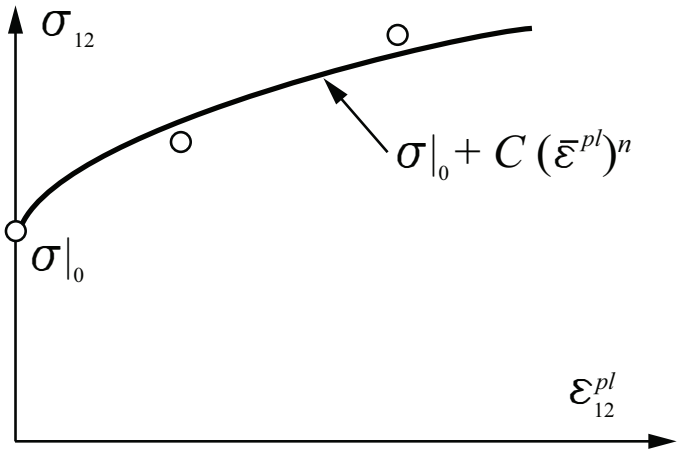


Figure 1: Shear hardening curve of bidirectional fabric-reinforced composite materials.

You can specify rate-dependent effects using the Cowper-Symonds overstress power law form or the Johnson-Cook form. For more information on these methods, see *Rate-Dependent Yield*.

Input File Usage: Use the following option to define Johnson-Cook hardening behavior:

**PLY FABRIC HARDENING*, TYPE=JOHNSON COOK (default)

Use the following option to define the hardening behavior by giving the yield stress as a function of the yield strain:

**PLY FABRIC HARDENING*, TYPE=TABULAR

Use both of the following options to specify rate-dependent effects:

**PLY FABRIC HARDENING (to specify the static yield stress $\sigma^0\left(\bar{\varepsilon}^{pl}\right)$)*

**RATE DEPENDENT (to specify the ratio $R\left(\dot{\bar{\varepsilon}}^{pl}\right)$)*

Elements

The shear plasticity model for bidirectional fabric-reinforced composite materials can be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements in Abaqus/Explicit.

Paperboard Material Model

Products: Abaqus/Standard Abaqus/Explicit

References:

- **PAPERBOARD PLASTICITY*
- **PAPERBOARD HARDENING*
- **PAPERBOARD THICKNESS COMPRESSION ELASTIC*
- **PAPERBOARD THICKNESS COMPRESSION PLASTICITY*
- **PAPERBOARD TRANSVERSE SHEAR PLASTICITY*

Overview

The paperboard plasticity model:

- consists of the combination of constitutive models for the in-plane, thickness, and out-of-plane shear responses, which Abaqus assumes to be decoupled;
- can be used to model the anisotropic in-plane elastic-plastic response; the in-plane response can be used in conjunction with the ductile damage and failure (*About Damage and Failure for Ductile Metals*) that allows for the progressive degradation of the material stiffness and the removal of elements from the mesh;
- considers the out-of-plane behavior to be linear elastic in tension and elastic-plastic in compression;
- can be used to model the out-of-plane shear plastic response; and
- must be used with the orthotropic linear elastic material model (*Linear Elastic Behavior*) for which the in-plane and out-of-plane behaviors are decoupled ($\nu_{13} = 0$ and $\nu_{23} = 0$).

Local Directions

Paperboard materials are made with fiber and water. During the manufacturing process, most of the fibers are oriented in the machine direction (MD), and almost no fibers are oriented in the thickness direction (ZD). These orientations mean that the mechanical response of paperboard material is orthotropic. The two in-plane principal directions are along the machine (MD) and cross (CD) directions, and the third principal direction is along the thickness (ZD) direction, as shown in *Figure 1*.

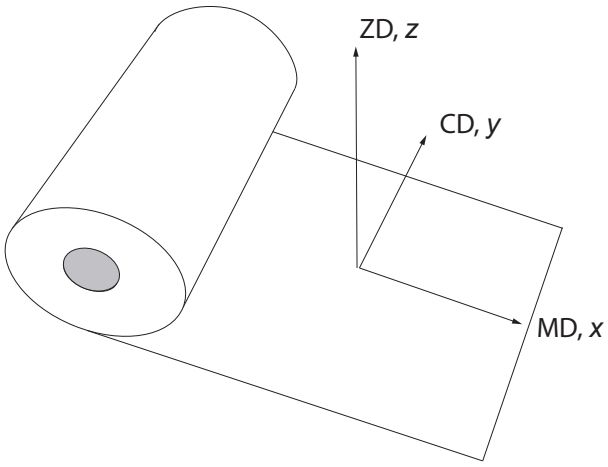


Figure 1: Principal directions of a paperboard material.

In-plane Response

The in-plane stress-strain response of paperboard is highly dependent on the fiber directions (see *Figure 2*). The elastic in-plane response of the paperboard model can be defined in the form

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix},$$

where E_1, E_2, υ_{12}, and G_{12} are the engineering constants.

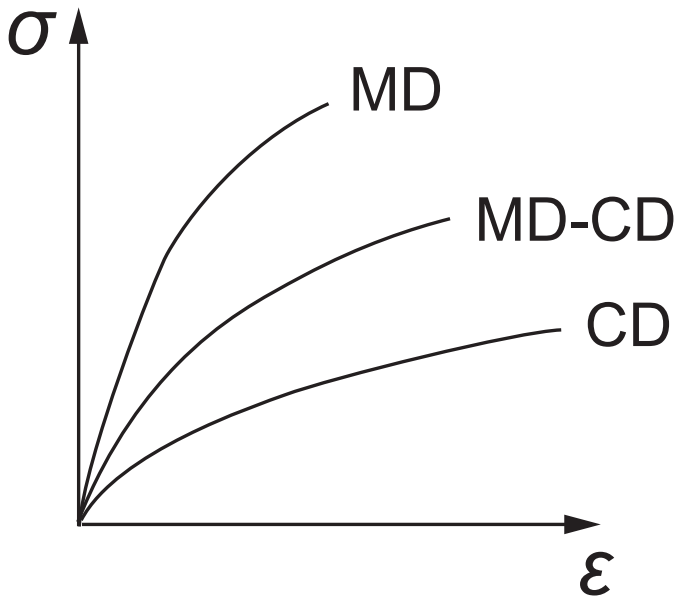


Figure 2: Typical stress-strain response of paper board under uniaxial tension in MD, MD-CD, and CD directions.

Yield Criteria

A yield criterion was proposed by *Xia (2002)*:

$$f\left(\boldsymbol{\sigma},\bar{\varepsilon}^{pl}\right)=\sum_{I=1}^{6}\chi_I\left(\frac{\boldsymbol{\sigma}:\mathbf{N}_I}{\bar{\sigma}^I\left(\bar{\varepsilon}^{pl}\right)}\right)^{2k}-1=0,$$

with yield surface normals (in Voigt notation and same order as stress components, see *Convention Used for Stress and Strain Components*):

$$N_1=\left[\frac{1}{\sqrt{1+v_{1p}^2}}\quad\frac{-v_{1p}}{\sqrt{1+v_{1p}^2}}\quad 0\quad 0\quad 0\quad 0\right]^T,$$

$$N_2=\left[\frac{-v_{2p}}{\sqrt{1+v_{2p}^2}}\quad\frac{1}{\sqrt{1+v_{2p}^2}}\quad 0\quad 0\quad 0\quad 0\right]^T,$$

$$N_3=\left[0\quad 0\quad 0\quad \sqrt{2}\quad 0\quad 0\right]^T,$$

$$N_4=\left[\frac{-1}{\sqrt{1+v_{4p}^2}}\quad\frac{v_{4p}}{\sqrt{1+v_{4p}^2}}\quad 0\quad 0\quad 0\quad 0\right]^T,$$

$$N_5=\left[\frac{v_{5p}}{\sqrt{1+v_{5p}^2}}\quad\frac{-1}{\sqrt{1+v_{5p}^2}}\quad 0\quad 0\quad 0\quad 0\right]^T,$$

$$N_6=-N_3.$$

The yield surface normals are defined by the plastic "Poisson's ratios" fitted from test data as shown in *Figure 3*:

$$v_{1p}=-d\varepsilon_{CD}^{pl+}/d\varepsilon_{MD}^{pl+},$$

$$v_{2p}=-d\varepsilon_{MD}^{pl+}/d\varepsilon_{CD}^{pl+},$$

$$v_{4p}=-d\varepsilon_{CD}^{pl-}/d\varepsilon_{MD}^{pl-},$$

$$v_{5p}=-d\varepsilon_{MD}^{pl-}/d\varepsilon_{CD}^{pl-},$$

where ε_{MD}^{pl+} and ε_{MD}^{pl-} are the plastic strains in the MD direction when tension/compression load is applied and ε_{CD}^{pl+} and ε_{CD}^{pl-} are the plastic strains in the CD direction when tension/compression load is applied. When determining the yield surface normals, Xia's model assumes that only one yield surface is active when the material is under uniaxial tension, uniaxial compression, or pure shear. This assumption is not strictly true but is reasonable for high values of k. Due to this approximation, the plastic Poisson's ratios predicted by the model can differ slightly from the values used to define the yield surface normals.

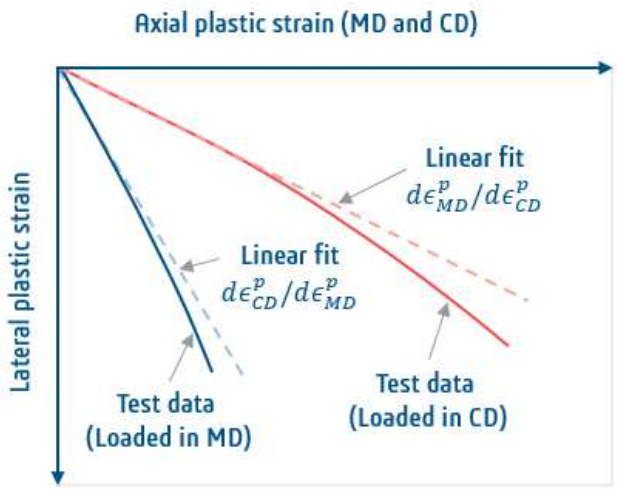


Figure 3: Schematic representation of plastic Poisson's ratios.

Figure 4 shows a typical yield surface that is constructed by six subsurfaces (listed in *Table 1*).

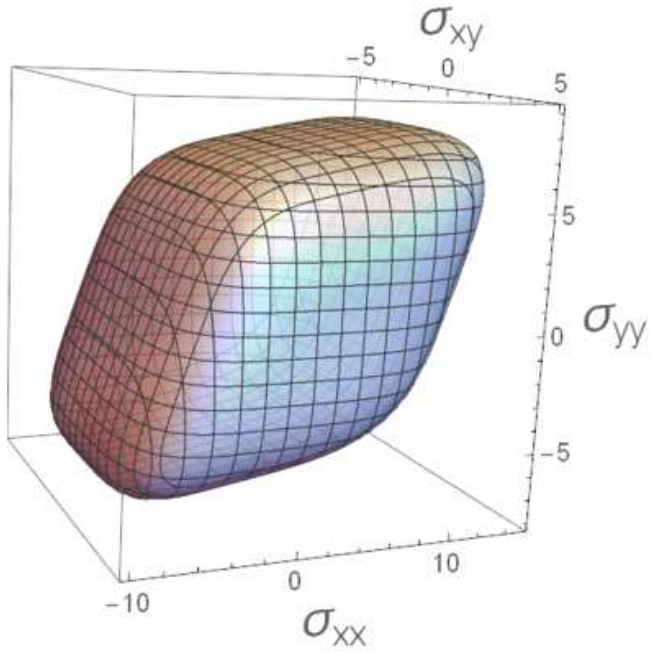


Figure 4: Yield surfaces as proposed by Xia.

Table 1: In-plane Yield Subsurfaces.

Yield Plane	Stress State
1	Tension in MD direction
2	Tension in CD direction
3	Positive shear in MD-CD direction
4	Compression in MD direction
5	Compression in CD direction

Yield Plane	Stress State
6	Negative shear in MD-CD direction

The exponent 2k is a positive even integer, which determines the smoothening between the planes as shown in *Figure 5*—the corners are sharper with larger values of the exponent. $\bar{\sigma}^I$ is the yield stress of the Ith subsurface, which is a function of the equivalent plastic strain $\bar{\varepsilon}^{pl}$. The parameter χ_I is defined as

$$\chi_I = \begin{cases} 1 & \text{if } \boldsymbol{\sigma} : \mathbf{N}_I > 0 \\ 0 & \text{otherwise} \end{cases}.$$

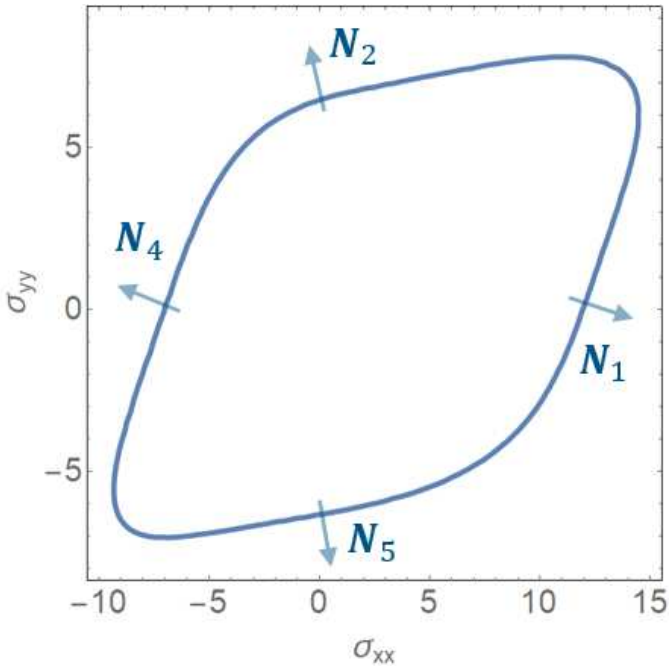


Figure 5: Yield surface proposed by Xia when $\sigma_{xy} = 0$.

Hardening

The yield stress of the Ith subsurface $\bar{\sigma}^I$ can be expressed as

$$\bar{\sigma}^I = \bar{\sigma}_0^I + A_I \tanh\left(B_I \bar{\varepsilon}^{pl}\right) + C_I \bar{\varepsilon}^{pl},$$

where A_I, B_I, and C_I are the hardening parameters. $\bar{\varepsilon}^{pl}$ is the effective plastic strain, which evolves with associated flow rule.

Input File Usage: Use the following option to specify the in-plane plastic "Poisson's ratios":

**PAPERBOARD PLASTICITY*

Use the following option to specify the hardening parameters of the in-plane plasticity:

**PAPERBOARD HARDENING*

ZD Response

Abaqus predicts the normal out-of-plane response of the paperboard material using the model proposed by *Andersson (2006)*. The stress component σ_{33} is considered to be dependent only on the deformation in the thickness direction:

$$\sigma_{33} = \begin{cases} E_3 \varepsilon_{33} & \varepsilon_{33}^{el} \geq 0 \\ \sigma_Z^t \left(1 - \exp\left[\frac{1+r_{orig}}{\mu}\left(\exp\left[\varepsilon_{33}^{pl}\right] - \exp[\varepsilon_{33}]\right)\right]\right) & \varepsilon_{33}^{el} < 0 \end{cases},$$

where E_3 is the elastic modulus in the thickness direction. ε_{33}, ε_{33}^{el}, and ε_{33}^{pl} are the total strain, elastic strain, and plastic strain in the thickness direction, respectively. r_{orig} is the original void to solid ratio. σ_Z^t and μ are parameters for the elastic behavior in ZD.

The yield criterion is defined as:

$$\sigma_{33} = \sigma_s \left(\varepsilon_{33}^{pl}\right),$$

with the hardening function defined as:

$$\sigma_s = A_\sigma + B_\sigma \exp\left(-C_\sigma \varepsilon_{33}^{pl}\right),$$

where A_σ, B_σ, and C_σ are the hardening parameters.

Input File Usage: Use the following option to specify the elastic ZD compression response:

**PAPERBOARD THICKNESS COMPRESSION ELASTIC*

Use the following option to specify the hardening parameters:

**PAPERBOARD THICKNESS COMPRESSION PLASTICITY*

Out-of-Plane Shear Response

The elastic out-of-plane shear response can be defined in the form

$$\begin{Bmatrix} \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \begin{bmatrix} G_{13} & 0 \\ 0 & G_{23} \end{bmatrix} \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix},$$

where G_{13} and G_{23} are the out-of-plane shear moduli.

Abaqus uses a yield criterion proposed by *Andersson (2006)*. The yield function is defined as:

$$f\left(\boldsymbol{\sigma}, \bar{\varepsilon}^{pl}\right) = \sqrt{\sigma_{13}^2 + \sigma_{23}^2} - \tau_s = 0,$$

with the hardening function defined as:

$$\tau_s = A_\tau + B_\tau \tanh\left(C_\tau \bar{\gamma}^{pl}\right),$$

where A_τ, B_τ, and C_τ are the hardening parameters. $\bar{\gamma}^{pl}$ is the effective plastic strain, which evolves with associated flow rule.

Input File Usage: Use the following option to specify the hardening parameters:

**PAPERBOARD TRANSVERSE SHEAR PLASTICITY*

Damage and Failure

The paperboard plasticity model (in-plane response only) can be used with the ductile damage criterion (*Damage Initiation Criteria for Fracture of Metals*) and damage evolution response (*Damage Evolution and Element Removal for Ductile Metals*). The paperboard plasticity model can also be used with the generalized stress- and strain-based damage initiation criteria (*Generalized Damage Initiation Criteria for Anisotropic Materials*).

Input File Usage: Use the following option to specify the ductile damage criterion:

**DAMAGE INITIATION*, CRITERION=DUCTILE

Use the following option to specify the displacement-based damage evolution:

**DAMAGE EVOLUTION*, TYPE=DISPLACEMENT

Use the following option to specify the energy-based damage evolution:

**DAMAGE EVOLUTION*, TYPE=ENERGY

Use the following option to specify the Tsai-Wu stress-based damage initiation criterion:

**DAMAGE INITIATION*, CRITERION=TSAIWU

Use the following option to specify the Tsai-Wu strain-based damage initiation criterion:

**DAMAGE INITIATION*, CRITERION=TSAIWUE

Elements

The paperboard plasticity model can be used with plane stress elements (including shells and membranes) and 3D continuum elements that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the paperboard plasticity models:

PEEQ Equivalent plastic strain for paperboard in-plane plasticity,

$\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}|_0 + \int_0^t \sqrt{\left(\dot{\varepsilon}_{11}^{pl}\right)^2 + \left(\dot{\varepsilon}_{22}^{pl}\right)^2 + 2\left(\dot{\varepsilon}_{12}^{pl}\right)^2}\, dt$, where $\bar{\varepsilon}^{pl}|_0$ is the initial equivalent in-plane plastic strain (zero or user-specified; see *Initial Conditions*).

PEEQTSHR Equivalent plastic strain for paperboard out-of-plane plasticity,

$\bar{\gamma}^{pl} = \bar{\gamma}^{pl}|_0 + \int_0^t \sqrt{\left(\dot{\varepsilon}_{13}^{pl}\right)^2 + \left(\dot{\varepsilon}_{23}^{pl}\right)^2}\, dt$, where $\bar{\gamma}^{pl}|_0$ is the initial equivalent transverse plastic strain (zero or user-specified; see *Initial Conditions*).

References

- Andersson, T., “A Small Deformation Model for the Elasto-Plastic Behavior of Paper and Paperboard,” *Master's thesis*, 2006.
- Xia, Q.S., “Mechanics of Inelastic Deformation and Delamination in Paperboard,” *Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2002*

Progressive Damage and Failure

In this section:

- *About Progressive Damage and Failure*
- *Damage and Failure for Ductile Metals*
- *Damage and Failure for Fiber-Reinforced Composites*
- *Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling*
- *Generalized Damage and Failure for Anisotropic Materials*
- *Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis*

About Progressive Damage and Failure

Overview

Abaqus provides several models to predict progressive damage and failure.

Progressive damage and failure for ductile metals

Abaqus offers a general capability for modeling progressive damage and failure in ductile metals. You can use the functionality with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models (*About Damage and Failure for Ductile Metals*). The capability supports the specification of one or more damage initiation criteria, including ductile, shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), Müschenborn-Sonne forming limit diagram (MSFLD), and Marciniak-Kuczynski (M-K) criteria. After damage initiation, the material stiffness is degraded progressively according to the specified damage evolution response. The progressive damage models allow for a smooth degradation of the material stiffness, which makes them suitable for both quasi-static and dynamic situations, a great advantage over the dynamic failure models (*Dynamic Failure Models*).

The Johnson-Cook and Marciniak-Kuczynski (M-K) damage initiation criteria are not available in Abaqus/Standard.

Progressive damage and failure for fiber-reinforced materials

Abaqus offers a capability to model anisotropic damage in fiber-reinforced materials, including both unidirectional fiber-reinforced composite materials and bidirectional fabric-reinforced composite materials (*About Damage and Failure for Fiber-Reinforced Composites*).

For unidirectional fiber-reinforced composite materials, the response of the undamaged material is assumed to be linearly elastic. The model is intended to predict the behavior of fiber-reinforced materials for which damage can be initiated without a large amount of plastic deformation. The capability supports the Hashin and LaRC05 damage initiation criteria to predict the onset of damage. You can specify both criteria for a material. You can use the Hashin initiation criterion with a damage evolution law that describes the rate of degradation of the material stiffness once the corresponding initiation criterion is reached. The LaRC05 initiation criterion supports damage evolution only when used with enriched elements to model discontinuities (such as cracks) in an extended finite element method (XFEM) analysis. The LaRC05 initiation criterion is available only in Abaqus/Standard.

For bidirectional fabric-reinforced composite materials, the fabric-reinforced ply is modeled as a homogeneous orthotropic elastic material with the potential to sustain progressive stiffness degradation because of fiber/matrix cracking. The material response along the fiber directions is characterized with damaged elasticity. The model differentiates between tensile and compressive fiber failure modes by activating the corresponding damage variable depending on the stress state in the fiber directions. The in-plane shear response of the matrix is dominated by the nonlinear behavior of the matrix, which includes both plasticity and stiffness degradation because of matrix microcracking. The capability uses the ply fabric damage initiation criterion to predict the onset of damage in combination with a damage evolution law that describes the rate of degradation of the material stiffness once the corresponding initiation criterion is reached. The ply fabric damage initiation criterion is available only in Abaqus/Explicit.

Progressive damage and failure for fabric-reinforced materials

Abaqus offers a capability to model anisotropic damage in fabric-reinforced composites. The fabric-reinforced ply is modeled as a homogeneous orthotropic elastic material with the potential to sustain progressive stiffness

degradation because of fiber/matrix cracking and rate-dependent plastic deformation under shear loading. The material response along the fiber directions is characterized with damaged elasticity, and the model differentiates between tensile and compressive fiber failure modes by activating the corresponding damage variable depending on the stress state in the fiber directions. The shear response is dominated by the nonlinear behavior of the matrix, which includes both plasticity and stiffness degradation because of matrix microcracking. The capability supports the ply fabric damage initiation criterion to predict the onset of damage with a damage evolution law that describes the rate of degradation of the material stiffness once the corresponding initiation criterion is reached.

Progressive damage and failure for fiber-reinforced composites using multiscale modeling

Abaqus/Standard offers a capability to model progressive damage and failure for fiber-reinforced composites using multiscale material modeling. This capability can model damage and failure in both unidirectional fiber-reinforced and fabric-reinforced (woven) composites. The capability uses a specific damage initiation criterion to predict the onset of damage at the constituent level. Damage evolution laws are used to describe the rate of degradation of the material stiffness once the corresponding initiation criterion is reached. The responses of the constituents are then homogenized to predict the material response at the composite level.

Progressive damage and failure for ductile materials in low-cycle fatigue analysis

Abaqus/Standard offers a capability to model progressive damage and failure for ductile materials because of stress reversals and the accumulation of inelastic strain in a low-cycle fatigue analysis using the direct cyclic approach (see *Low-Cycle Fatigue Analysis Using the Direct Cyclic Approach*). The damage initiation criterion and damage evolution are characterized by the accumulated inelastic hysteresis energy per stabilized cycle (see *About Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis*). After damage initiation, the elastic material stiffness is degraded progressively according to the specified damage evolution response.

In addition, Abaqus offers a concrete damaged model (*Concrete Damaged Plasticity*), dynamic failure models (*Dynamic Failure Models*), and specialized capabilities for modeling damage and failure in cohesive elements (*Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*) and in connectors (*Connector Damage Behavior*).

This section provides an overview of the progressive damage and failure capability and a brief description of the concepts of damage initiation and evolution. The discussion in this section is limited to damage models for ductile metals and fiber-reinforced materials.

References:

General Framework for Modeling Damage and Failure

Abaqus offers a general framework for material failure modeling that allows the combination of multiple failure mechanisms acting simultaneously on the same material. Material failure refers to the complete loss of load-carrying capacity that results from progressive degradation of the material stiffness. The stiffness degradation process is modeled using damage mechanics.

To help understand the failure modeling capabilities in Abaqus, consider the response of a typical metal specimen during a uniaxial tensile test. *Figure 1*(a), shows the nominal (or engineering) stress-strain response measured directly in a tensile test. The nominal stress reaches a maximum (ultimate tensile strength) value at point $\boldsymbol{x}$, corresponding to the onset of incipient necking. Beyond this point, deformation localizes in a neck region and the stress-strain state is not uniform in the specimen. Therefore, after point $\boldsymbol{x}$, the curve is no longer representative of the actual stress-strain state at any given location of the specimen. At point $\boldsymbol{c}$, there is a marked reduction of load-carrying capacity until rupture occurs, $\boldsymbol{c-d}$. Because of the nonhomogeneous deformation in the specimen, the nominal stress-strain curve cannot be used directly to characterize the material model. Instead, Abaqus requires the material model to be defined in terms of the true stress-strain response.

Figure 1(b) shows the true stress-strain response at the critical location of the specimen where failure occurs. The labels a, b, x, c, and d on this curve indicate the local true stress-strain states of the material corresponding to each stage of the nominal stress-strain curve (*Figure 1*(a)). The material response is initially linear elastic, $a - b$, followed by plastic yielding with strain hardening, $b - c$. The ultimate strength point, x, lies on the strain-hardening portion of the true stress-strain curve. Beyond point c, there is a marked reduction of load-carrying capacity until rupture, $c - d$. Point c in this curve identifies the material state at the onset of damage, which is referred to as the damage initiation criterion. Beyond this point, the true stress-strain response $c - d$ is governed by the evolution of the degradation of the stiffness leading to complete failure. In the context of damage mechanics, $c - d$ can be viewed as the degraded response of the curve $c - d'$ that the material would have followed in the absence of damage.

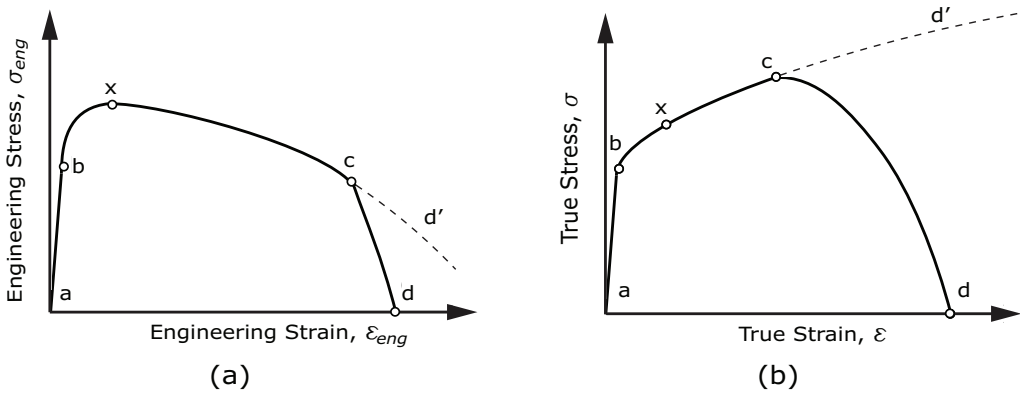


Figure 1: Typical uniaxial stress-strain response of a metal specimen: (a) nominal stress-strain response and (b) true-stress strain response at the critical location of failure.

Thus, in Abaqus, you must specify a failure mechanism in terms of the true stress-strain response, consisting of four distinct parts:

- The definition of the effective (or undamaged) material response (for example, $a - b - c - d'$ in *Figure 1*(b))
- A damage initiation criterion (for example, c in *Figure 1*(b))
- A damage evolution law (for example, $c - d$ in *Figure 1*(b))
- A choice of element deletion where elements can be removed from the calculations once the material stiffness is fully degraded (for example, d in *Figure 1*(b))

These parts are discussed separately for ductile metals (*About Damage and Failure for Ductile Metals*) and fiber-reinforced materials (*About Damage and Failure for Fiber-Reinforced Composites*).

Mesh Dependency

In continuum mechanics the constitutive model is normally expressed in terms of stress-strain relations. When the material exhibits strain-softening behavior, leading to strain localization, this formulation results in a strong mesh dependency of the finite element results in that the energy dissipated decreases on mesh refinement. In Abaqus, all the available damage evolution models use a formulation intended to alleviate the mesh dependency. This is accomplished by introducing a characteristic length into the formulation, which in Abaqus is related to the element size, and expressing the softening part of the constitutive law as a stress-displacement relation. In this case the energy dissipated during the damage process is specified per unit area, not per unit volume. This energy is treated as an additional material parameter, and it is used to compute the displacement at which full material damage occurs. This is consistent with the concept of critical energy release rate as a material parameter

for fracture mechanics. This formulation ensures that the correct amount of energy is dissipated and greatly alleviates the mesh dependency.

Damage and Failure for Ductile Metals

In this section:

- *About Damage and Failure for Ductile Metals*
- *Damage Initiation for Ductile Metals*
- *Damage Evolution and Element Removal for Ductile Metals*

About Damage and Failure for Ductile Metals

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Ductile Metals*
- *Damage Evolution and Element Removal for Ductile Metals*
- **DAMAGE INITIATION*
- **DAMAGE EVOLUTION*
- *Defining damage*

Overview

Abaqus/Standard and Abaqus/Explicit offer general capabilities to predict the onset of damage and model the progressive damage and failure of metals and other ductile materials. In the most general case this requires the specification of the following:

- the undamaged elastic-plastic response of the material (*Classical Metal Plasticity*);
- a damage initiation criterion (*Damage Initiation for Ductile Metals*); and
- a damage evolution response, including a choice of element removal (*Damage Evolution and Element Removal for Ductile Metals*).

Although primarily intended for ductile metals, these capabilities can also be used with the permanent set model for rubberlike materials, the parallel rheological framework, and the paperboard plasticity model.

A summary of the general framework for progressive damage and failure in Abaqus is given in *About Progressive Damage and Failure*. This section provides an overview of the damage initiation criteria and damage evolution law for ductile metals. In addition, Abaqus/Explicit offers dynamic failure models that are suitable for high-strain-rate dynamic problems (*Dynamic Failure Models*).

Damage Initiation Criterion

Abaqus offers a variety of choices of damage initiation criteria for ductile metals, each associated with distinct types of material failure. They can be classified in the following categories:

- Damage initiation criteria for the fracture of metals, including ductile (tabular, Johnson-Cook, and Hosford-Coulomb) and shear.
- Damage initiation criteria for the necking instability of sheet metal. These include forming limit diagrams (FLD, FLSD, and MSFLD) intended to assess the formability of sheet metal and the Marciniak-Kuczynski (M-K) criterion (available only in Abaqus/Explicit) to numerically predict necking instability in sheet metal taking into account the deformation history.

These criteria are discussed in *Damage Initiation for Ductile Metals*. Each damage initiation criterion has an associated output variable to indicate whether the criterion has been met during the analysis. A value of 1.0 or higher indicates that the initiation criterion has been met.

More than one damage initiation criterion can be specified for a given material. If multiple damage initiation criteria are specified for the same material, they are treated independently. Once a particular initiation criterion

is satisfied, the material stiffness is degraded according to the specified damage evolution law for that criterion; in the absence of a damage evolution law, however, the material stiffness is not degraded. A failure mechanism for which no damage evolution response is specified is said to be inactive. Abaqus will evaluate the initiation criterion for an inactive mechanism for output purposes only, but the mechanism will have no effect on the material response.

Input File Usage: Use the following option to define each damage initiation criterion (repeat as needed to define multiple criteria):

**DAMAGE INITIATION*, CRITERION=*criterion 1*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***

Damage Evolution

The damage evolution law describes the rate of degradation of the material stiffness once the corresponding initiation criterion has been reached. For damage in ductile metals Abaqus assumes that the degradation of the stiffness associated with each active failure mechanism can be modeled using a scalar damage variable, d_i ($i \in N_{\text{act}}$), where N_{act} represents the set of active mechanisms. At any given time during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\,\overline{\boldsymbol{\sigma}},$$

where D is the overall damage variable and $\overline{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor computed in the current increment. $\overline{\boldsymbol{\sigma}}$ are the stresses that would exist in the material in the absence of damage. The material has lost its load-carrying capacity when $D = 1$. By default, an element is removed from the mesh on material failure.

The overall damage variable, D, captures the combined effect of all active mechanisms and is computed in terms of the individual damage variables, d_i, according to a user-specified rule.

Abaqus supports different models of damage evolution in ductile metals and provides controls associated with element deletion due to material failure, as described in *Damage Evolution and Element Removal for Ductile Metals*. All of the available models use a formulation intended to alleviate the strong mesh dependency of the results that can arise from strain localization effects during progressive damage.

Input File Usage: Use the following option immediately after the corresponding **DAMAGE INITIATION* option to specify the damage evolution behavior:

**DAMAGE EVOLUTION*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion*: Suboptions->Damage Evolution**

Elements

The failure modeling capability for ductile metals can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

For coupled temperature-displacement elements the thermal properties of the material are not affected by the progressive damage of the material stiffness until the condition for element deletion is reached; at this point the thermal contribution of the element is also removed.

The damage initiation criteria for sheet metal necking instability (FLD, FLSD, MSFLD, and M-K) are available only for elements that include mechanical behavior and use a plane stress formulation (that is, plane stress, shell, continuum shell, and membrane elements).

Damage Initiation for Ductile Metals

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- **DAMAGE INITIATION*
- *Defining damage*
- *Parallel Rheological Framework*
- *Permanent Set in Rubberlike Materials*
- *Output Variables*

Overview

- is intended as a general capability for predicting initiation of damage in metals, including sheet, extrusion, and cast metals as well as other materials;
- can be used in combination with the damage evolution models for ductile metals described in *Damage Evolution and Element Removal for Ductile Metals*;
- allows the specification of more than one damage initiation criterion;
- includes ductile (tabular, Johnson-Cook, and Hosford-Coulomb), shear, forming limit diagram (FLD), forming limit stress diagram (FLSD), and Müschenborn-Sonne forming limit diagram (MSFLD) criteria for damage initiation;
- includes in Abaqus/Explicit the Marciniak-Kuczynski (M-K) and Johnson-Cook criteria for damage initiation;
- can be used in Abaqus/Standard with Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity (ductile, shear, FLD, FLSD, and MSFLD criteria);
- can be used in Abaqus/Explicit with Mises and Johnson-Cook plasticity (ductile, shear, FLD, FLSD, MSFLD, Johnson-Cook, and MK criteria) and with Hill and Drucker-Prager plasticity (ductile, shear, FLD, FLSD, MSFLD, and Johnson-Cook criteria);
- can be used with the permanent set model for rubberlike materials (ductile, shear, and Hosford-Coulomb criteria, and in Abaqus/Explicit the Johnson-Cook criterion);
- can be used with models defined using the parallel rheological framework (ductile, shear, and Hosford-Coulomb criteria, and in Abaqus/Explicit the Johnson-Cook criterion); and
- can be used with paperboard plasticity models (in-plane only).

Damage Initiation Criteria for Fracture of Metals

Two main mechanisms can cause the fracture of a ductile metal: ductile fracture because of the nucleation, growth, and coalescence of voids; and shear fracture because of shear band localization. Based on phenomenological observations, these two mechanisms call for different forms of the criteria for the onset of damage (*Hooputra et al., 2004*). The functional forms provided by Abaqus for these criteria are discussed below. These criteria can be used in combination with the damage evolution models for ductile metals discussed in *Damage Evolution and Element Removal for Ductile Metals* to model fracture of a ductile metal. (See *Progressive failure analysis of thin-wall aluminum extrusion under quasi-static and dynamic loads* for an example.)

Ductile Criterion

The ductile criterion is a phenomenological model for predicting the onset of damage because of nucleation, growth, and coalescence of voids. The model assumes that the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}_{\mathrm{D}}^{pl}$, is a function of stress triaxiality and strain rate:

$$\bar{\varepsilon}_{\mathrm{D}}^{pl}\left(\eta, \dot{\bar{\varepsilon}}^{pl}\right),$$

where $\eta = -p/q$ is the stress triaxiality, p is the pressure stress, q is the Mises equivalent stress, and $\dot{\bar{\varepsilon}}^{pl}$ is the equivalent plastic strain rate. The criterion for damage initiation is met when the following condition is satisfied:

$$\omega_{\mathrm{D}} = \int \frac{\mathrm{d}\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_{\mathrm{D}}^{pl}\left(\eta, \dot{\bar{\varepsilon}}^{pl}\right)} = 1,$$

where ω_{D} is a state variable that increases monotonically with plastic deformation. At each increment during the analysis the incremental increase in ω_{D} is computed as

$$\Delta\omega_{\mathrm{D}} = \frac{\Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}_{\mathrm{D}}^{pl}\left(\eta, \dot{\bar{\varepsilon}}^{pl}\right)} \geq 0.$$

In Abaqus/Standard the ductile criterion can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models and in Abaqus/Explicit with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state.

Input File Usage: Use the following option to specify the equivalent plastic strain at the onset of damage as a tabular function of stress triaxality, strain rate, and, optionally, temperature and predefined field variables:

**DAMAGE INITIATION*, CRITERION=DUCTILE, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->Ductile Damage**

Defining Dependency of Ductile Criterion on Lode Angle

Experimental results for aluminum alloys and other metals (*Bai and Wierzbicki, 2008*) reveal that, in addition to stress triaxiality and strain rate, ductile fracture can also depend on the third invariant of deviatoric stress, which is related to the Lode angle (or deviatoric polar angle). Specific models that account for this dependency include the Modified Mohr-Coulomb (MMC) fracture criterion proposed by *Bai and Wierzbicki (2010)* and the Hosford-Coulomb criterion proposed by *Mohr and Marcadet (2015)*. Abaqus allows the definition of the equivalent plastic strain at the onset of ductile damage, $\bar{\varepsilon}_{\mathrm{D}}^{pl}$, as a function of the Lode angle, Θ, by way of the functional form

$$\bar{\varepsilon}_{\mathrm{D}}^{pl}\left(\eta, \xi(\Theta), \dot{\bar{\varepsilon}}^{pl}\right),$$

where the lode angle term, $\xi(\Theta)$, is defined as

$$\xi(\Theta) = \cos(3\Theta) = \left(\frac{r}{q}\right)^3,$$

q is the Mises equivalent stress, and r is the third invariant of deviatoric stress, $r = \left(\frac{9}{2}\mathbf{S}\cdot\mathbf{S}:\mathbf{S}\right)^{\frac{1}{3}}$. The lode angle term $\xi(\Theta)$ can take values from $\xi = -1$, for stress states on the compressive meridian, to $\xi = 1$, for stress states on the tensile meridian.

Input File Usage: Use the following option to indicate that the equivalent plastic strain at the onset of ductile damage is a function of the Lode angle:

**DAMAGE INITIATION*, CRITERION=DUCTILE, LODE DEPENDENT

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->Ductile Damage**: **Use Lode-dependent data**

Johnson-Cook Criterion

The Johnson-Cook criterion (available only in Abaqus/Explicit) is a special case of the ductile criterion in which the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}_{\mathrm{D}}^{pl}$, is assumed to be of the form

$$\bar{\varepsilon}_{\mathrm{D}}^{pl} = \left[d_1 + d_2 \exp\left(-d_3\eta\right)\right]\left[1 + d_4 \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right]\left(1 + d_5\hat{\theta}\right),$$

where d_1–d_5 are failure parameters and $\dot{\varepsilon}_0$ is the reference strain rate. This expression differs from the original formula published by *Johnson and Cook (1985)* in the sign of the parameter d_3. This difference arises because most materials experience a decrease in $\bar{\varepsilon}_{\mathrm{D}}^{pl}$ with increasing stress triaxiality; therefore, d_3 in the above expression usually takes positive values. $\hat{\theta}$ is the nondimensional temperature defined as

$$\hat{\theta} \equiv \begin{cases} 0 & \text{for} \quad \theta < \theta_{\text{transition}} \\ \left(\theta - \theta_{\text{transition}}\right) / \left(\theta_{\text{melt}} - \theta_{\text{transition}}\right) & \text{for} \quad \theta_{\text{transition}} \le \theta \le \theta_{\text{melt}} \\ 1 & \text{for} \quad \theta > \theta_{\text{melt}} \end{cases},$$

where θ is the current temperature, θ_{melt} is the melting temperature, and $\theta_{\text{transition}}$ is the transition temperature defined as the temperature at or below which there is no temperature dependence on the expression of the damage strain $\bar{\varepsilon}_{\mathrm{D}}^{pl}$. The material parameters must be measured at or below the transition temperature.

The Johnson-Cook criterion can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state. When used with the Johnson-Cook plasticity model, the specified values of the melting and transition temperatures should be consistent with the values specified in the plasticity definition. The Johnson-Cook damage initiation criterion can also be specified together with any other initiation criteria, including the ductile criteria; each initiation criterion is treated independently.

Input File Usage: Use the following option to specify the parameters for the Johnson-Cook initiation criterion:

**DAMAGE INITIATION*, CRITERION=JOHNSON COOK

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->Johnson-Cook Damage**

Hosford-Coulomb Criterion

The Hosford-Coulomb criterion is based on the work of *Mohr and Marcadet (2015)*, *Roth and Mohr (2014)*, and *Roth and Mohr (2016)* and is a special case of the ductile criterion in which the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}_{\mathrm{D}}^{pl}$, is assumed to be of the form

$$\bar{\varepsilon}_D^{pl} = b\left(1 + d \cdot \ln\left(\frac{\dot{\bar{\varepsilon}}^{pl}}{\dot{\varepsilon}_0}\right)\right)\left(1 + c\right)^{\frac{1}{n}}\left(\left\{\tfrac{1}{2}\left(|f_1 - f_2|^a + |f_2 - f_3|^a + |f_1 - f_3|^a\right)\right\}^{\frac{1}{a}} + c\left(2\eta + f_1 + f_3\right)\right)^{-\frac{1}{n}},$$

where $\eta = -p/q$ is the stress triaxiality; a, b, c, d, and n are material parameters; $\dot{\varepsilon}_0$ is the reference strain rate; and

$$\begin{aligned} f_1\left(\bar{\theta}\right) &= \tfrac{2}{3}\cos\left(\tfrac{\pi}{6}\left(1 - \bar{\theta}\right)\right), \\ f_2\left(\bar{\theta}\right) &= \tfrac{2}{3}\cos\left(\tfrac{\pi}{6}\left(3 + \bar{\theta}\right)\right), \\ f_3\left(\bar{\theta}\right) &= -\tfrac{2}{3}\cos\left(\tfrac{\pi}{6}\left(1 + \bar{\theta}\right)\right), \end{aligned}$$

where the parameter $\bar{\theta}$ is defined in terms of the Lode angle Θ as

$$\bar{\theta} = 1 - \tfrac{6}{\pi}\Theta.$$

The Hosford exponent $a \geq 1$ controls the effect of the Lode angle parameter and can reproduce Tresca criteria ($a = 1$ or $a = \infty$) and von Mises criteria ($a = 2$ or $a = 4$). The parameter $b \geq 0$ equals the strain to fracture for uniaxial tension (or equibiaxial tension). The friction coefficient $c \geq 0$ controls the influence of stress triaxiality. The parameter n can be regarded as the hardening exponent in a power law isotropic hardening model; typically $n = 0.1$ for metals.

The Hosford-Coulomb criterion can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models. It can also be specified together with any other initiation criteria; each initiation criterion is treated independently.

Input File Usage: Use the following option to specify the parameters for the Hosford-Coulomb initiation criterion:

**DAMAGE INITIATION*, CRITERION=HC

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->Hosford-Coulomb Damage**

Shear Criterion

The shear criterion is a phenomenological model for predicting the onset of damage because of shear band localization. The model assumes that the equivalent plastic strain at the onset of damage, $\bar{\varepsilon}_{\mathrm{S}}^{pl}$, is a function of the shear stress ratio and strain rate:

$$\bar{\varepsilon}_{\mathrm{S}}^{pl}\left(\theta_s, \dot{\bar{\varepsilon}}^{pl}\right).$$

Here $\theta_s = \left(q + k_s p\right)/\tau_{\max}$ is the shear stress ratio, $\tau_{\max}$ is the maximum shear stress, and k_s is a material parameter. A typical value of k_s for aluminum is k_s = 0.3 (Hooputra et al., 2004). The criterion for damage initiation is met when the following condition is satisfied:

$$\omega_S = \int \frac{\mathrm{d}\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_S\left(\theta_s, \dot{\bar{\varepsilon}}^{pl}\right)} = 1,$$

where ω_S is a state variable that increases monotonically with plastic deformation proportional to the incremental change in equivalent plastic strain. At each increment during the analysis, the incremental increase in ω_S is computed as

$$\Delta\omega_S = \frac{\Delta\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_S\left(\theta_s, \dot{\bar{\varepsilon}}^{pl}\right)} \geq 0.$$

Alternatively, the shear stress ratio, θ_s, can be expressed as a function of triaxiality, η, and the Lode angle, Θ. Rewriting the expression for the shear stress ratio as

$$\theta_s = \frac{q}{\tau_{\max}}(1 - k_s\eta),$$

the maximum shear stress can be expressed as a function of the Mises stress and the Lode angle:

$$\tau_{\max} = \frac{q}{\sqrt{3}}\sin\left(\Theta + \frac{\pi}{3}\right).$$

Substituting the above relation in the expression for the shear stress ratio, we obtain

$$\theta_s = \sqrt{3}\csc\left(\Theta + \frac{\pi}{3}\right)(1 - k_s\eta).$$

This expression shows that the criterion depends on the Lode angle and stress triaxiality, and the parameter, k_s, represents the influence of the stress triaxiality in shear failure.

In Abaqus/Explicit the shear criterion can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager plasticity models, including equation of state. In Abaqus/Standard it can be used with the Mises, Johnson-Cook, Hill, and Drucker-Prager models.

Input File Usage: Use the following option to specify k_s and to specify the equivalent plastic strain at the onset of damage as a tabular function of the shear stress ratio, strain rate, and, optionally, temperature and predefined field variables:

**DAMAGE INITIATION*, CRITERION=SHEAR, KS=k_s, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->Shear Damage**

Nonlinear Accumulation of the Damage Initiation Criterion

The linear form of accumulation of the damage initiation criterion described above might not predict the correct onset of damage in situations when the deformation history is not linear (for example, changing stress triaxiality). It has been shown that you can achieve better predictions if you use a formulation based on a power law rule to characterize the nonlinearity in the accumulation of the damage initiation criterion. This power law formulation is available with the ductile, shear, and Hosford-Coulomb criteria in both Abaqus/Standard and Abaqus/Explicit and with the Johnson-Cook criterion in Abaqus/Explicit. For the ductile and Johnson-Cook criteria, it has the form:

$$\omega_D = \int n\left(\frac{\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_D\left(\eta, \dot{\bar{\varepsilon}}^{pl}\right)}\right)^{n-1} \frac{d\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_D\left(\eta, \dot{\bar{\varepsilon}}^{pl}\right)} = 1,$$

and for the shear criterion it can be expressed as:

$$\omega_S = \int n \left(\frac{\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_S\left(\theta_s, \dot{\bar{\varepsilon}}^{pl}\right)} \right)^{n-1} \frac{d\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_S\left(\theta_s, \dot{\bar{\varepsilon}}^{pl}\right)} = 1,$$

where n is the accumulation power.

Input File Usage: Use the following option to defined the accumulation power:

**DAMAGE INITIATION*, ACCUMULATION POWER=*n*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion*: Accumulation Power:** N

Initial Conditions

Optionally, you can specify the initial work hardened state of the material by providing the initial equivalent plastic strain values (see *Defining Initial Values of State Variables for Plastic Hardening*) and, if residual stresses are also present, the initial stress values (see *Defining Initial Stresses*). Abaqus uses this information to initialize the values of the ductile and shear damage initiation criteria, ω_D and ω_S, assuming constant values of stress triaxiality and shear shear ratio (linear stress path).

Input File Usage: Use the following options to specify that material hardening and residual stresses have occurred before the current analysis:

**INITIAL CONDITIONS*, TYPE=HARDENING
**INITIAL CONDITIONS*, TYPE=STRESS

Abaqus/CAE Usage: Use the following options to specify that material hardening and residual stresses have occurred before the current analysis:

Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** and **Stress** for the **Types for Selected Step**

Damage Initiation Criteria for Sheet Metal Instability

Necking instability plays a determining factor in sheet metal forming processes: the size of the local neck region is typically of the order of the thickness of the sheet, and local necks can rapidly lead to fracture. Localized necking cannot be modeled with traditional shell elements used in sheet metal forming simulations because the size of the neck is of the order of the thickness of the element. Abaqus supports four criteria for predicting the onset of necking instability in sheet metals: forming limit diagram (FLD); forming limit stress diagram (FLSD); Müschenborn-Sonne forming limit diagram (MSFLD); and Marciniak-Kuczynski (M-K) criteria, which is available only in Abaqus/Explicit. These criteria apply only to elements with a plane stress formulation (plane stress, shell, continuum shell, and membrane elements); Abaqus ignores these criteria for other elements. The initiation criteria for necking instability can be used in combination with the damage evolution models discussed in *Damage Evolution and Element Removal for Ductile Metals* to account for the damage induced by necking.

Classical strain-based forming limit diagrams (FLDs) are known to be dependent on the strain path. Changes in the deformation mode (for example, equibiaxial loading followed by uniaxial tensile strain) might result in major modifications in the level of the limit strains. Therefore, the FLD damage initiation criterion should be used with care if the strain paths in the analysis are nonlinear. In practical industrial applications, significant changes in the strain path might be induced by multistep forming operations, complex geometry of the tooling, and interface friction, among other factors. For problems with highly nonlinear strain paths Abaqus offers three additional

damage initiation criteria: the forming limit stress diagram (FLSD) criterion, the Müschenborn-Sonne forming limit diagram (MSFLD) criterion, and in Abaqus/Explicit the Marciniak-Kuczynski (M-K) criterion; these alternatives to the FLD damage initiation criterion are intended to minimize load path dependence.

The characteristics of each criterion available in Abaqus for predicting damage initiation in sheet metals are discussed below.

Forming Limit Diagram (FLD) Criterion

The forming limit diagram (FLD) is a useful concept introduced by *Keeler and Backofen (1964)* to determine the amount of deformation that a material can withstand before the onset of necking instability. The maximum strains that a sheet material can sustain before the onset of necking are referred to as the forming limit strains. A FLD is a plot of the forming limit strains in the space of principal (in-plane) logarithmic strains. In the discussion that follows *major* and *minor* limit strains refer to the maximum and minimum values of the in-plane principal limit strains, respectively. The major limit strain is usually represented on the vertical axis and the minor strain on the horizontal axis, as illustrated in *Figure 1*. The line connecting the states at which deformation becomes unstable is referred to as the forming limit curve (FLC). The FLC gives a sense of the formability of a sheet of material. Strains computed numerically by Abaqus can be compared to an FLC to determine the feasibility of the forming process under analysis.

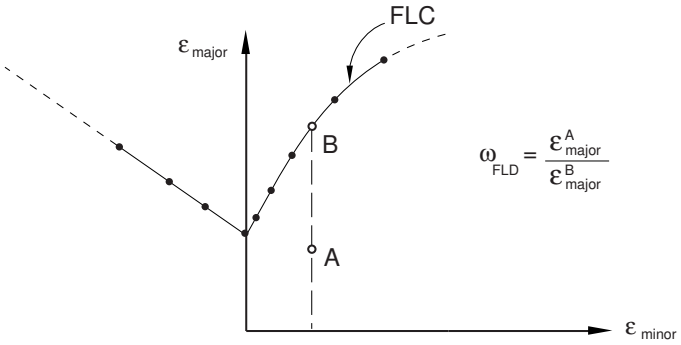


Figure 1: Forming limit diagram (FLD).

The FLD damage initiation criterion requires the specification of the FLC in tabular form by giving the major principal strain at damage initiation as a tabular function of the minor principal strain and, optionally, temperature and predefined field variables, $\varepsilon_{\mathbf{major}}^{\mathbf{FLD}}\left(\varepsilon_{\mathbf{minor}}, \boldsymbol{\theta}, \boldsymbol{f_i}\right)$. The damage initiation criterion for the FLD is given by the condition $\omega_{\mathbf{FLD}} = \mathbf{1}$, where the variable $\omega_{\mathbf{FLD}}$ is a function of the current deformation state and is defined as the ratio of the current major principal strain, $\varepsilon_{\mathbf{major}}$, to the major limit strain on the FLC evaluated at the current values of the minor principal strain, $\varepsilon_{\mathbf{minor}}$; temperature, $\boldsymbol{\theta}$; and predefined field variables, $\boldsymbol{f_i}$:

$$\omega_{\mathbf{FLD}} = \frac{\varepsilon_{\mathbf{major}}}{\varepsilon_{\mathbf{major}}^{\mathbf{FLD}}\left(\varepsilon_{\mathbf{minor}}, \boldsymbol{\theta}, \boldsymbol{f_i}\right)}.$$

For example, for the deformation state given by point A in *Figure 1* the damage initiation criterion is evaluated as $\omega_{\mathbf{FLD}} = \varepsilon_{\mathbf{major}}^{\mathbf{A}} / \varepsilon_{\mathbf{major}}^{\mathbf{B}}$.

If the value of the minor strain lies outside the range of the specified tabular values, Abaqusextrapolates the value of the major limit strain on the FLC by assuming that the slope at the endpoint of the curve remains constant. Extrapolation with respect to temperature and field variables follows the standard conventions: the property is assumed to be constant outside the specified range of temperature and field variables (see *Material Data Definition*).

Experimentally, FLDs are measured under conditions of biaxial stretching of a sheet, without bending effects. Under bending loading, however, most materials can achieve limit strains that are much greater than those on

the FLC. To avoid the prediction of early failure under bending deformation, Abaqus evaluates the FLD criterion using the strains at the midplane through the thickness of the element. For composite shells with several layers the criterion is evaluated at the midplane of each layer for which a FLD curve has been specified, which ensures that only biaxial stretching effects are taken into account. Therefore, the FLD criterion is not suitable for modeling failure under bending loading; other failure models (such as ductile and shear failure) are more appropriate for such loading. Once the FLD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the FLD criterion, they might affect the rate of evolution of damage.

In sheet metal forming analyses, elastic strains are very small compared to the plastic strains and, therefore, can be ignored. Abaqus evaluates the FLD criterion using the plastic strains instead of the total strains. You can define the initial FLD criterion by specifying the initial values of plastic strain. Abaqus initializes the FLD criterion from the specified initial plastic strains assuming the material followed a linear deformation path.

Input File Usage: Use the following option to specify the limit major strain as a tabular function of minor strain:

**DAMAGE INITIATION*, CRITERION=FLD

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->FLD Damage**

Forming Limit Stress Diagram (FLSD) Criterion

When strain-based FLCs are converted into stress-based FLCs, the resulting stress-based curves have been shown to be minimally affected by changes to the strain path (*Stoughton, 2000*); that is, different strain-based FLCs, corresponding to different strain paths, are mapped onto a single stress-based FLC. This property makes forming limit stress diagrams (FLSDs) an attractive alternative to FLDs for the prediction of necking instability under arbitrary loading. However, the apparent independence of the stress-based limit curves on the strain path might simply reflect the small sensitivity of the yield stress to changes in plastic deformation. This topic is still under discussion in the research community.

A FLSD is the stress counterpart of the FLD, with the major and minor principal in-plane stresses corresponding to the onset of necking localization plotted on the vertical and horizontal axes, respectively. In Abaqus, the FLSD damage initiation criterion requires the specification of the major principal in-plane stress at damage initiation as a tabular function of the minor principal in-plane stress and, optionally, temperature and predefined field variables, $\sigma_{\mathrm{major}}^{\mathrm{FLSD}}(\sigma_{\mathrm{minor}}, \theta, f_i)$. The damage initiation criterion for the FLSD is met when the condition $\omega_{\mathrm{FLSD}} = 1$ is satisfied, where the variable ω_{FLSD} is a function of the current stress state and is defined as the ratio of the current major principal stress, σ_{major}, to the major stress on the FLSD evaluated at the current values of minor stress, σ_{minor}; temperature, θ; and predefined field variables, f_i:

$$\omega_{\mathrm{FLSD}} = \frac{\sigma_{\mathrm{major}}}{\sigma_{\mathrm{major}}^{\mathrm{FLSD}}(\sigma_{\mathrm{minor}}, \theta, f_i)}.$$

If the value of the minor stress lies outside the range of specified tabular values, Abaqus extrapolates the value of the major limit stress assuming that the slope at the endpoints of the curve remains constant. Extrapolation with respect to temperature and field variables follows the standard conventions: the property is assumed to be constant outside the specified range of temperature and field variables (see *Material Data Definition*).

For reasons similar to those discussed earlier for the FLD criterion, Abaqus evaluates the FLSD criterion using the stresses averaged through the thickness of the element (or the layer, in the case of composite shells with several layers), ignoring bending effects. Therefore, the FLSD criterion cannot be used to model failure under bending loading; other failure models (such as ductile and shear failure) are more suitable for such loading. Once

the FLSD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the FLSD criterion, they might affect the rate of evolution of damage.

Input File Usage: Use the following option to specify the limit major stress as a tabular function of minor stress:

**DAMAGE INITIATION*, CRITERION=FLSD

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->FLSD Damage**

Marciniak-Kuczynski (M-K) Criterion

Another approach available in Abaqus/Explicit for accurately predicting the forming limits for arbitrary loading paths is based on the localization analysis proposed by Marciniak and Kuczynski (1967). The approach can be used with the Mises and Johnson-Cook plasticity models, including kinematic hardening. In M-K analysis, virtual thickness imperfections are introduced as grooves simulating preexisting defects in an otherwise uniform sheet material. The deformation field is computed inside each groove because of the applied loading outside the groove. Necking is considered to occur when the ratio of the deformation in the groove relative to the nominal deformation (outside the groove) is greater than a critical value.

Figure 2 shows schematically the geometry of the groove considered for M-K analysis. In the figure, *a* denotes the nominal region in the shell element outside the imperfection, and *b* denotes the weak groove region. The initial thickness of the imperfection relative to the nominal thickness is given by the ratio $f_0 = t_0^b/t_0^a$, with the subscript 0 denoting quantities in the initial, strain-free state. The groove is oriented at a zero angle with respect to the 1-direction of the local material orientation.

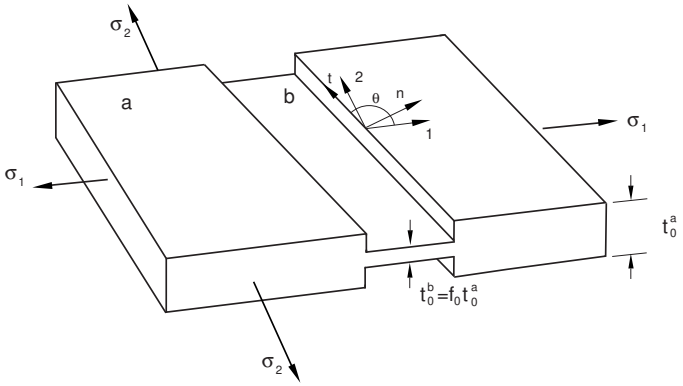


Figure 2: Imperfection model for the M-K analysis.

Abaqus/Explicit allows the specification of an anisotropic distribution of thickness imperfections as a function of angle with respect to the local material orientation, $f_0\,(\theta)$. Abaqus/Explicit first solves for the stress-strain field in the nominal area ignoring the presence of imperfections; then it considers the effect of each groove alone. The deformation field inside each groove is computed by enforcing the strain compatibility condition

$$\varepsilon_{tt}^b = \varepsilon_{tt}^a$$

and the force equilibrium equations

$$F_{nn}^b = F_{nn}^a \quad \text{and}$$
$$F_{nt}^b = F_{nt}^a.$$

The subscripts n and t refer to the directions normal and tangential to the groove. In the above equilibrium equations F_{nn} and F_{nt} are forces per unit width in the t-direction.

The onset of necking instability is assumed to occur when the ratio of the rate of deformation inside a groove relative to the rate of deformation if no groove were present is greater than a critical value. In addition, it might not be possible to find a solution that satisfies equilibrium and compatibility conditions once localization starts at a particular groove; consequently, failure to find a converged solution is also an indicator of the onset of localized necking. For the evaluation of the damage initiation criterion, Abaqus/Explicit uses the following measures of deformation severity:

$$f_{eq} = \frac{\Delta \bar{\varepsilon}^{pl}_{b}}{\Delta \bar{\varepsilon}^{pl}_{a}},$$
$$f_{nn} = \frac{\Delta \varepsilon^{b}_{nn}}{\Delta \varepsilon^{a}_{nn}},$$
$$f_{nt} = \frac{\Delta \varepsilon^{b}_{nt}}{\Delta \varepsilon^{a}_{nt}}.$$

These deformation severity factors are evaluated on each of the specified groove directions and compared with the critical values. (The evaluation is performed only if the incremental deformation is primarily plastic; the M-K criterion does not predict damage initiation if the deformation increment is elastic.) The most unfavorable groove direction is used for the evaluation of the damage initiation criterion, which is given as

$$\omega_{\mathrm{MK}} = \max\left(\frac{f_{eq}}{f^{\mathrm{crit}}_{eq}}, \frac{f_{nn}}{f^{\mathrm{crit}}_{nn}}, \frac{f_{nt}}{f^{\mathrm{crit}}_{nt}}\right),$$

where f^{crit}_{eq}, f^{crit}_{nn}, and f^{crit}_{nt} are the critical values of the deformation severity indices. Damage initiation occurs when $\omega_{\mathrm{MK}} = 1$ or when a converged solution to the equilibrium and compatibility equations cannot be found.

By default, Abaqus/Explicit assumes $f^{\mathrm{crit}}_{eq} = f^{\mathrm{crit}}_{nn} = f^{\mathrm{crit}}_{nt} = 10$; you can specify different values. If one of these parameters is set equal to zero, its corresponding deformation severity factor is not included in the evaluation of the damage initiation criterion. If all of these parameters are set equal to zero, the M-K criterion is based solely on nonconvergence of the equilibrium and compatibility equations.

You must specify the fraction, f_0, equal to the initial thickness at the virtual imperfection divided by the nominal thickness (see *Figure 2*), as well as the number of imperfections to be used for the evaluation of the M-K damage initiation criterion. It is assumed that these directions are equally spaced angularly. By default, Abaqus/Explicit uses four imperfections located at 0°, 45°, 90°, and 135° with respect to the local 1-direction of the material.

The initial imperfection size can be defined as a tabular function of angular direction, $f_0(\theta)$; this allows the modeling of an anisotropic distribution of flaws in the material. Abaqus/Explicit uses this table to evaluate the thickness of each of the imperfections that is used for the evaluation of the M-K analysis method. In addition, the initial imperfection size can also be a function of initial temperature and field variables; this allows defining a nonuniform spatial distribution of imperfections. Abaqus/Explicit computes the initial imperfection size based on the values of temperature and field variables at the beginning of the analysis. The initial size of the imperfection remains a constant property during the rest of the analysis.

A general recommendation is to choose the value of f_0 such that the forming limit predicted numerically for uniaxial strain loading conditions ($\varepsilon_{\mathrm{minor}} = 0$) matches the experimental result.

The virtual grooves are introduced to evaluate the onset of necking instability; they do not influence the results in the underlying element. Once the criterion for necking instability is met, the material properties in the element are degraded according to the specified damage evolution law.

Input File Usage: Use the following option to specify the initial imperfection thickness relative to the nominal thickness as a tabular function of the angle with respect to the 1-direction of the local material orientation and, optionally, initial temperature and field variables:

**DAMAGE INITIATION*, CRITERION=MK, DEPENDENCIES=*n*

Use the following option to specify critical deformation severity factors:

**DAMAGE INITIATION*, CRITERION=MK, FEQ=f_{eq}^{crit}, FNN=f_{nn}^{crit}, FNT=f_{nt}^{crit}

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->M-K Damage**

Performance Considerations for the M-K Criterion

There can be a substantial increase in the overall computational cost when the M-K criterion is used. For example, the cost of processing a shell element with three section points through the thickness and four imperfections, which is the default for the M-K criterion, increases by approximately a factor of two compared to the cost without the M-K criterion. You can mitigate the cost of evaluating this damage initiation criterion by reducing the number of flaw directions considered or by increasing the number of increments between M-K computations, as explained below. Of course, the effect on the overall analysis cost depends on the fraction of the elements in the model that use this damage initiation criterion. The computational cost per element with the M-K criterion increases by approximately a factor of

$$1 + 0.25\frac{n_{\text{imp}}}{N_{\text{incr}}},$$

where n_{imp} is the number of imperfections specified for the evaluation of the M-K criterion and N_{incr} is the frequency, in number of increments, at which the M-K computations are performed. The coefficient of 0.25 in the above formula gives a reasonable estimate of the cost increase in most cases, but the actual cost increase might vary from this estimate. By default, Abaqus/Explicit performs the M-K computations on each imperfection at each time increment, $N_{\text{incr}} = 1$. Care must be taken to ensure that the M-K computations are performed frequently enough to ensure the accurate integration of the deformation field on each imperfection.

Input File Usage: Use the following option to specify the number of imperfections and frequency of the M-K analysis:

**DAMAGE INITIATION*, CRITERION=MK,
NUMBER IMPERFECTIONS=n_{imp}, FREQUENCY= N_{incr}

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->M-K Damage**: **Number of imperfections** and **Frequency**

Müschenborn-Sonne Forming Limit Diagram (MSFLD) Criterion

Müschenborn and Sonne (1975) proposed a method to predict the influence of the deformation path on the forming limits of sheet metals on the basis of the equivalent plastic strain, by assuming that the forming limit curve represents the sum of the highest attainable equivalent plastic strains. Abaqus makes use of a generalization of this idea to establish a criterion of necking instability of sheet metals for arbitrary deformation paths. The approach requires transforming the original forming limit curve (without predeformation effects) from the space

of major versus minor strains to the space of equivalent plastic strain, $\bar{\varepsilon}^{pl}$, versus ratio of principal strain rates, $\alpha = \dot{\varepsilon}_{\text{minor}}/\dot{\varepsilon}_{\text{major}}$.

For linear strain paths, assuming plastic incompressibility and neglecting elastic strains:

$$\alpha = \varepsilon_{\text{minor}}/\varepsilon_{\text{major}} \quad \text{and}$$
$$\bar{\varepsilon}^{pl} = \varepsilon_{\text{major}}\sqrt{\frac{4}{3}\left(1+\alpha+\alpha^2\right)}.$$

As illustrated in *Figure 3*, linear deformation paths in the FLD transform onto vertical paths in the $\bar{\varepsilon}^{pl}$–α diagram (constant value of α).

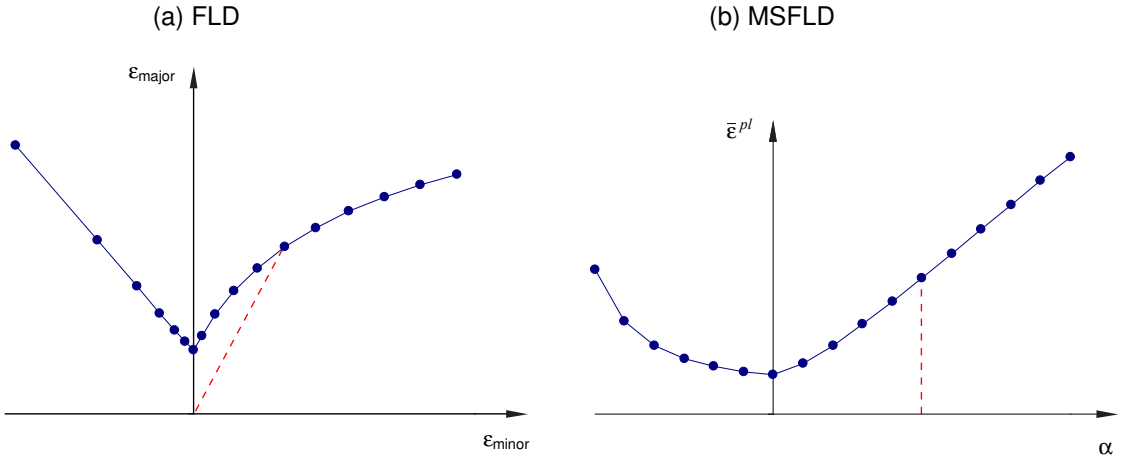


Figure 3: Transformation of the forming limit curve from traditional FLD representation (a) to MSFLD representation (b). Linear deformation paths transform onto vertical paths.

According to the MSFLD criterion, the onset of localized necking occurs when the sequence of deformation states in the $\bar{\varepsilon}^{pl}$–α diagram intersects the forming limit curve, as discussed below. It is emphasized that for linear deformation paths both FLD and MSFLD representations are identical and give rise to the same predictions. For arbitrary loading, however, the MSFLD representation takes into account the effects of the history of deformation through the use of the accumulated equivalent plastic strain.

For the specification of the MSFLD damage initiation criterion in Abaqus, you can directly provide the equivalent plastic strain at damage initiation as a tabular function of α and, optionally, equivalent plastic strain rate, temperature, and predefined field variables, $\bar{\varepsilon}^{pl}_{\text{MSFLD}}\left(\alpha, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)$. Alternatively, you can specify the curve in the traditional FLD format (in the space of major and minor strains) by providing a tabular function of the form $\varepsilon_{\text{major}}\left(\varepsilon_{\text{minor}}, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)$. In this case, Abaqus automatically transforms the data into the $\bar{\varepsilon}^{pl}$–α format.

Let ω_{MSFLD} represent the ratio of the current equivalent plastic strain, $\bar{\varepsilon}^{pl}$, to the equivalent plastic strain on the limit curve evaluated at the current values of α; strain rate, $\dot{\bar{\varepsilon}}^{pl}$; temperature, θ; and predefined field variables, f_i:

$$\omega_{\text{MSFLD}} = \frac{\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_{\text{MSFLD}}\left(\alpha, \dot{\bar{\varepsilon}}^{pl}, \theta, f_i\right)}.$$

The MSFLD criterion for necking instability is met when the condition $\omega_{\text{MSFLD}} = 1$ is satisfied. Necking instability also occurs if the sequence of deformation states in the $\bar{\varepsilon}^{pl}$–α diagram intersects the limit curve due

to a sudden change in the straining direction. This situation is illustrated in *Figure 4*. As α changes from α_t to $\alpha_{t+\Delta t}$, the line connecting the corresponding points in the $\bar{\varepsilon}^{pl}$–α diagram intersects with the forming limit curve.

When this situation occurs, the MSFLD criterion is reached despite the fact that $\bar{\varepsilon}^{pl}_{t+\Delta t} < \bar{\varepsilon}^{pl}_{\text{MSFLD}}\left(\alpha_{t+\Delta t}\right)$. For output purposes Abaqus sets the value of ω_{MSFLD} equal to one to indicate that the criterion has been met.

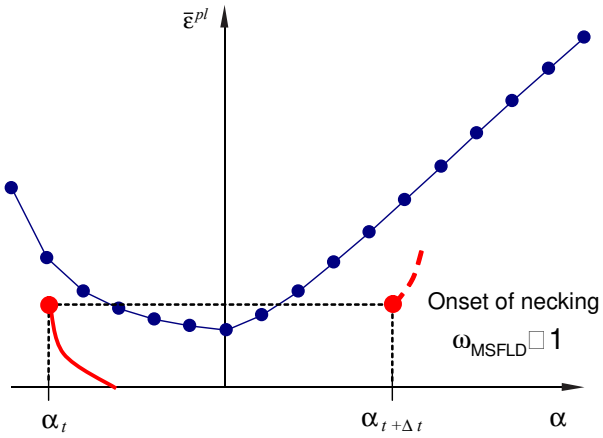


Figure 4: Illustration of how a sudden change in the straining direction, from α_t to $\alpha_{t+\Delta t}$, can produce a horizontal intersection with the limit curve and lead to onset of necking.

The equivalent plastic strain $\bar{\varepsilon}^{pl}$ used for the evaluation of the MSFLD criterion in Abaqus is accumulated only over increments that result in an increase of the element area. Strain increments associated with a reduction of the element area cannot cause necking and do not contribute toward the evaluation of the MSFLD criterion.

If the value of lies outside the range of specified tabular values, Abaqus extrapolates the value of equivalent plastic strain for initiation of necking assuming that the slope at the endpoints of the curve remains constant. Extrapolation with respect to strain rate, temperature, and field variables follows the standard conventions: the property is assumed to be constant outside the specified range of strain rate, temperature, and field variables (see *Material Data Definition*).

As discussed in *Progressive damage and failure of ductile metals*, predictions of necking instability based on the MSFLD criterion agree remarkably well with predictions based on the Marciniak and Kuczynski criterion, at significantly less computational cost than the Marciniak and Kuczynski criterion. There are some situations, however, in which the MSFLD criterion might overpredict the amount of formability left in the material. This occurs in situations when, sometime during the loading history, the material reaches a state that is very close to the point of necking instability and is subsequently strained in a direction along which it can sustain further deformation. In this case, the MSFLD criterion might predict that the amount of additional formability in the new direction is greater than that predicted with the Marciniak and Kuczynski criterion. However, this situation is often not a concern in practical forming applications where safety factors in the forming limit diagrams are

commonly used to ensure that the material state is sufficiently far away from the point of necking. Refer to *Progressive damage and failure of ductile metals* for a comparative analysis of these two criteria.

For reasons similar to those discussed earlier for the FLD criterion, Abaqus evaluates the MSFLD criterion using the strains at the midplane through the thickness of the element (or the layer, in the case of composite shells with several layers), ignoring bending effects. Therefore, the MSFLD criterion cannot be used to model failure under bending loading; other failure models (such as ductile and shear failure) are more suitable for such loading. Once the MSFLD damage initiation criterion is met, the evolution of damage is driven independently at each material point through the thickness of the element based on the local deformation at that point. Thus, although bending effects do not affect the evaluation of the MSFLD criterion, they might affect the rate of evolution of damage.

Input File Usage: Use the following option to specify the MSFLD damage initiation criterion by providing the limit equivalent plastic strain as a tabular function of α (default):

**DAMAGE INITIATION*, CRITERION=MSFLD, DEFINITION=MSFLD

Use the following option to specify the MSFLD damage initiation criterion by providing the limit major strain as a tabular function of minor strain:

**DAMAGE INITIATION*, CRITERION=MSFLD, DEFINITION=FLD

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->MSFLD Damage**

Numerical Evaluation of the Principal Strain Rates Ratio

The ratio of principal strain rates, $\alpha = \dot{\varepsilon}_{\text{minor}}/\dot{\varepsilon}_{\text{major}}$, can jump in value due to sudden changes in the deformation path. Special care is required during explicit dynamic simulations to avoid nonphysical jumps in α triggered by numerical noise, which might cause a horizontal intersection of the deformation state with the forming limit curve and lead to the premature prediction of necking instability.

To overcome this problem, rather than computing α as a ratio of instantaneous strain rates, Abaqus/Explicit periodically updates α based on accumulated strain increments after small but significant changes in the equivalent plastic strain. The threshold value for the change in equivalent plastic strain triggering an update of α is denoted as $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$, and α is approximated as

$$\alpha = \frac{(\Delta\varepsilon)^*_{\text{minor}}}{(\Delta\varepsilon)^*_{\text{major}}},$$

where $(\Delta\varepsilon)^*_{\text{minor}}$ and $(\Delta\varepsilon)^*_{\text{major}}$ are principal values of the accumulated plastic strain since the previous update of α. The default value of $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$ is 0.002 (0.2%).

In addition, Abaqus/Explicit supports the following filtering method for the computation of α:

$$\alpha_{t+(\Delta t)^*} = (1-\omega)\,\alpha_t + \omega\frac{(\Delta\varepsilon)^*_{\text{minor}}}{(\Delta\varepsilon)^*_{\text{major}}},$$

where $(\Delta t)^*$ represents the accumulated time over the analysis increments required to have an increase in equivalent plastic strain of at least $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$. The factor ω ($0 < \omega \leq 1$) facilitates filtering high-frequency oscillations. This filtering method is usually not required provided that an appropriate value of $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$ is used. You can specify the value of ω directly. The default value is $\omega = 1.0$ (no filtering).

In Abaqus/Standard, α is computed at every analysis increment as $\alpha_{t+\Delta t} = \Delta\varepsilon_{\text{minor}}/\Delta\varepsilon_{\text{major}}$, without using either of the above filtering methods. However, you can still specify values for $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$ and ω; and these values can be imported into any subsequent analysis in Abaqus/Explicit.

Input File Usage: **DAMAGE INITIATION*, CRITERION=MSFLD, PEINC=$\left(\Delta\bar{\varepsilon}^{pl}\right)^*$, OMEGA=$\omega$

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->MSFLD Damage**: **Omega:** ω, **Peinc:** $\left(\Delta\bar{\varepsilon}^{pl}\right)^*$

Initial Conditions

When we need to study the behavior of a material that has been previously subjected to deformations, such as those originated during the manufacturing process, initial equivalent plastic strain values can be provided to specify the initial work hardened state of the material (see *Defining Initial Values of State Variables for Plastic Hardening*).

In addition, when the initial equivalent plastic strain is greater than the minimum value on the forming limit curve, the initial value of α plays an important role in determining whether the MSFLD damage initiation criterion is met during subsequent deformation. It is, therefore, important to specify the initial value of α in these situations. To this end, you can specify initial values of the plastic strain tensor (see *Defining Initial Values of Plastic Strain*). Abaqus uses this information to compute the initial value of α as the ratio of the minor and major principal plastic strains; that is, neglecting the elastic component of deformation and assuming a linear deformation path.

Input File Usage: Use both of the following options to specify that material hardening and plastic strain have occurred before the current analysis:

**INITIAL CONDITIONS*, TYPE=HARDENING
**INITIAL CONDITIONS*, TYPE=PLASTIC STRAIN

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Initial plastic strain conditions are not supported in Abaqus/CAE.

Elements

The damage initiation criteria for ductile metals can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom) except for the pipe elements in Abaqus/Explicit.

The models for sheet metal necking instability (FLD, FLSD, MSFLD, and M-K) are available only with elements that include mechanical behavior and use a plane stress formulation (that is, plane stress, shell, continuum shell, and membrane elements).

Output

In addition to the standard output identifiers available in Abaqus(*Output Variables*), the following variables have special meaning when a damage initiation criterion is specified:

ERPRATIO
Ratio of principal strain rates, α, used for the MSFLD damage initiation criterion.

SHRRATIO
Shear stress ratio, $\theta_s = (q + k_s p)/\tau_{\max}$, used for the evaluation of the shear damage initiation criterion.

DMICRT
All damage initiation criteria components listed below.

DUCTCRT
Ductile damage initiation criterion, ω_D.

JCCRT
Johnson-Cook damage initiation criterion (available only in Abaqus/Explicit).

SHRCRT
Shear damage initiation criterion, ω_S.

FLDCRT
Maximum value of the FLD damage initiation criterion, ω_{FLD}, during the analysis.

FLSDCRT
Maximum value of the FLSD damage initiation criterion, ω_{FSLD}, during the analysis.

MSFLDCRT
Maximum value of the MSFLD damage initiation criterion, ω_{MSFLD}, during the analysis.

MKCRT
Marciniak-Kuczynski damage initiation criterion, ω_{MK} (available only in Abaqus/Explicit).

HCCRT
Hosford-Coulomb damage initiation criterion.

A value of 1 or greater for output variables associated with a damage initiation criterion indicates that the criterion has been met. Abaqus limits the maximum value of the output variable to 1 if a damage evolution law has been prescribed for that criterion (see *Damage Evolution and Element Removal for Ductile Metals*). However, if no damage evolution is specified, the criterion for damage initiation continues to be computed beyond the point of damage initiation; in this case, the output variable can take values greater than 1, indicating by how much the initiation criterion has been exceeded.

References

- Bai, Y., and T. Wierzbicki, "A New Model of Metal Plasticity and Fracture with Pressure and Lode Dependence," *International Journal of Plasticity*, vol. 24, no. 6, pp. 1071–1096, 2008.
- Bai, Y., and T. Wierzbicki, "Application of Extended Mohr-Coulomb Criterion to Ductile Fracture," *International Journal of Fracture*, vol. 161, pp. 1–20, 2010.
- Hooputra, H., H. Gese, H. Dell, and H. Werner, "A Comprehensive Failure Model for Crashworthiness Simulation of Aluminium Extrusions," *International Journal of Crashworthiness*, vol. 9, no. 5, pp. 449–464, 2004.
- Johnson, G. R., and W. H. Cook, "Fracture Characteristics of Three Metals Subjected to Various Strains, Strain rates, Temperatures and Pressures," *Engineering Fracture Mechanics*, vol. 21, no. 1, pp. 31–48, 1985.
- Keeler, S. P., and W. A. Backofen, "Plastic Instability and Fracture in Sheets Stretched over Rigid Punches," *ASM Transactions Quarterly*, vol. 56, pp. 25–48, 1964.

- Marciniak, Z., and K. Kuczynski, "Limit Strains in the Processes of Stretch Forming Sheet Metal," *International Journal of Mechanical Sciences*, vol. 9, pp. 609–620, 1967.
- Mohr, D., and S. J. Marcadet, "Micromechanically-Motivated Phenomenological Hosford-Coulomb Model for Predicting Ductile Fracture Initiation at Low Stress Triaxialities," *International Journal of Solids and Structures*, vol. 67–68, pp. 40–55, 2015.
- Müschenborn, W., and H. Sonne, "Influence of the Strain Path on the Forming Limits of Sheet Metal," *Archiv fur das Eisenhüttenwesen*, vol. 46, no. 9, pp. 597–602, 1975.
- Roth, C. C., and D. Mohr, "Effect of Strain Rate on Ductile Fracture Initiation in Advanced High Strength Steel Sheets: Experiments and Modeling," *International Journal of Plasticity*, vol. 56, pp. 19–44, 2014.
- Roth, C. C., and D. Mohr, "Ductile Fracture Experiments with Locally Proportional Loading Histories," *International Journal of Plasticity*, vol. 79, pp. 328–354, 2016.
- Stoughton, T. B., "A General Forming Limit Criterion for Sheet Metal Forming," *International Journal of Mechanical Sciences*, vol. 42, pp. 1–27, 2000.

Damage Evolution and Element Removal for Ductile Metals

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- **DAMAGE EVOLUTION*
- *Damage evolution*

Overview

The damage evolution capability for ductile metals:

- assumes that damage is characterized by the progressive degradation of the material stiffness, leading to material failure;
- must be used in combination with a damage initiation criterion for ductile metals (*Damage Initiation for Ductile Metals*);
- uses mesh-independent measures (either plastic displacement or physical energy dissipation) to drive the evolution of damage after damage initiation;
- takes into account the combined effect of different damage mechanisms acting simultaneously on the same material and includes options to specify how each mechanism contributes to the overall material degradation; and
- offers options for what occurs upon failure, including the removal of elements from the mesh.

Damage Evolution

Figure 1 illustrates the characteristic true stress-strain behavior of a material undergoing damage. (As discussed in *General Framework for Modeling Damage and Failure*, this curve should not be confused with the nominal stress-strain response measured in a uniaxial test.) In the context of an elastic-plastic material with isotropic hardening, the damage manifests itself in two forms: softening of the yield stress and degradation of the elasticity. The solid curve in the figure represents the damaged stress-strain response, while the dashed curve is the response in the absence of damage. As discussed later, the damaged response depends on the element dimensions such that mesh dependency of the results is minimized.

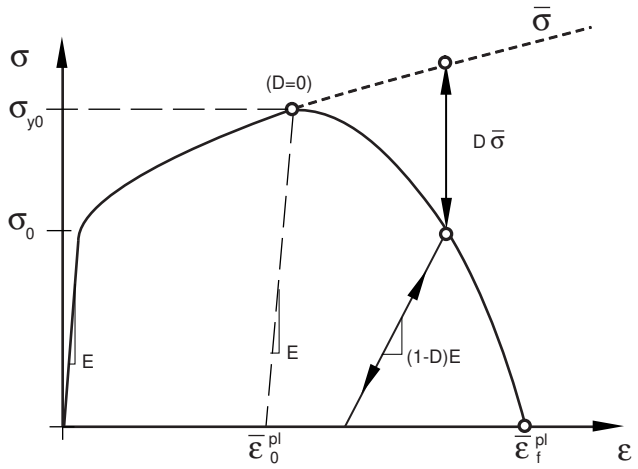


Figure 1: True stress-strain curve with progressive damage degradation.

In the figure σ_{y0} and $\bar{\varepsilon}_0^{pl}$ are the yield stress and equivalent plastic strain at the onset of damage, and $\bar{\varepsilon}_f^{pl}$ is the equivalent plastic strain at failure; that is, when the overall damage variable reaches the value $D = 1$. The overall damage variable, D, captures the combined effect of all active damage mechanisms and is computed in terms of the individual damage variables, d_i, as discussed later in this section (see *Evaluating Overall Damage When Multiple Criteria Are Active*).

The value of the equivalent plastic strain at failure, $\bar{\varepsilon}_f^{pl}$, depends on the characteristic length of the element and cannot be used as a material parameter for the specification of the damage evolution law. Instead, the damage evolution law is specified in terms of equivalent plastic displacement, $\bar{u}^{pl}$, or in terms of fracture energy dissipation, G_f; these concepts are defined next.

Mesh Dependency and Characteristic Length

When material damage occurs, the stress-strain relationship no longer accurately represents the material's behavior. Continuing to use the stress-strain relation introduces a strong mesh dependency based on strain localization, such that the energy dissipated decreases as the mesh is refined. A different approach is required to follow the strain-softening branch of the stress-strain response curve. Hillerborg's (1976) fracture energy proposal is used to reduce mesh dependency by creating a stress-displacement response after damage is initiated. Using brittle fracture concepts, Hillerborg defines the energy required to open a unit area of crack, G_f, as a material parameter. With this approach, the softening response after damage initiation is characterized by a stress-displacement response rather than a stress-strain response.

The implementation of this stress-displacement concept in a finite element model requires the definition of a characteristic length, L, associated with an integration point. The fracture energy is then given as

$$G_f = \int_{\bar{\varepsilon}_0^{pl}}^{\bar{\varepsilon}_f^{pl}} L\sigma_y d\bar{\varepsilon}^{pl} = \int_0^{\bar{u}_f^{pl}} \sigma_y d\bar{u}^{pl}.$$

This expression introduces the definition of the equivalent plastic displacement, $\bar{u}^{pl}$, as the fracture work conjugate of the yield stress after the onset of damage (work per unit area of the crack). Before damage initiation $\dot{\bar{u}}^{pl} = 0$; after damage initiation $\dot{\bar{u}}^{pl} = L\dot{\bar{\varepsilon}}^{pl}$.

The definition of the characteristic length depends on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order

element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the r–z plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic length is used because the direction in which fracture occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which they crack: some mesh sensitivity remains because of this effect, and elements that have aspect ratios close to unity are recommended. Alternatively, this mesh dependency could be reduced by directly specifying the characteristic length as a function of the element topology and material orientation in user subroutine *VUCHARLENGTH* (see *Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit*).

Each damage initiation criterion described in *Damage Initiation for Ductile Metals* might have an associated damage evolution law. The damage evolution law can be specified in terms of equivalent plastic displacement, $\overline{u}^{pl}$, or in terms of fracture energy dissipation, G_f. Both of these options take into account the characteristic length of the element to alleviate mesh dependency of the results.

Evaluating Overall Damage When Multiple Criteria Are Active

The overall damage variable, D, captures the combined effect of all active mechanisms and is computed in terms of individual damage variables, d_i, for each mechanism. You can choose to combine some of the damage variables in a multiplicative sense to form an intermediate variable, d_{mult}, as follows:

$$d_{\text{mult}} = 1 - \prod_{k \in N_{\text{mult}}} (1 - d_k) .$$

Then, the overall damage variable is computed as the maximum of d_{mult} and the remaining damage variables:

$$D = \max\left\{ d_{\text{mult}}, \max_{j \in N_{\text{max}}} (d_j) \right\}.$$

In the above expressions N_{mult} and N_{max} represent the sets of active mechanisms that contribute to the overall damage in a multiplicative and a maximum sense, respectively, with $N_{\text{act}} = N_{\text{mult}} \cup N_{\text{max}}$.

Input File Usage: Use the following option to specify that the damage associated with a particular criterion contributes to the overall damage variable in a maximum sense (default):

**DAMAGE EVOLUTION*, DEGRADATION=MAXIMUM

Use the following option to specify that the damage associated with a particular criterion contributes to the overall damage variable in a multiplicative sense:

**DAMAGE EVOLUTION*, DEGRADATION=MULTIPLICATIVE

Abaqus/CAE Usage: Use the following options to specify that the damage associated with a particular criterion contributes to the overall damage variable in a maximum sense (default) or in a multiplicative sense, respectively:

Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**: **Degradation:Maximum** or **Multiplicative**

Defining Damage Evolution Based on Effective Plastic Displacement

As discussed previously, once the damage initiation criterion has been reached, the effective plastic displacement, $\overline{u}^{pl}$, is defined with the evolution equation

$$\dot{\overline{u}}^{pl} = L\dot{\overline{\varepsilon}}^{pl},$$

where L is the characteristic length of the element.

The evolution of the damage variable with the relative plastic displacement can be specified in tabular, linear,

or exponential form. Instantaneous failure occurs if the plastic displacement at failure, $\overline{u}_f^{pl}$, is specified as 0; however, this choice is not recommended and should be used with care because it causes a sudden drop of the stress at the material point that can lead to dynamic instabilities.

Tabular Form

You can specify the damage variable directly as a tabular function of equivalent plastic displacement, $d = d\left(\overline{u}^{pl}\right)$, as shown in *Figure 2*(a).

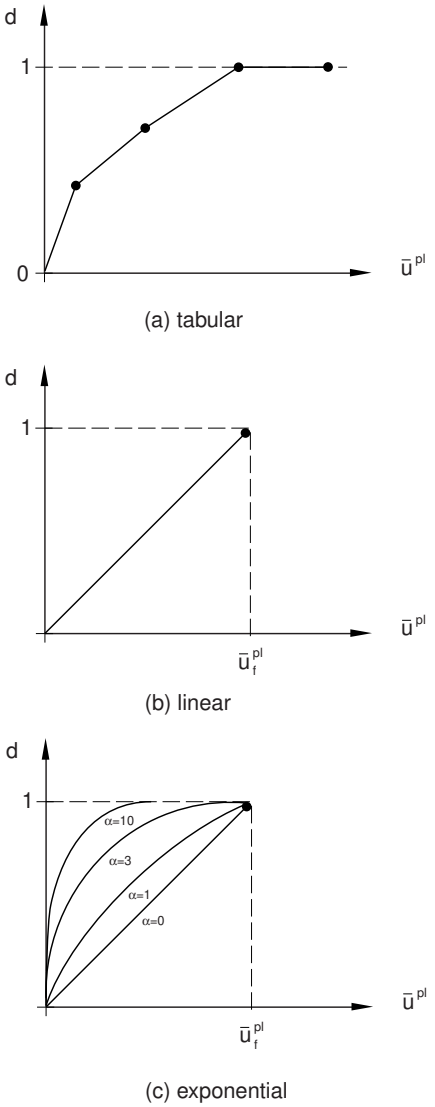


Figure 2: Different definitions of damage evolution based on plastic displacement: (a) tabular, (b) linear, and (c) exponential.

Input File Usage: **DAMAGE EVOLUTION*, TYPE=DISPLACEMENT, SOFTENING=TABULAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion*: Suboptions->Damage Evolution**: **Type:Displacement**: **Softening:Tabular**

Linear Form

Assume a linear evolution of the damage variable with effective plastic displacement, as shown in *Figure 2*(b).

You can specify the effective plastic displacement, $\bar{u}^{pl}_f$, at the point of failure (full degradation). Then, the damage variable increases according to

$$\dot{d} = \frac{L\dot{\bar{\varepsilon}}^{pl}}{\bar{u}_f^{pl}} = \frac{\dot{\bar{u}}^{pl}}{\bar{u}_f^{pl}}.$$

This definition ensures that when the effective plastic displacement reaches the value $\bar{u}^{pl} = \bar{u}_f^{pl}$, the material stiffness will be fully degraded ($d = 1$). The linear damage evolution law defines a truly linear stress-strain softening response only if the effective response of the material is perfectly plastic (constant yield stress) after damage initiation.

Input File Usage: **DAMAGE EVOLUTION*, TYPE=DISPLACEMENT, SOFTENING=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**: **Type:Displacement**: **Softening:Linear**

Exponential Form

Assume an exponential evolution of the damage variable with plastic displacement, as shown in *Figure 2*(c).

You can specify the relative plastic displacement at failure, $\bar{u}_f^{pl}$, and the exponent α. The damage variable is given as

$$d = \frac{1 - e^{-\alpha\left(\bar{u}^{pl}/\bar{u}_f^{pl}\right)}}{1 - e^{-\alpha}}.$$

Input File Usage: **DAMAGE EVOLUTION*, TYPE=DISPLACEMENT, SOFTENING=EXPONENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**: **Type:Displacement**: **Softening:Exponential**

Defining Damage Evolution Based on Energy Dissipated during the Damage Process

You can specify the fracture energy per unit area, G_f, to be dissipated during the damage process directly.

Instantaneous failure occurs if G_f is specified as 0. However, this choice is not recommended and should be used with care because it causes a sudden drop in the stress at the material point that can lead to dynamic instabilities.

The evolution in the damage can be specified in linear or exponential form.

Linear Form

Assume a linear evolution of the damage variable with plastic displacement. You can specify the fracture energy per unit area, G_f. Then, once the damage initiation criterion is met, the damage variable increases according to

$$\dot{d} = \frac{L\dot{\bar{\varepsilon}}^{pl}}{\bar{u}_f^{pl}} = \frac{\dot{\bar{u}}^{pl}}{\bar{u}_f^{pl}},$$

where the equivalent plastic displacement at failure is computed as

$$\bar{u}_f^{pl} = \frac{2G_f}{\sigma_{y0}}$$

and σ_{y0} is the value of the yield stress at the time when the failure criterion is reached. Therefore, the model becomes equivalent to that shown in *Figure 2*(b). The model ensures that the energy dissipated during the damage evolution process is equal to G_f only if the effective response of the material is perfectly plastic (constant yield stress) beyond the onset of damage.

Input File Usage: **DAMAGE EVOLUTION*, TYPE=ENERGY, SOFTENING=LINEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**: **Type:Energy**: **Softening:Linear**

Exponential Form

Assume an exponential evolution of the damage variable given as

$$d = 1 - \exp\left(-\int_0^{\bar{u}^{pl}} \frac{\bar{\sigma}_y d\bar{u}^{pl}}{G_f}\right).$$

The formulation of the model ensures that the energy dissipated during the damage evolution process is equal to G_f, as shown in *Figure 3*(a). In theory, the damage variable reaches a value of 1 only asymptotically at infinite equivalent plastic displacement (*Figure 3*(b)). In practice, Abaqus/Explicit sets d equal to one when the dissipated energy reaches a value of $0.99G_f$.

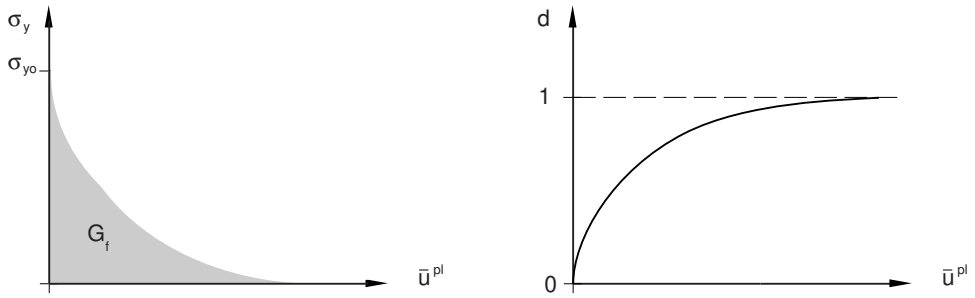


Figure 3: Energy-based damage evolution with exponential law: evolution of (a) yield stress and (b) damage variable.

Input File Usage: **DAMAGE EVOLUTION*, TYPE=ENERGY, SOFTENING=EXPONENTIAL

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Ductile Metals->*criterion***: **Suboptions->Damage Evolution**: **Type:Energy**: **Softening:Exponential**

Maximum Degradation and Choice of Element Removal

You have control over how Abaqus treats elements with severe damage. You can specify an upper bound, D_{max}, to the overall damage variable, D; and you can choose whether to delete an element once maximum degradation is reached. The latter choice also affects which stiffness components are damaged.

Specifying the Value of Maximum Degradation

The default setting of D_{max} depends on whether elements are to be deleted upon material failure. For the default case of element deletion and in all cases for cohesive elements, $D_{max} = 1.0$; otherwise, $D_{max} = 0.99$. Output variable SDEG contains the value of D. No further damage is accumulated at an integration point once D reaches D_{max} (except, of course, any remaining stiffness is lost upon element deletion).

Input File Usage: Use the following option to specify D_{max}:

**SECTION CONTROLS*, MAX DEGRADATION=D_{max}

Removing the Element from the Mesh

A material point is assumed to fail upon reaching maximum degradation. Except for cohesive elements with traction-separation response (see *Defining the Constitutive Response of Cohesive Elements Using a Traction-Separation Description*), Abaqus applies damage to all stiffness components equally at material points during the progressive damage process. At any given time during the analysis, the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\,\overline{\boldsymbol{\sigma}},$$

where $\overline{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor computed in the current increment. By default, an element is deleted (or removed) from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

In a heat transfer analysis the thermal properties of the material are not affected by the damage status of the material stiffness until the element deletion condition is reached; at this point, the thermal contribution of the element is also removed.

Input File Usage: Use the following option to delete the element from the mesh (default):

**SECTION CONTROLS*, ELEMENT DELETION=YES

Keeping the Element in the Computations

Optionally, you can choose not to remove the element from the mesh, except in the case of three-dimensional beam elements. With element deletion turned off, the overall damage variable is enforced to be $D \leq D_{max}$. The default value is $D_{max} = 0.99$ if element deletion is turned off, which ensures that elements remain active in the simulation with a residual stiffness of at least 1% of the original stiffness. The dimensionality of the stress state of the element affects which stiffness components can become damaged, as discussed below.

In a heat transfer analysis the thermal properties of the material are not affected by damage of the material stiffness.

Input File Usage: Use the following option to keep the element in the computation:

**SECTION CONTROLS*, ELEMENT DELETION=NO

Elements with Three-Dimensional Stress States in Abaqus/Explicit

For elements with three-dimensional stress states (including generalized plane strain elements), the shear stiffness will be degraded up to a maximum value, $D_{\max}$, leading to softening of the deviatoric stress components. The bulk stiffness, however, will be degraded only while the material is subjected to negative pressures (that is, hydrostatic tension); there is no bulk degradation under positive pressures. This corresponds to a fluid-like behavior. Therefore, the degraded deviatoric, $\mathbf{S}$, and pressure, p, stresses are computed as

$$\begin{aligned} \mathbf{S} &= (1 - D_{\text{dev}})\,\overline{\mathbf{S}}, \\ p &= (1 - D_{\text{vol}})\,\overline{p}, \end{aligned}$$

where the deviatoric and volumetric damage variables are given as

$$\begin{aligned} D_{\text{dev}} &= D, \\ D_{\text{vol}} &= \begin{cases} D \text{ if } & \overline{p} \le 0, \\ 0 \text{ if } & \overline{p} > 0. \end{cases} \end{aligned}$$

In this case the output variable SDEG contains the value of D_{dev}.

Elements with Three-Dimensional Stress States in Abaqus/Standard

For elements with three-dimensional stress states (including generalized plane strain elements) the stiffness will be degraded uniformly until the maximum degradation, $D_{\max}$, is reached. Output variable SDEG contains the value of D.

Elements with Plane Stress States

For elements with a plane stress formulation (plane stress, shell, continuum shell, and membrane elements) the stiffness will be degraded uniformly until the maximum degradation, $D_{\max}$, is reached. Output variable SDEG contains the value of D.

Elements with One-Dimensional Stress States

For elements with a one-dimensional stress state (i.e., truss elements, rebar, and cohesive elements with gasket behavior) their only stress component will be degraded if it is positive (tension). The material stiffness will remain unaffected under compression loading. The stress is, therefore, given by $\sigma = (1 - D_{\text{uni}})\,\overline{\sigma}$, where the uniaxial damage variable is computed as

$$D_{\text{uni}} = \begin{cases} D \text{ if } & \overline{\sigma} \ge 0, \\ 0 \text{ if } & \overline{\sigma} < 0. \end{cases}$$

In this case $D_{\max}$ determines the maximum allowed degradation in uniaxial tension ($D \le D_{\max}$). Output variable SDEG contains the value of D_{uni}.

Convergence Difficulties in Abaqus/Standard

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. Some techniques are available in Abaqus/Standard to improve convergence for analyses involving these materials.

Viscous Regularization in Abaqus/Standard

You can overcome some of the convergence difficulties associated with softening and stiffness degradation by using the viscous regularization scheme, which causes the tangent stiffness matrix of the softening material to be positive for sufficiently small time increments.

In this regularization scheme a viscous damage variable is defined by the evolution equation:

$$\dot{d}_v = \frac{1}{\eta}\left(d - d_v\right),$$

where η is the viscosity coefficient representing the relaxation time of the viscous system and d is the damage variable evaluated in the inviscid base model. The damaged response of the viscous material is computed using the viscous value of the damage variable. Using viscous regularization with a small value of the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results. The basic idea is that the solution of the viscous system relaxes to that of the inviscid case as $t/\eta \to \infty$, where t represents time.

In Abaqus/Standard you can specify the viscous coefficients as part of a section controls definition. For more information, see *Using Viscous Regularization with Cohesive Elements, Connector Elements, and Elements That Can Be Used with the Damage Evolution Models for Ductile Metals and Fiber-Reinforced Composites Including the Multiscale Model for Fiber-Reinforced Composites in Abaqus/Standard*.

Unsymmetric Equation Solver

In general, if any of the ductile evolution models is used, the material Jacobian matrix will be nonsymmetric. To improve convergence, it is recommended that the unsymmetric equation solver is used in this case.

Using the Damage Models with Rebar

It is possible to use material damage models in elements for which rebar are also defined. The base material contribution to the element stress-carrying capacity diminishes according to the behavior described previously in this section. The rebar contribution to the element stress-carrying capacity is not affected unless damage is also included in the rebar material definition; in that case the rebar contribution to the element stress-carrying capacity will also be degraded after the damage initiation criterion specified for the rebar is met. For the default choice of element deletion, the element is removed from the mesh when at any one integration location all section points in the base material and rebar are fully degraded.

Elements

Damage evolution for ductile metals can be defined for any element that can be used with the damage initiation criteria for ductile metals in Abaqus (*Damage Initiation for Ductile Metals*).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning when damage evolution is specified:

STATUS
Status of element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).

STATUSMP
Status of each material point in the element (1.0 if a material point is active, 0.0 if it is not). Abaqus/Explicit only.

SDEG
Overall scalar stiffness degradation, D.

References

- Hillerborg, A., M. Modeer, and P. E. Petersson, "Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements," *Cement and Concrete Research*, vol. 6, pp. 773–782, 1976.

Damage and Failure for Fiber-Reinforced Composites

In this section:

- *About Damage and Failure for Fiber-Reinforced Composites*
- *Damage Initiation for Fiber-Reinforced Composites*
- *Damage Evolution and Element Removal for Fiber-Reinforced Composites*

About Damage and Failure for Fiber-Reinforced Composites

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Fiber-Reinforced Composites*
- *Damage Evolution and Element Removal for Fiber-Reinforced Composites*
- **DAMAGE INITIATION*
- **DAMAGE EVOLUTION*
- **DAMAGE STABILIZATION*
- *Hashin damage*

Overview

Abaqus/Standard and Abaqus/Explicit offer general capabilities for predicting the onset of damage and for modeling progressive damage and failure in fiber-reinforced composites, which include unidirectional fiber-reinforced composite materials and bidirectional fabric-reinforced composite materials. These capabilities require you to specify:

- the undamaged response of the material, which must be linearly elastic (see *Linear Elastic Behavior*); optionally in-plane shear plasticity is allowed for bidirectional fabric-reinforced composites (see *Damage Initiation for Fiber-Reinforced Composites* and *Damage Evolution and Element Removal for Fiber-Reinforced Composites*);
- a damage initiation criterion (see *About Progressive Damage and Failure* and *Damage Initiation for Fiber-Reinforced Composites*); and
- a damage evolution response (except as noted below), including a choice of element removal (see *About Progressive Damage and Failure* and *Damage Evolution and Element Removal for Fiber-Reinforced Composites*).

Undamaged Response

Unidirectional fiber-reinforced composite materials exhibit elastic-brittle behavior; damage in these materials is initiated without significant plastic deformation. Consequently, plasticity is often neglected when modeling behavior of these materials. You must specify material properties in a user-defined local coordinate system, with the local 1-direction aligned with the fiber direction, as shown in *Figure 1*. You can use any of the available methods to define orthotropic elastic behavior to define the undamaged response.

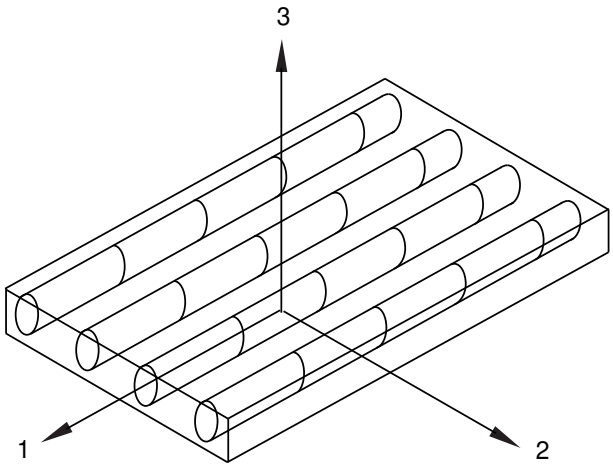


Figure 1: Unidirectional lamina.

For bidirectional fabric-reinforced composite materials, the shear response is dominated by the nonlinear behavior of the matrix, which includes both plasticity and stiffness degradation due to matrix microcracking. The fiber directions are assumed to be orthogonal. You must specify material properties in a user-defined local coordinate system, with the local 1-direction and 2-direction aligned with the fiber directions, as shown in *Figure 2*. The material response along the fiber directions is characterized with damaged elasticity. The model incorporates different initial (undamaged) stiffness in tension and compression and differentiates between tensile and compressive fiber failure modes. Therefore, you must use bilamina elasticity to define the undamaged response (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*).

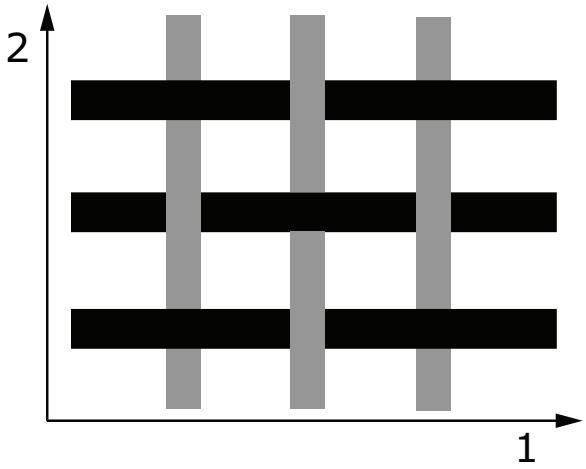


Figure 2: Bidirectional fabric.

Damage Initiation Criteria

Abaqus offers the following damage initiation criteria for unidirectional fiber-reinforced composites:

- Hashin model, based on the work of *Matzenmiller et al. (1995)*, *Hashin and Rotem (1973)*, *Hashin (1980)*, and *Camanho and Davila (2002)*.
- LaRC05 model (available only in Abaqus/Standard), based on the work of *Pinho et al. (2012)*.

Abaqus offers the following damage initiation criterion for bidirectional fabric-reinforced composites:

- Ply fabric model (available only in Abaqus/Explicit), based on the work of *Johnson (2001)* and *Sokolinsky et al. (2011)*.

These models include initiation criteria for various failure mechanisms commonly observed in fiber-reinforced composites, such as fiber fracture in tension, fiber buckling/kinking under compression, and matrix cracking/crushing under tension/compression. These criteria are discussed in *Damage Initiation for Fiber-Reinforced Composites*.

Once a particular damage initiation criterion is satisfied, the material stiffness is degraded according to the specified damage evolution law for that criterion. In the absence of a damage evolution law, the material stiffness is not degraded.

Damage Evolution

The damage evolution law describes the rate of degradation of the material stiffness once the corresponding initiation criterion is reached. At any given time during the analysis, the stress tensor in the material is given by:

$$\sigma = \mathbf{C_d} \varepsilon^{el},$$

where $\mathbf{C_d}$ is the damaged elasticity matrix. Abaqus assumes that the degradation of the stiffness of the fiber and matrix components can be expressed in terms of three scalar damage variables that reflect the current states of fiber, matrix, and shear damages, respectively. The evolution of the elasticity matrix due to damage is discussed in more detail in *Damage Evolution and Element Removal for Fiber-Reinforced Composites*; that section also discusses:

- options for treating severe damage (*Maximum Degradation and Choice of Element Removal*); and
- viscous regularization (*Viscous Regularization*).

The LaRC05 damage initiation model supports damage evolution only when used with enriched elements to model discontinuities (such as cracks) in an extended finite element method (XFEM) analysis. For more information, see *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*.

Elements

The Hashin damage model and ply fabric damage model must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements. The LaRC05 damage model is available with three-dimensional solid, plane stress, shell, solid shell, and membrane elements.

References

- Camanho, P. P., and C. G. Davila, "Mixed-Mode Decohesion Finite Elements for the Simulation of Delamination in Composite Materials," *NASA/TM-2002–211737*, pp. 1–37, 2002.
- Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, vol. 47, pp. 329–334, 1980.
- Hashin, Z., and A. Rotem, "A Fatigue Criterion for Fiber-Reinforced Materials," *Journal of Composite Materials*, vol. 7, pp. 448–464, 1973.
- Johnson, A. F., "Modelling Fabric-Reinforced Composites under Impact Loads," *Composites Part A: Applied Science and Manufacturing*, vol. 32, no. 9, pp. 1197–1206, 2001.
- Lapczyk, I., and J. A. Hurtado, "Progressive Damage Modeling in Fiber-Reinforced Materials," *Composites Part A: Applied Science and Manufacturing*, vol. 38, no. 11, pp. 2333–2341, 2007.
- Matzenmiller, A., J. Lubliner, and R. L. Taylor, "A Constitutive Model for Anisotropic Damage in Fiber-Composites," *Mechanics of Materials*, vol. 20, pp. 125–152, 1995.

- Pinho, S. T., R. Darvizeh, P. Robinson, C. Schuecker, and P. P. Camanho, "Material and Structural Response of Polymer-Matrix Fibre-Reinforced Composites," *Journal of Composite Materials*, vol. 46, no. 19-20, pp. 2313–2341, 2012.
- Sokolinsky, V. S., K. C. Indermuehle, and J. A. Hurtado, "Numerical Simulation of the Crushing Process of a Corrugated Composite Plate," *Composites Part A: Applied Science and Manufacturing*, vol. 42, no. 9, pp. 1119–1126, 2011.

Damage Initiation for Fiber-Reinforced Composites

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- *Damage Evolution and Element Removal for Fiber-Reinforced Composites*
- **DAMAGE INITIATION*
- *Hashin damage*

Overview

The material damage initiation capability for fiber-reinforced materials:

- is intended as a general capability for predicting initiation of damage in fiber-reinforced composite materials;
- is based on the Hashin (*Hashin and Rotem, 1973*, and *Hashin, 1980*) or the LaRC05 (*Pinho et al. (2012)*) models for unidirectional fiber-reinforced composites or on the ply fabric (*Johnson, 2001* and *Sokolinsky et al., 2011*) model for bidirectional fabric-reinforced composites;
- takes into account different failure modes involving the response of the fiber and the matrix components to various loading conditions (the failure modes depend on which model you use); and
- can be used with damage evolution models with the following restriction: the LaRC05 model supports damage evolution only when used with enriched elements to model discontinuities (such as cracks) in an extended finite element method (XFEM) analysis.

Damage Initiation

Damage initiation refers to the onset of degradation at a material point. Abaqus supports two damage initiation criteria for unidirectional fiber-reinforced composites: the Hashin criterion and the LaRC05 criterion; and the ply fabric damage initiation criterion for bidirectional fabric-reinforced composites.

Damage is characterized by the degradation of material stiffness. It plays an important role in the analysis of fiber-reinforced composite materials.

Unidirectional fiber-reinforced composite materials exhibit elastic-brittle behavior; that is, damage in these materials is initiated without significant plastic deformation. Consequently, plasticity is often neglected when modeling the behavior of these materials. The fibers in the fiber-reinforced material are assumed to be parallel, as depicted in *Figure 1*.

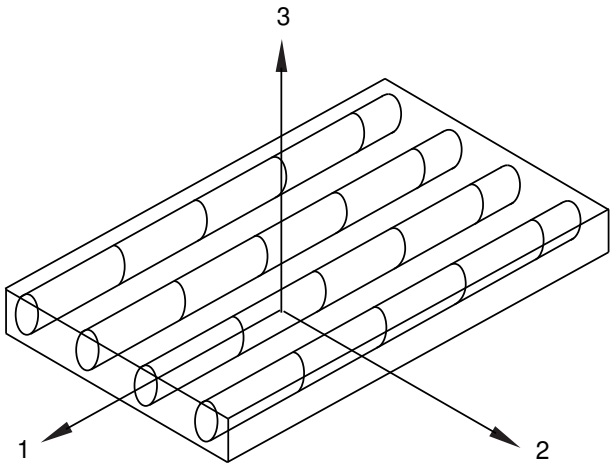


Figure 1: Unidirectional lamina.

For unidirectional fiber-reinforced composite materials, you must specify material properties in a user-defined local coordinate system. The lamina is in the 1–2 plane, and the local 1-direction corresponds to the fiber direction. You must specify the undamaged material response using one of the methods for defining an orthotropic linear elastic material (see *Linear Elastic Behavior*); the most convenient of which is the method for defining an orthotropic material in plane stress (see *Defining Orthotropic Elasticity in Plane Stress*). However, the material response can also be defined in terms of the engineering constants or by specifying the elastic stiffness matrix directly.

For bidirectional fabric-reinforced composite materials, the shear response is dominated by the nonlinear behavior of the matrix, which includes both plasticity and stiffness degradation due to matrix microcracking. The fiber directions are assumed to be orthogonal. You must specify material properties in a user-defined local coordinate system, with the local 1-direction and 2-direction aligned with the fiber directions, as shown in *#unique_1189/unique_1189_Connect_42_simamat-c-init-fabricreinforcedmaterial*. The material response along the fiber directions is characterized with damaged elasticity, and the model differentiates between tensile and compressive fiber failure modes. The elastic undamaged response of the material must be defined using the bilamina elasticity model (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*).

Hashin Criterion

The Abaqus anisotropic damage model for unidirectional fiber-reinforced composite is based on the work of *Matzenmiller et al. (1995)*, *Hashin and Rotem (1973)*, *Hashin (1980)*, and *Camanho and Davila (2002)*.

Four different modes of failure are considered:

- fiber rupture in tension;
- fiber buckling and kinking in compression;
- matrix cracking under transverse tension and shearing; and
- matrix crushing under transverse compression and shearing.

The initiation criteria have the following general forms:

Fiber tension $(\hat{\sigma}_{11} \geq 0)$:

$$F_f^t = \left(\frac{\hat{\sigma}_{11}}{X^T}\right)^2 + \alpha\left(\frac{\hat{\tau}_{12}}{S^L}\right)^2.$$

Fiber compression ($\widehat{\sigma}_{11} < 0$):

$$F_f^c = \left(\frac{\widehat{\sigma}_{11}}{X^C}\right)^2.$$

Matrix tension ($\widehat{\sigma}_{22} \geq 0$):

$$F_m^t = \left(\frac{\widehat{\sigma}_{22}}{Y^T}\right)^2 + \left(\frac{\hat{\tau}_{12}}{S^L}\right)^2.$$

Matrix compression ($\widehat{\sigma_{22}} < 0$):

$$F_m^c = \left(\frac{\widehat{\sigma}_{22}}{2S^T}\right)^2 + \left[\left(\frac{Y^C}{2S^T}\right)^2 - 1\right]\frac{\widehat{\sigma}_{22}}{Y^C} + \left(\frac{\hat{\tau}_{12}}{S^L}\right)^2.$$

In the above equations

X^T

denotes the longitudinal tensile strength;

X^C

denotes the longitudinal compressive strength;

Y^T

denotes the transverse tensile strength;

Y^C

denotes the transverse compressive strength;

S^L

denotes the longitudinal shear strength;

S^T

denotes the transverse shear strength;

α

is a coefficient that determines the contribution of the shear stress to the fiber tensile initiation criterion; and

$\widehat{\sigma}_{11}, \widehat{\sigma}_{22}, \hat{\tau}_{12}$

are components of the effective stress tensor, $\widehat{\boldsymbol{\sigma}}$, that is used to evaluate the initiation criteria and which is computed from:

$$\widehat{\boldsymbol{\sigma}} = \mathbf{M}\boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the true stress and $\mathbf{M}$ is the damage operator:

$$\mathbf{M} = \begin{bmatrix} \frac{1}{(1-d_f)} & 0 & 0 \\ 0 & \frac{1}{(1-d_m)} & 0 \\ 0 & 0 & \frac{1}{(1-d_s)} \end{bmatrix}.$$

d_f, d_m, and d_s are internal (damage) variables that characterize fiber, matrix, and shear damage, which are derived from damage variables d_f^t, d_f^c, d_m^t, and d_m^c, corresponding to the four modes previously discussed, as follows:

$$d_f = \begin{cases} d_f^t \text{ if } \quad \hat{\sigma}_{11} \geq 0, \\ d_f^c \text{ if } \quad \hat{\sigma}_{11} < 0, \end{cases}$$

$$d_m = \begin{cases} d_m^t \text{ if } \quad \hat{\sigma}_{22} \geq 0, \\ d_m^c \text{ if } \quad \hat{\sigma}_{22} < 0, \end{cases}$$

$$d_s = 1 - \left(1 - d_f^t\right)\left(1 - d_f^c\right)\left(1 - d_m^t\right)\left(1 - d_m^c\right).$$

Prior to any damage initiation and evolution the damage operator, $\mathbf{M}$, is equal to the identity matrix, so $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}$. Once damage initiation and evolution has occurred for at least one mode, the damage operator becomes significant in the criteria for damage initiation of other modes (see *Damage Evolution and Element Removal for Fiber-Reinforced Composites* for discussion of damage evolution). The effective stress, $\hat{\boldsymbol{\sigma}}$, is intended to represent the stress acting over the damaged area that effectively resists the internal forces.

The initiation criteria presented above can be specialized to obtain the model proposed in Hashin and Rotem (1973) by setting $\alpha = 0.0$ and $S^T = Y^C/2$ or the model proposed in Hashin (1980) by setting $\alpha = 1.0$.

An output variable is associated with each initiation criterion (fiber tension, fiber compression, matrix tension, matrix compression) to indicate whether the criterion has been met. A value of 1.0 or higher indicates that the initiation criterion has been met (see *Output* for further details). If you define a damage initiation model without defining an associated evolution law, the initiation criteria will affect only output. Thus, you can use these criteria to evaluate the propensity of the material to undergo damage without modeling the damage process.

Input File Usage: Use the following option to define the Hashin damage initiation criterion:

**DAMAGE INITIATION*, CRITERION=HASHIN, ALPHA=α
X^T, X^C, Y^T, Y^C, S^L, S^T

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Fiber-Reinforced Composites->Hashin Damage**

LaRC05 Criterion

The LaRC05 criterion (available only in Abaqus/Standard) for unidirectional fiber-reinforced composite is a three-dimensional model based on the theory of *Pinho et al. (2012)*. It can be applied generally to polymer-matrix fiber-reinforced composites. It can also be used for fiber-reinforced composite lamina similar to the Hashin

criterion. The criterion considers four different damage initiation mechanisms (matrix cracking, fiber kinking, fiber splitting, and fiber tension), as follows:

Matrix cracking:

$$F_m^{crack} = \sqrt{\left(\frac{\tau_T}{S_T - \eta_T \sigma_N}\right)^2 + \left(\frac{\tau_L}{S_L - \eta_L \sigma_N}\right)^2 + \left(\frac{\langle \sigma_N \rangle_+}{Y_T}\right)^2},$$

where

$$\begin{cases} \sigma_N &= \sigma_{22}\cos^2\alpha + \sigma_{33}\sin^2\alpha + \sigma_{23}\sin(2\alpha) \\ \tau_T &= \frac{1}{2}(\sigma_{33} - \sigma_{22})\sin(2\alpha) + \sigma_{23}\cos(2\alpha). \\ \tau_L &= \sigma_{12}\cos\alpha + \sigma_{31}\sin\alpha \end{cases}$$

The angle, α, represents the orientation of a critical plane perpendicular to the local 2–3 plane. The value of α is found numerically such that it maximizes F_m^{crack}.

Fiber tension $(\sigma_{11} \geq 0)$:

$$F_f^{tens} = \frac{\sigma_{11}}{X_T}$$

Fiber splitting $\left(-\frac{X_C}{2} \leq \sigma_{11} \leq 0\right)$:

$$F_f^{split} = \sqrt{\left(\frac{\tau_{23}^m}{S_T - \eta_T \sigma_2^m}\right)^2 + \left(\frac{\tau_{12}^m}{S_L - \eta_L \sigma_2^m}\right)^2 + \left(\frac{\langle \sigma_2^m \rangle_+}{Y_T}\right)^2}$$

Fiber kinking $\left(\sigma_{11} \leq -\frac{X_C}{2}\right)$:

$$F_f^{kink} = \sqrt{\left(\frac{\tau_{23}^m}{S_T - \eta_T \sigma_2^m}\right)^2 + \left(\frac{\tau_{12}^m}{S_L - \eta_L \sigma_2^m}\right)^2 + \left(\frac{\langle \sigma_2^m \rangle_+}{Y_T}\right)^2}$$

In the fiber splitting and fiber kinking equations above, stresses in the fiber-misalignment frame are computed as:

$$\begin{cases} \sigma_2^m &= \sin^2\varphi\sigma_{11} + \cos^2\varphi\sigma_{22}^\psi - 2\sin\varphi\cos\varphi\tau_{12}^\psi \\ \tau_{12}^m &= \sin\varphi\cos\varphi\left(\sigma_{22}^\psi - \sigma_{11}\right) + \left(\cos^2\varphi - \sin^2\varphi\right)\tau_{12}^\psi, \\ \tau_{23}^m &= \tau_{23}^\psi\cos\varphi - \tau_{31}^\psi\sin\varphi \end{cases}$$

where σ_{ij}^ψ represents the stresses in the fiber-kinking plane:

$$\begin{cases} \sigma_{22}^\psi &= \cos^2\psi\sigma_{22} + \sin^2\psi\sigma_{33} + 2\sin\psi\cos\psi\tau_{23} \\ \tau_{12}^\psi &= \tau_{12}\cos\psi + \tau_{31}\sin\psi, \quad \tau_{31}^\psi = \tau_{31}\cos\psi - \tau_{12}\sin\psi. \\ \tau_{23}^\psi &= \sin\psi\cos\psi(\sigma_{33} - \sigma_{22}) + (\cos^2\psi - \sin^2\psi)\tau_{23} \end{cases}$$

For the three fiber-failure mechanisms above, Abaqus/Standard first checks the sign of σ_{11}. If σ_{11} is positive, the fiber tension criterion, F_f^{tens}, is calculated; otherwise, for fiber splitting and kinking, stresses are rotated to the fiber-kinking plane, and then these obtained stresses are further rotated to the misalignment frame. During this process, the angles ψ and φ are varied together until the maximum value of F_f^{split} or F_f^{kink} is found.

In the above equations:

X_T

denotes the longitudinal tensile strength;

X_C

denotes the longitudinal compressive strength;

Y_T

denotes the transverse tensile strength;

Y_C

denotes the transverse compressive strength;

S_L

denotes the plane shear strength;

S_T

denotes the transverse shear strength;

α_0

denotes the fracture plane angle for pure compression (by default, α_0=53°);

η_L

denotes the longitudinal shear friction coefficient; and

η_T

denotes the transverse shear friction coefficient.

If η_L and η_T are not defined, they are calculated as:

$$\eta_L = \frac{S_L \cos(2\alpha_0)}{Y_C \cos^2\alpha_0},\ \eta_T = -\frac{1}{\tan(2\alpha_0)}.$$

An output variable is associated with each initiation criterion (matrix cracking, fiber kinking, fiber splitting, fiber tension) to indicate whether the criterion has been met. A value of 1.0 or higher indicates that the initiation criterion has been met (see *Output*). The initiation criteria affect only output. Therefore, you can use these criteria to evaluate the propensity of the material to undergo damage without modeling the damage process.

Input File Usage: Use the following option to define the LaRC05 damage initiation criterion:

**DAMAGE INITIATION*, CRITERION=LARC05
X_T, X_C, Y_T, Y_C, S_L, α_0, φ_0, S_T
η_L, η_T

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Fiber-Reinforced Composites->LaRC05 Damage**

Ply Fabric Criterion

The ply fabric damage initiation criterion for bidirectional fiber-reinforced composite is an anisotropic damage model based on the work of *Johnson, 2001* and *Sokolinsky et al., 2011*. The material response along the fiber directions is characterized with damaged elasticity. The model incorporates different initial (undamaged) stiffness in tension and compression (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*), and differentiates between tensile and compressive fiber failure modes by activating the corresponding damage variable depending on the stress state in the fiber directions. The shear response is dominated by the nonlinear behavior of the matrix, which includes both plasticity (see *Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials*) and stiffness degradation due to matrix microcracking.

Four different modes of damage initiation for the fibers and one for shear response of the matrix are considered:

- Fiber rupture in tension in the local 1-direction, with a damage variable d_{1+};
- Fiber buckling and kinking in compression in the local 1-direction, with a damage variable d_{1-};
- Fiber rupture in tension in the local 2-direction, with a damage variable d_{2+};
- Fiber buckling and kinking in compression in the local 2-direction, with a damage variable d_{2-}; and
- Matrix microcracking due to shear deformation, with a damage variable d_{12}.

The four fiber damage initiation criteria have the following general form:

$$F_\alpha = \phi_\alpha - r_\alpha.$$

In the above equation, an index α is used to simplify notation and is used in subsequent discussions, such that it takes the values $1+, 1-, 2+, 2-$ depending on the sign of the corresponding stresses.

In the above equation:

ϕ_α

are functions that provide initiation criteria for fiber damage and are assumed to take the form:

$$\phi_\alpha = \frac{\widehat{\sigma}_\alpha}{X_\alpha},$$

where X_α are the tensile/compressive strengths for uniaxial loading along the fiber directions, and $\widehat{\sigma}_\alpha$ are effective stresses defined as:

$$\widehat{\sigma}_{1+} = \frac{<\sigma_{11}>}{(1-d_{1+})}, \qquad \widehat{\sigma}_{1-} = \frac{<-\sigma_{11}>}{(1-d_{1-})},$$
$$\widehat{\sigma}_{2+} = \frac{<\sigma_{22}>}{(1-d_{2+})}, \qquad \widehat{\sigma}_{2-} = \frac{<-\sigma_{22}>}{(1-d_{2-})}.$$

The symbol $\langle\rangle$ in the equations above represents the Macaulay bracket operator, which is defined for every $x \in \Re$ as $\langle x \rangle = (x + |x|)/2$.

r_α

are damage thresholds that are initially set to one. After damage activation ($\phi_\alpha = 1$), they increase with increasing damage according to:

$$r_\alpha(t) = \max_{\tau \le t} \phi_\alpha(t).$$

The definition ensures that the damage thresholds are nondecreasing quantities.

The damage initiation criterion for matrix shear failure has the following form:

$$F_{12} = \phi_{12} - r_{12}.$$

In the above equation:

ϕ_{12}

is the matrix shear failure criterion and is assumed to take the form:

$$\phi_{12} = \max\left(\frac{\hat{\sigma}_{12}}{S}, \frac{\bar{\varepsilon}^{pl}}{\bar{\varepsilon}^{pl}_{\max}}\right),$$

where S is the shear stress for initial matrix damage, $\hat{\sigma}_{12}$ is the effective shear stress defined as $\hat{\sigma}_{12} = \sigma_{12}/(1 - d_{12})$, $\bar{\varepsilon}^{pl}$ is the equivalent plastic strain due to shear deformation, and $\bar{\varepsilon}^{pl}_{\max}$ is a specified maximum value. The damage shear stress-strain relations are computed based as discussed in *Plasticity Model for Bidirectional Fabric-Reinforced Composite Materials*.

r_{12}

is the shear damage threshold that is initially set to one. After damage activation ($\phi_{12} = 1$), it increases with increasing damage according to:

$$r_{12}(t) = \max_{\tau \le t} \phi_{12}(t).$$

Once a particular damage initiation criterion is satisfied, the material stiffness is degraded according to the specified damage evolution law for that criterion. For a discussion of damage evolution, see *Damage Evolution and Element Removal for Fiber-Reinforced Composites*.

An output variable is associated with each initiation criterion (fiber tension in the local 1-direction, fiber compression in the local 1-direction, fiber tension in the local 2-direction, fiber compression in the local 2-direction, and matrix shear) to indicate whether the criterion has been met. A value of 1.0 or higher indicates that the initiation criterion has been met (for more details, see *Output*). If you define a damage initiation model without defining an associated evolution law, the initiation criteria affects only output. Therefore, you can use these criteria to evaluate the propensity of the material to undergo damage without modeling the damage process.

Input File Usage: Use the following option to define the ply fabric damage initiation criterion:

**DAMAGE INITIATION*, CRITERION=PLY FABRIC
X_{1+}, X_{1-}, X_{2+}, X_{2-}, S

Abaqus/CAE Usage: Defining the ply fabric damage initiation criterion is not supported in Abaqus/CAE.

Elements

The Hashin and ply fabric damage initiation criterion must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements. The LaRC05damage initiation criterion can be used with three-dimensional solid, plane stress, shell, solid shell, and membrane elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables relate specifically to damage initiation at a material point in the fiber-reinforced composite damage model:

DMICRT All damage initiation criteria components.

HSNFTCRT Maximum value of the fiber tensile initiation criterion experienced during the analysis.

HSNFCCRT Maximum value of the fiber compressive initiation criterion experienced during the analysis.

HSNMTCRT Maximum value of the matrix tensile initiation criterion experienced during the analysis.

HSNMCCRT Maximum value of the matrix compressive initiation criterion experienced during the analysis.

LARCMCCRT Maximum value of the matrix cracking initiation criterion experienced during the analysis.

LARCFKCRT Maximum value of the fiber kinking initiation criterion experienced during the analysis.

LARCFSCRT Maximum value of the fiber splitting initiation criterion experienced during the analysis.

LARCFTCRT Maximum value of the fiber tension initiation criterion experienced during the analysis.

PLF1TCRT Maximum value of the fiber tensile initiation criterion in the local 1-direction experienced during the analysis.

PLF1CCRT Maximum value of the fiber compressive initiation criterion in the local 1-direction experienced during the analysis.

PLF2TCRT Maximum value of the fiber tensile initiation criterion in the local 2-direction experienced during the analysis.

PLF2CCRT Maximum value of the fiber compressive initiation criterion in the local 2-direction experienced during the analysis.

PLSHRCRT Maximum value of the matrix shear initiation criterion experienced during the analysis.

For the variables above that indicate whether an initiation criterion in a damage mode has been satisfied or not, a value that is less than 1.0 indicates that the criterion has not been satisfied, while a value of 1.0 or higher

indicates that the criterion has been satisfied. If you define a damage evolution model, the maximum value of this variable does not exceed 1.0. However, if you do not define a damage evolution model, this variable can have values higher than 1.0, which indicates by how much the criterion has been exceeded.

References

- Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, vol. 47, pp. 329–334, 1980.
- Hashin, Z., and A. Rotem, "A Fatigue Criterion for Fiber-Reinforced Materials," *Journal of Composite Materials*, vol. 7, pp. 448–464, 1973.
- Johnson, A. F., "Modelling Fabric-Reinforced Composites under Impact Loads," *Composites Part A: Applied Science and Manufacturing*, vol. 32, no. 9, pp. 1197–1206, 2001.
- Lapczyk, I., and J. A. Hurtado, "Progressive Damage Modeling in Fiber-Reinforced Materials," *Composites Part A: Applied Science and Manufacturing*, vol. 38, no. 11, pp. 2333–2341, 2007.
- Pinho, S. T., R. Darvizeh, P. Robinson, C. Schuecker, and P. P. Camanho, "Material and Structural Response of Polymer-Matrix Fibre-Reinforced Composites," *Journal of Composite Materials*, vol. 46, no. 19–20, pp. 2313–2341, 2012.
- Sokolinsky, V. S., K. C. Indermuehle, and J. A. Hurtado, "Numerical Simulation of the Crushing Process of a Corrugated Composite Plate," *Composites Part A: Applied Science and Manufacturing*, vol. 42, no. 9, pp. 1119–1126, 2011.

Damage Evolution and Element Removal for Fiber-Reinforced Composites

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Fiber-Reinforced Composites*
- **DAMAGE EVOLUTION*
- *Damage evolution*

Overview

The damage evolution capability for fiber-reinforced materials in Abaqus:

- assumes that damage is characterized by progressive degradation of material stiffness, leading to material failure;
- requires linearly elastic behavior of the undamaged material (see *Linear Elastic Behavior*);
- can be used with the Hashin damage initiation criterion for unidirectional fiber-reinforced composites to take into account four different failure modes: fiber tension, fiber compression, matrix tension, and matrix compression, with four damage variables to describe damage for each failure mode;
- can be used with the LaRC05 damage initiation criterion for unidirectional fiber-reinforced composites in Abaqus/Standard when used with enriched elements to model discontinuities (such as cracks) in an extended finite element method (XFEM) analysis (see *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*);
- can be used with the ply fabric damage initiation criterion for bidirectional fabric-reinforced composites to take into account five different failure modes: four modes of fiber failure and one of matrix shear failure, with five damage variables to describe damage for each failure mode;
- is based on energy dissipation during the damage process;
- offers options for what occurs upon failure, including the removal of elements from the mesh; and
- can be used with a viscous regularization of the constitutive equations to improve the convergence rate in the softening regime if used with the Hashin damage initiation criterion.

Damage Evolution for the Hashin Model

The previous section (*Damage Initiation for Fiber-Reinforced Composites*) discussed the damage initiation in plane stress fiber-reinforced composites. This section will discuss the post-damage initiation behavior for cases in which a damage evolution model has been specified. Prior to damage initiation the material is linearly elastic, with the stiffness matrix of a plane stress orthotropic material. Thereafter, the response of the material is computed from

$$\sigma = \mathbf{C_d} \varepsilon^{el},$$

where ε^{el} is the elastic strain and $\mathbf{C_d}$ is the damaged elasticity matrix, which has the form

$$\mathbf{C_d} = \frac{1}{D} \begin{bmatrix} (1-d_f)\,E_1 & (1-d_f)\,(1-d_m)\,\nu_{21} E_1 & 0 \\ (1-d_f)\,(1-d_m)\,\nu_{12} E_2 & (1-d_m)\,E_2 & 0 \\ 0 & 0 & (1-d_s)\,GD \end{bmatrix},$$

where $D = 1 - (1 - d_f)\,(1 - d_m)\,\nu_{12}\nu_{21}$, d_f reflects the current state of fiber damage, d_m reflects the current state of matrix damage, d_s reflects the current state of shear damage, E_1 is the Young's modulus in the fiber direction, E_2 is the Young's modulus in the matrix direction, G is the shear modulus, and ν_{12} and ν_{21} are Poisson's ratios.

The damage variables d_f, d_m, and d_s are derived from damage variables d_f^t, d_f^c, d_m^t, and d_m^c, corresponding to the four failure modes previously discussed, as follows:

$$d_f = \begin{cases} d_f^t \text{ if } \quad \hat{\sigma}_{11} \geq 0, \\ d_f^c \text{ if } \quad \hat{\sigma}_{11} < 0, \end{cases}$$

$$d_m = \begin{cases} d_m^t \text{ if } \quad \hat{\sigma}_{22} \geq 0, \\ d_m^c \text{ if } \quad \hat{\sigma}_{22} < 0, \end{cases}$$

$$d_s = 1 - \left(1 - d_f^t\right)\left(1 - d_f^c\right)\left(1 - d_m^t\right)\left(1 - d_m^c\right),$$

$\hat{\sigma}_{11}$ and $\hat{\sigma}_{22}$ are components of the effective stress tensor. The effective stress tensor is primarily used to evaluate damage initiation criteria; see *Damage Initiation for Fiber-Reinforced Composites* for a description of how the effective stress tensor is computed.

Evolution of Damage Variables for Each Mode

To alleviate mesh dependency during material softening, Abaqus introduces a characteristic length into the formulation, so that the constitutive law is expressed as a stress-displacement relation. The damage variable evolves such that the stress-displacement behaves as shown in *Figure 1* in each of the four failure modes. The positive slope of the stress-displacement curve prior to damage initiation corresponds to linear elastic material behavior; the negative slope after damage initiation is achieved by evolution of the respective damage variables according to the equations shown below.

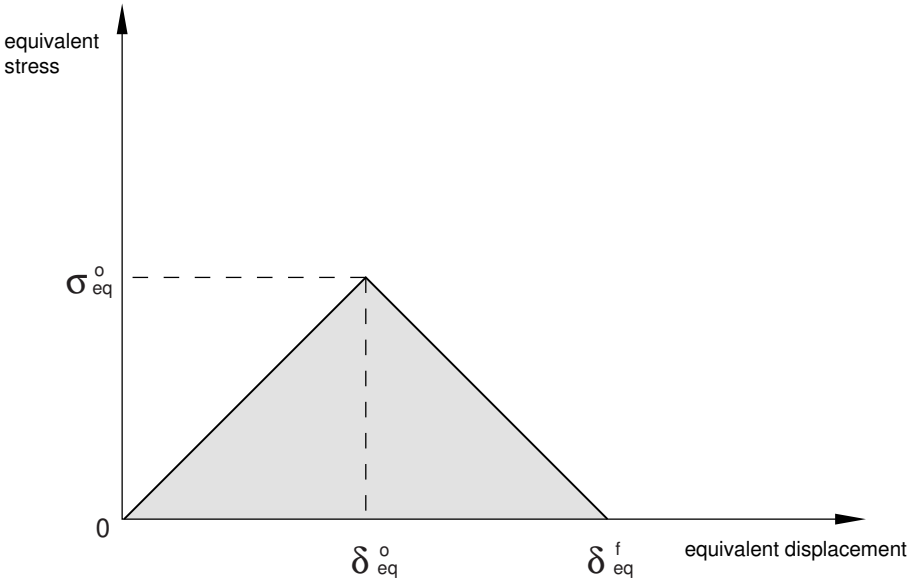


Figure 1: Equivalent stress versus equivalent displacement.

Equivalent displacement and stress for each of the four damage modes are defined as follows:

Fiber tension $(\hat{\sigma}_{11} \geq 0)$:

$$\delta_{eq}^{ft} = L^c \sqrt{\langle \varepsilon_{11} \rangle^2 + \alpha \varepsilon_{12}^2},$$
$$\sigma_{eq}^{ft} = \frac{\langle \sigma_{11} \rangle \langle \varepsilon_{11} \rangle + \alpha \tau_{12} \varepsilon_{12}}{\delta_{eq}^{ft} / L^c},$$

Fiber compression $(\hat{\sigma}_{11} < 0)$:

$$\delta_{eq}^{fc} = L^c \langle -\varepsilon_{11} \rangle,$$
$$\sigma_{eq}^{fc} = \frac{\langle -\sigma_{11} \rangle \langle -\varepsilon_{11} \rangle}{\delta_{eq}^{fc} / L^c}.$$

Matrix tension $(\hat{\sigma}_{22} \geq 0)$:

$$\delta_{eq}^{mt} = L^c \sqrt{\langle \varepsilon_{22} \rangle^2 + \varepsilon_{12}^2},$$
$$\sigma_{eq}^{mt} = \frac{\langle \sigma_{22} \rangle \langle \varepsilon_{22} \rangle + \tau_{12} \varepsilon_{12}}{\delta_{eq}^{mt} / L^c}.$$

Matrix compression $(\hat{\sigma}_{22} < 0)$:

$$\delta_{eq}^{mc} = L^c \sqrt{\langle -\varepsilon_{22} \rangle^2 + \varepsilon_{12}^2},$$
$$\sigma_{eq}^{mc} = \frac{\langle -\sigma_{22} \rangle \langle -\varepsilon_{22} \rangle + \tau_{12} \varepsilon_{12}}{\delta_{eq}^{mc} / L^c}.$$

The characteristic length, L^c, is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For membranes and shells it is a characteristic length in the reference surface, computed as the square root of the area. Alternatively, this characteristic length could be defined as a function of the element topology and material orientation in user subroutine *VUCHARLENGTH* (see *Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit*). The symbol $\langle \rangle$ in the equations above represents the Macaulay bracket operator, which is defined for every $\alpha \in \Re$ as $\langle \alpha \rangle = (\alpha + |\alpha|)/2$.

After damage initiation (that is, $\delta_{eq} \geq \delta_{eq}^0$) for the behavior shown in *Figure 1*, the damage variable for a particular mode is given by the following expression

$$d = \frac{\delta_{eq}^f \left(\delta_{eq} - \delta_{eq}^0 \right)}{\delta_{eq} \left(\delta_{eq}^f - \delta_{eq}^0 \right)},$$

where δ_{eq}^0 is the initial equivalent displacement at which the initiation criterion for that mode was met and δ_{eq}^f is the displacement at which the material is completely damaged in this failure mode. The above relation is presented graphically in *Figure 2*.

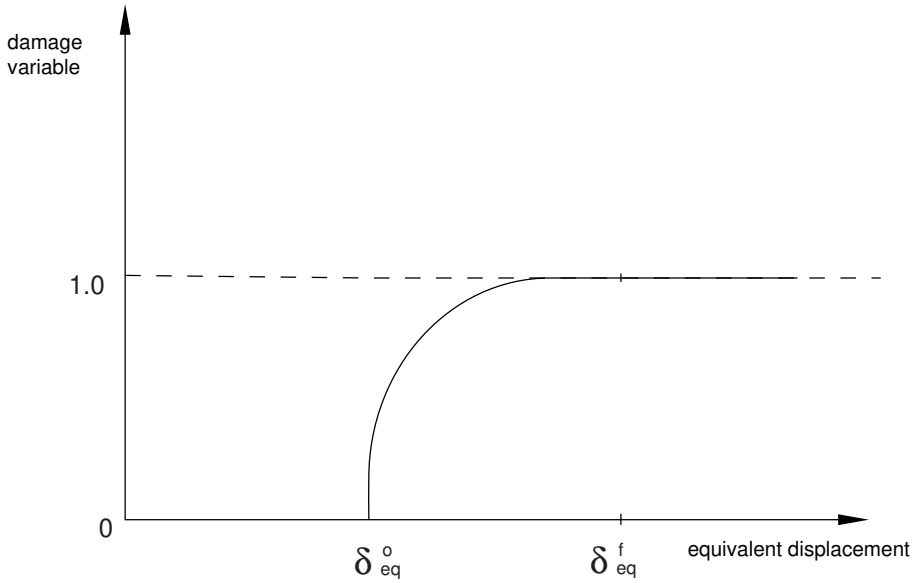


Figure 2: Damage variable as a function of equivalent displacement.

The values of δ_{eq}^{0} for the various modes depend on the elastic stiffness and the strength parameters specified as part of the damage initiation definition (see *Damage Initiation for Fiber-Reinforced Composites*). For each failure mode you must specify the energy dissipated due to failure, G^c, which corresponds to the area of the triangle OAC in *Figure 3*.

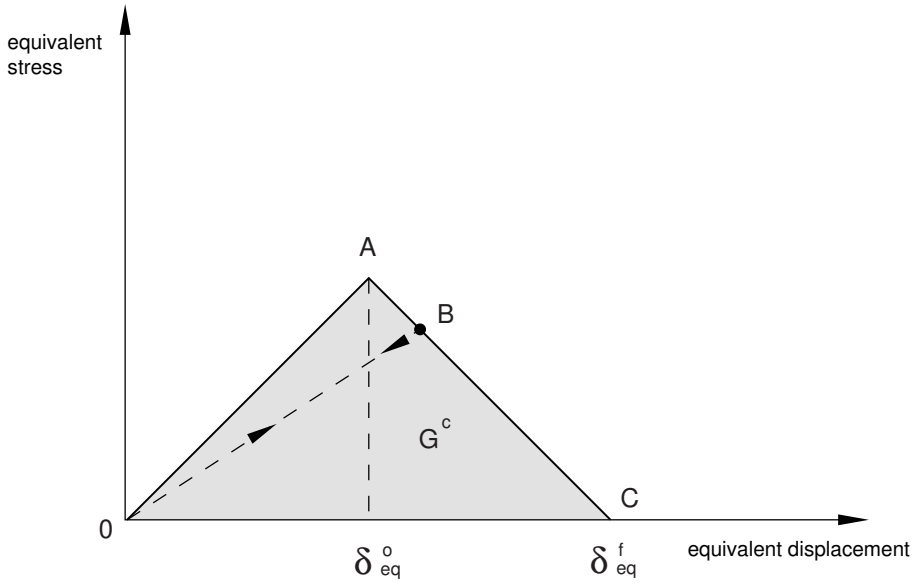


Figure 3: Linear damage evolution.

The values of δ_{eq}^{f} for the various modes depend on the respective G^c values.

Unloading from a partially damaged state, such as point B in *Figure 3*, occurs along a linear path toward the origin in the plot of equivalent stress versus equivalent displacement; this same path is followed back to point B upon reloading as shown in the figure.

Input File Usage: Use the following option to define the damage evolution law:

*DAMAGE EVOLUTION, TYPE=ENERGY, SOFTENING=LINEAR

G_{ft}^c, G_{fc}^c, G_{mt}^c, G_{mc}^c

where G_{ft}^c, G_{fc}^c, G_{mt}^c, and G_{mc}^c are energies dissipated during damage for fiber tension, fiber compression, matrix tension, and matrix compression failure modes, respectively.

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damage for Fiber-Reinforced Composites->Hashin Damage**: **Suboptions->Damage Evolution**: **Type:Energy**: **Softening:Linear**

Damage Evolution for the Ply Fabric Model

The ply fabric damage initiation criterion for bidirectional fabric-reinforced composites is described in *Damage Initiation for Fiber-Reinforced Composites*. This section discusses the post-damage initiation behavior for cases in which a damage evolution model is specified.

The elastic stress-strain relations are given by orthotropic damaged bilamina elasticity (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*). Thereafter, the damaged elastic response of the plane stress orthotropic material in the local orthogonal coordinate is computed from

$$\sigma = \mathbf{C_d}\varepsilon^{el},$$

where ε^{el} is the elastic strain, and $\mathbf{C_d}$ is the damaged elasticity matrix. The damaged elasticity matrix has the form

$$\mathbf{C_d} = \frac{1}{D}\begin{bmatrix} (1-d_1)E_1 & (1-d_1)(1-d_2)\nu_{21}E_1 & 0 \\ (1-d_1)(1-d_2)\nu_{12}E_2 & (1-d_2)E_2 & 0 \\ 0 & 0 & (1-d_{12})G_{12}D \end{bmatrix},$$

where $D = 1-(1-d_1)(1-d_2)\nu_{12}\nu_{21}$, d_1 reflects the current state of fiber damage in the local 1-direction, d_2 reflects the current state of fiber damage in the local 2-direction, d_{12} reflects the current state of matrix shear damage, E_1 is the Young's modulus in the local 1-direction, E_2 is the Young's modulus in the local 2-direction, G_{12} is the shear modulus, and ν_{12} and ν_{21} are Poisson's ratios.

The model differentiates between tensile and compressive fiber failure modes by activating the corresponding damage variable depending on the stress state in the fiber directions. Thus:

$$d_1 = d_{1+}\frac{<\sigma_{11}>}{\sigma_{11}} + d_{1-}\frac{<-\sigma_{11}>}{\sigma_{11}},$$

and

$$d_2 = d_{2+}\frac{<\sigma_{22}>}{\sigma_{22}} + d_{2-}\frac{<-\sigma_{22}>}{\sigma_{22}}.$$

The symbol $\langle\rangle$ in the equations above represents the Macaulay bracket operator. For the four damage variables $d_\alpha, (\alpha \in \{1+, 1-, 2+, 2-\})$ are associated with fiber failure. An index α is introduced to simplify notation and is used in subsequent discussions.

To incorporate different initial (undamaged) stiffness in tension and compression, the values of the elastic constants E_1, E_2, and ν_{12} are interpolated between their tensile values (E_{1+}, E_{2+}, and ν_{12+}) or compressive values (E_{1-}, E_{2-}, and ν_{12-}) depending on the state of the material. You must define these elastic constants

using bilamina elasticity (see *Defining Orthotropic Elasticity in Plane Stress with Different Moduli in Tension and Compression*).

Evolution of Damage Variables for Each Mode

For the four damage variables d_α, $(\alpha \in \{1+, 1-, 2+, 2-\})$ associated with fiber failure, the evolution of these damage variables is a function of the damage thresholds and the fracture energy per unit area under uniaxial tensile/compressive loading, $G_{f\alpha}$. The evolution of the damage variables is given by the equation:

$$d_\alpha = 1 - \frac{1}{r_\alpha} e^{-(A_\alpha (r_\alpha - 1))}, \qquad \dot{d}_\alpha \geq 0,$$

where

$$A_\alpha = \frac{2 g_0^\alpha L^c}{G_{f\alpha} - g_0^\alpha L^c}.$$

Here, L^c is the characteristic length of the element, $G_{f\alpha}$ is the fracture energy per unit area under uniaxial tensile/compressive loading, and g_0^α is the elastic energy density (that is. per unit volume) at the point of damage initiation:

$$g_0^\alpha = \frac{X_\alpha^2}{2E_\alpha},$$

where X_α are the tensile/compressive strengths for uniaxial loading along the fiber directions. The formulation of the damage evolution law ensures that the damage variables are monotonically increasing quantities. It also ensures that the correct amount of energy is dissipated when the lamina is subjected to uniaxial loading conditions along the fiber directions.

The characteristic length, L^c, is based on the element geometry and formulation. It is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For membrane and shell elements it is a characteristic length in the reference surface, computed as the square root of the area. Alternatively, you can define this characteristic length as a function of the element topology and material orientation in user subroutine *VUCHARLENGTH* (see *Defining the Characteristic Element Length at a Material Point in Abaqus/Explicit*).

It is assumed that the shear damage variable d_{12} increases with the logarithm of r_{12} until a maximum value $d_{12}^{\max}$ is reached. Thus:

$$d_{12} = \min(\alpha_{12} \ln(r_{12}), d_{12}^{\max}),$$

where $\alpha_{12} > 0$, and $d_{12}^{\max} \leq 1$ are material properties.

Input File Usage: Use the following option to define the damage evolution law:

**DAMAGE EVOLUTION*, TYPE=ENERGY, SOFTENING=EXPONENTIAL

$G_{f1+}, G_{f1-}, G_{f2+}, G_{f2-}, \alpha_{12}, d_{12}^{\max}$

Abaqus/CAE Usage: The ply fabric model is not supported in Abaqus/CAE.

Calibration Procedure

You can measure the elastic constants and the fiber tensile/compressive strengths, X_α, from standard coupon tests in uniaxial tension/compression loading of $0°/90°$ laminates. The calibration of damage evolution in the

fiber failure modes is based on the fracture energy per unit area of the material, $G_{f\alpha}$, which can be measured experimentally.

The shear response is usually calibrated with a cyclic tensile test on a $\pm 45°$ laminate, where the strains along the fiber directions can be neglected. *Figure 4* shows the typical shear response of a fabric-reinforced composite. The unloading/reloading paths correspond to an idealization of the actual response, which usually exhibits hysteretic behavior. *Figure 4* serves as a starting point for the discussion of a general calibration procedure for the parameters that enter the damage and plasticity equations.

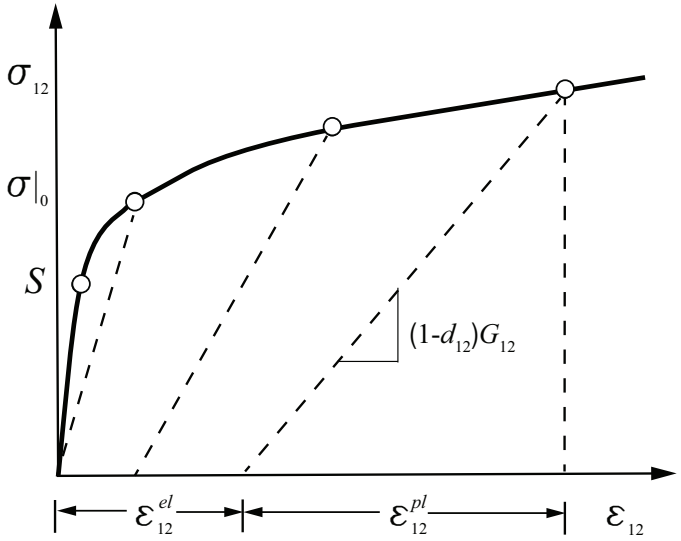


Figure 4: Shear response of a bidirectional fabric-reinforced composite.

The level of damage can be measured from the ratio of the unloading stiffness to the initial (undamaged) elastic stiffness. You can compute pairs of stress-damage values, (σ_{12}, d_{12}), for each unloading curve. This data can be represented in the space of d_{12} versus $\ln(\hat{\sigma}_{12})$, where $\hat{\sigma}_{12} = \sigma_{12}/(1 - d_{12})$. A linear fit of the data provides the values of α_{12} (slope of the line) and S (intersection with the horizontal axis), as shown in *Figure 5*. Sometimes the damage data show indication of a saturation value, which would be used to determine $d_{12}^{\max} \leq 1$. Otherwise, you should use a value of $d_{12}^{\max} = 1$.

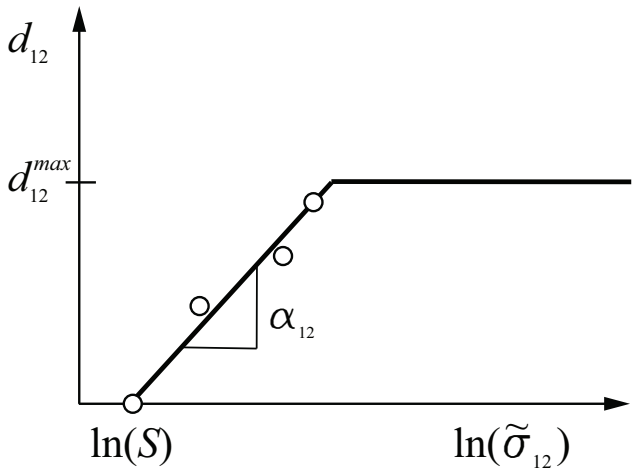


Figure 5: Calibration of the shear damage parameters.

Finally, for each unloading curve in *Figure 4*, the plastic strain ε_{12}^{pl} at the onset of unloading is determined from the value of residual deformation in the unloaded state. The values of $\left(\hat{\sigma}_{12}, \varepsilon_{12}^{pl}\right)$ at the onset of unloading are then used to fit the Johnson-Cook parameters of the yield function or, alternatively, they can be used directly as tabular input of the hardening curve.

Maximum Degradation and Choice of Element Removal

You have control over how Abaqus treats elements with severe damage. By default, the upper bound to all damage variables at a material point is $d_{max} = 1.0$. You can reduce this upper bound as discussed in *Controlling Element Deletion and Maximum Degradation for Materials with Damage Evolution*.

In Abaqus/Standard a material point is considered to fail once all failure modes reach d_{max}. In Abaqus/Explicit a material point is assumed to fail when either of the damage variables associated with fiber failure modes (tensile or compressive) reaches d_{max}. By default, an element is deleted from a mesh upon material failure. Details for element deletion driven by material failure are described in *Material Failure and Element Deletion*. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Alternatively, you can specify that an element should remain in the model even after all the damage variables reach d_{max} (see *Controlling Element Deletion and Maximum Degradation for Materials with Damage Evolution*). In this case, once all the damage variables reach the maximum value, the stiffness, $\mathbf{C_d}$, remains constant (see the expression for $\mathbf{C_d}$ earlier in this section).

For ply fabric damage evolution, you have more control over the criteria for material failure. You can consider a material point to fail if any one of the following criteria is met:

- when either fiber damage variable reaches a specified maximum value $d_1 = d_{max}$ or $d_2 = d_{max}$, or when both fiber damage variables reach the maximum specified value $d_1 = d_{max}$ and $d_2 = d_{max}$;
- when the shear plastic strain reaches a specified maximum value, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}_{\mathbf{max}}^{pl}$; or

- when a deformation-based material failure criterion depending on the values of the maximum ($\hat{\varepsilon}_{\max} > 0$) and minimum ($\hat{\varepsilon}_{\min} < 0$) principal logarithmic strains that the material can sustain is met.

If you do not define conditions that trigger material failure for the ply fabric model, the material point fails when either fiber damage variable reaches a specified maximum value.

Input File Usage: Use the following option to define the conditions that trigger material failure for the ply fabric model:

**PLY FABRIC FAILURE*, NUM FIBERS

$\bar{\varepsilon}^{pl}_{\max}$, $\hat{\varepsilon}_{\max}$, $\hat{\varepsilon}_{\min}$

Abaqus/CAE Usage: The ply fabric model is not supported in Abaqus/CAE.

Viscous Regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs, such as Abaqus/Standard. You can overcome some of these convergence difficulties using the viscous regularization scheme, which causes the tangent stiffness matrix of the softening material to be positive for sufficiently small time increments.

In this regularization scheme a viscous damage variable is defined by the evolution equation:

$$\dot{d}_v = \frac{1}{\eta}\left(d - d_v\right),$$

where η is the viscosity coefficient representing the relaxation time of the viscous system and d is the damage variable evaluated in the inviscid backbone model. The damaged response of the viscous material is given as

$$\sigma = \mathbf{C_d}\epsilon,$$

where the damaged elasticity matrix, $\mathbf{C_d}$, is computed using viscous values of damage variables for each failure mode. Using viscous regularization with a small value of the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime, without compromising results. The basic idea is that the solution of the viscous system relaxes to that of the inviscid case as $t/\eta \to \infty$, where t represents time.

Viscous regularization is also available in Abaqus/Explicit for the Hashin damage model. Viscous regularization slows down the rate of increase of damage and leads to increased fracture energy with increasing deformation rates, which can be exploited as an effective method of modeling rate-dependent material behavior.

In Abaqus/Standard the approximate amount of energy associated with viscous regularization over the whole model or over an element set is available using output variable ALLCD.

You cannot use the viscous regularization scheme with damage evolution for the ply fabric damage initiation criterion.

Defining Viscous Regularization Coefficients

You can specify different values of viscous coefficients for different failure modes.

Input File Usage: Use the following option to define viscous coefficients:

**DAMAGE STABILIZATION*

η_{ft}, η_{fc}, η_{mt}, η_{mc}

where $\eta_{ft}, \eta_{fc}, \eta_{mt}, \eta_{mc}$ are viscosity coefficients for fiber tension, fiber compression, matrix tension, and matrix compression failure modes, respectively.

Abaqus/CAE Usage: Use the following input to define the viscous coefficients for fiber-reinforced materials:

Property module: material editor: **Mechanical->Damage for Fiber-Reinforced Composites->Hashin Damage**: **Suboptions->Damage Stabilization**

Applying a Single Viscous Coefficient in Abaqus/Standard

Alternatively, in Abaqus/Standard you can specify the viscous coefficients as part of a section controls definition. In this case, the same viscous coefficient is applied to all failure modes. For more information, see *Using Viscous Regularization with Cohesive Elements, Connector Elements, and Elements That Can Be Used with the Damage Evolution Models for Ductile Metals and Fiber-Reinforced Composites Including the Multiscale Model for Fiber-Reinforced Composites in Abaqus/Standard*.

Material Damping

If stiffness proportional damping is specified in combination with the damage evolution law for fiber-reinforced materials, Abaqus calculates the damping stresses using the damaged elastic stiffness.

Elements

The damage evolution law for fiber-reinforced materials must be used with elements with a plane stress formulation, which include plane stress, shell, continuum shell, and membrane elements.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables relate specifically to damage evolution in the fiber-reinforced composite damage model:

STATUS Status of the element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).

STATUSMP Status of each material point in the element (1.0 if a material point is active, 0.0 if it is not). Abaqus/Explicit only.

DAMAGEFT Fiber tensile damage variable for the Hashin model.

DAMAGEFC Fiber compressive damage variable for the Hashin model.

DAMAGEMT Matrix tensile damage variable for the Hashin model.

DAMAGEMC Matrix compressive damage variable for the Hashin model.

DAMAGESHR Shear damage variable for the Hashin model and ply fabric model.

DAMAGEF1T	Fiber tensile damage variable in the local 1-direction for the ply fabric model.
DAMAGEF1C	Fiber compressive damage variable in the local 1-direction for the ply fabric model.
DAMAGEF2T	Fiber tensile damage variable in the local 2-direction for the ply fabric model.
DAMAGEF2C	Fiber compressive damage variable in the local 2-direction for the ply fabric model.
EDMDDEN	Energy dissipated per unit volume in the element by damage.
ELDMD	Total energy dissipated in the element by damage.
DMENER	Energy dissipated per unit volume by damage.
ALLDMD	Energy dissipated in the whole (or partial) model by damage.
ALLPD	Energy dissipated in the whole (or partial) model by shear plasticity in the ply fabric model.
ECDDEN	Energy per unit volume in the element that is associated with viscous regularization.
ELCD	Total energy in the element that is associated with viscous regularization.
CENER	Energy per unit volume that is associated with viscous regularization.
ALLCD	The approximate amount of energy over the whole model or over an element set that is associated with viscous regularization.

References

- Lapczyk, I., and J. A. Hurtado, "Progressive Damage Modeling in Fiber-Reinforced Materials," *Composites Part A: Applied Science and Manufacturing*, vol. 38, no. 11, pp. 2333–2341, 2007.
- Sokolinsky, V. S., K. C. Indermuehle, and J. A. Hurtado, "Numerical Simulation of the Crushing Process of a Corrugated Composite Plate," *Composites Part A: Applied Science and Manufacturing*, vol. 42, no. 9, pp. 1119–1126, 2011.

Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling

In this section:

- *About Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling*
- *Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling*
- *Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling*

About Damage and Failure for Fiber-Reinforced Composites Using Multiscale Modeling

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling*
- *Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling*
- *Multiscale Material Modeling*
- **DAMAGE INITIATION*
- **DAMAGE EVOLUTION*
- **ALLOWABLE STRESS*

Overview

With this approach, damage and failure occurs at the constituent level in fiber-reinforced composites, including both unidirectional and woven fiber-reinforced composite materials. These capabilities require you to specify:

- the undamaged response of the fiber and matrix material in the composite, which must be linearly elastic (see *Linear Elastic Behavior* and *Multiscale Material Modeling*);
- the multiscale material definition for the composite, including the constituent material and the microstructure of the composite (see *Multiscale Material Modeling*);
- a damage initiation criterion in the fiber and matrix material (see *Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling*); and
- a damage evolution response in the fiber and matrix material (see *Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling*) when modeling progressive damage.

The Multiscale Approach

Compared to the composite-level damage model (see *About Damage and Failure for Fiber-Reinforced Composites*), the multiscale approach takes into account the constituent material behavior and the microstructure of the composite (see *Mean-Field Homogenization*). The multiscale approach has an advantage over the traditional composite-level approach because it can:

- determine composite properties for any given volume fraction and other geometry specifications before the composite is even manufactured, which facilitates the design process of new composite materials;
- capture damage and failure at a more fundamental scale (the constituent level);
- allow you to specify simpler constitutive models for the constituents;
- allow you to specify simpler and more fundamental damage and failure criteria; and
- capture progressive damage of the composite with simpler evolution laws at the constituent level.

Undamaged Response

Unidirectional fiber-reinforced composite materials exhibit elastic-brittle behavior; damage in these materials is initiated without significant plastic deformation. Consequently, plasticity is often neglected when modeling

the behavior of these materials. You must specify material properties in a user-defined local coordinate system with the local 1-direction aligned with the fiber direction. In addition, you must specify isotropic or transversely isotropic elasticity in the constituent materials. You can also specify isotropic or transversely isotropic thermal expansion in the constituent materials.

The example below shows how to define a multiscale material for a unidirectional fiber-reinforced composite (continuous fiber):

Input file template

```
*MATERIAL, NAME=MatMatrix
…
*MATERIAL, NAME=MatFiber
…
*MATERIAL, NAME=COMPOSITE
*MEAN FIELD HOMOGENIZATION, FORMULATION=MT
*CONSTITUENT, TYPE=MATRIX, MATERIAL=MatMatrix,
NAME=MATRIX_MAT1
*CONSTITUENT, TYPE=INCLUSION, MATERIAL=MatFiber,
NAME=FIBER_MAT2, SHAPE=CYLINDER, DIRECTION=FIXED
v_f, , 1.0,0.0,0.0
```

Abaqus supports woven composite materials that are made of orthogonal yarns (weave and weft) and matrix material. The yarns are unidirectional fiber-reinforced composites and are considered to have a continuous cylindrical inclusion with an elliptical cross-section shape. You must specify a multilevel multiscale material model to specify damage and failure properties in the fiber and matrix constituents of the yarn. The yarn directions must align with the local 1- and 2-directions of the composite material.

The example below shows how to define a multilevel multiscale material for a woven composite:

Input file template

```
*MATERIAL, NAME=MatMatrix
…
*MATERIAL, NAME=YarnMatrix
…
*MATERIAL, NAME=YarnFiber
…
*MATERIAL, NAME=Yarn
*MEAN FIELD HOMOGENIZATION, FORMULATION=MT
*CONSTITUENT, TYPE=MATRIX, MATERIAL=YarnMatrix, NAME=MATRIX_MAT1
*CONSTITUENT, TYPE=INCLUSION, MATERIAL=YarnFiber, NAME=FIBER_MAT2,
 SHAPE=CYLINDER, DIRECTION=FIXED
v_f, , 1.0,0.0,0.0
```

```
*MATERIAL, NAME=COMPOSITE
*MEAN FIELD HOMOGENIZATION, FORMULATION=MT
*CONSTITUENT, TYPE=MATRIX, MATERIAL=MatMatrix, NAME=MATRIX_MAT1
*CONSTITUENT, TYPE=INCLUSION, MATERIAL=Yarn, NAME=FIBER_MAT2,
SHAPE=ELLIPTIC CYLINDER, DIRECTION=FIXED
v_yarn1, ar, 1.0,0.0,0.0
*CONSTITUENT, TYPE=INCLUSION, MATERIAL=Yarn, NAME=FIBER_MAT2,
SHAPE=ELLIPTIC CYLINDER, DIRECTION=FIXED
v_yarn2, ar, 0.0,1.0,0.0
```

You must specify isotropic or transversely isotropic elasticity in the constituent materials. Abaqus/Standard does not support thermal expansion in woven composites when you specify progressive damage.

Damage Initiation Criteria

Abaqus/Standard offers the following damage initiation criteria for constituents inside the unidirectional fiber-reinforced composites and woven composites:

- Constituent-level failure criterion based on the work of *Mayes (2001)*.
- Matrix failure criterion based on the work of *Shultz (2013)*.

These models include initiation criteria for various failure mechanisms commonly observed in fiber-reinforced composites, such as fiber fracture in tension, fiber buckling/kinking under compression, and matrix cracking/crushing under tension/compression, all at the constituent level.

Once a particular damage initiation criterion is satisfied, the material stiffness of the constituent is degraded according to the specified damage evolution law for that criterion. The effective material behavior of the composite can then be determined through homogenization. In the absence of a damage evolution law, the material stiffness is not degraded.

Damage Evolution

The damage evolution law describes the rate of degradation of the material stiffness once the corresponding initiation criterion is reached. At any given time during the analysis, the stress tensor in the material is given by

$$\sigma = \mathbf{C_d} \varepsilon^{el},$$

where $\mathbf{C_d}$ is the damaged elasticity matrix. The evolution of the elasticity matrix arising from damage is discussed in more detail in *Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling*; that section also discusses viscous regularization (*Viscous Regularization*).

Elements

The multiscale damage model can be used with any elements in Abaqus that include mechanical behavior; that is, elements that have displacement degrees of freedom.

References

- Mayes, J. S., and A. C. Hansen, "Multicontinuum Failure Analysis of Composite Structural Laminates," *Mechanics of Composite Materials and Structures*, vol. 8, pp. 249–262, 2001.
- Shultz, J. A., and M. R. Garnich, "Meso-Scale and Multicontinuum Modeling of a Triaxial Braided Textile Composite," *Journal of Composite Materials*, vol. 47, pp. 303–314, 2013.

Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- *Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling*
- **DAMAGE INITIATION*

Overview

The material damage initiation capability for fiber-reinforced materials using multiscale modeling:

- is intended as a general capability for predicting the initiation of damage in unidirectional and woven fiber-reinforced composite materials;
- can predict damage initiation at the constituent level for unidirectional and woven fiber-reinforced composites based on the work of *Mayes (2001)* and *Shultz (2013)*; and
- can be used in combination with damage evolution models.

Damage Initiation

Damage initiation refers to the onset of degradation at a material point. Abaqus/Standard supports the failure criterion by *Mayes (2001)*, who followed the work by *Hashin (1980)*. The general form of Hashin's quadratic failure criterion is

$$F = A_1 I_1 + B_1 I_1^2 + A_2 I_2 + B_2 I_2^2 + C_{12} I_1 I_2 + A_3 I_3 + A_4 I_4,$$

where I_{1-4} are the transversely isotropic stress invariants written as

$$I_1 = \sigma_{11},$$

$$I_2 = \sigma_{22} + \sigma_{33},$$

$$I_3 = \sigma_{22}^2 + \sigma_{33}^2 + 2\sigma_{23}^2, \text{and}$$

$$I_4 = \sigma_{12}^2 + \sigma_{13}^2.$$

Based on the theory of *Mayes (2001)*, Hashin's quadratic criterion can be simplified into the constituent-level criterion by dropping the linear term (accounted for by multiscale modeling) and ignoring the normal stress interaction term $C_{12} I_1 I_2$:

$$F_{f,m} = K_1^{f,m} I_1^2 + K_2^{f,m} I_2^2 + K_3^{f,m} I_3 + K_4^{f,m} I_4.$$

Fiber Failure Criterion

Assuming the transverse failure of unidirectional fiber-reinforced composites is matrix dominated, the coefficients K_2 and K_3 can be set to zero, which reduces the fiber failure criterion to:

Fiber tension ($\widehat{\sigma}_{11}^f \geq 0$):

$$F_f^t = K_1^{f,t} I_1^2 + K_4^f I_4.$$

Fiber compression ($\widehat{\sigma}_{11}^{f} < 0$): $$F_f^c = K_1^{f,c} I_1^2 + K_4^f I_4.$$

$\widehat{\sigma}_{11}^{f}$ is the 11- component of the effective stress tensor, $\widehat{\boldsymbol{\sigma}}$, that is used to evaluate the initiation criteria and is computed from

$$\widehat{\boldsymbol{\sigma}} = \frac{1}{1-d_f}\boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the true stress of the fiber and d_f is the damage variable for the fiber.

The coefficients in the failure criterion can be determined from measured unidirectional composite strengths for pure tension, compression, and in-plane shear loading. In the case of pure tension and compression, $\sigma_{11}^{f} \neq 0, \sigma_{12}^{f} = 0$; therefore, the coefficients $K_1^{f,t}, K_1^{f,c}$ can be determined by

$$K_1^{f,t} = \left(\frac{1}{S_{11}^{f,+}}\right)^2$$

and

$$K_1^{f,c} = \left(\frac{1}{S_{11}^{f,-}}\right)^2,$$

where $S_{11}^{f,+}$ and $S_{11}^{f,-}$ denote the fiber tension and compression stress at failure, respectively. In the case of pure shear, $\sigma_{11}^{f} = 0, \sigma_{12}^{f} \neq 0$; therefore, the coefficients K_4^f can be determined by

$$K_4^f = \alpha\left(\frac{1}{S_{12}^{f}}\right)^2,$$

where S_{12}^{f} denotes the fiber shear stress at failure and α is a coefficient that determines the contribution of the shear stress to the fiber initiation criterion.

The same fiber failure criterion also applies to the fiber material inside the yarn of a woven composite.

Input File Usage: Use the following option to define the fiber damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=HSNFIBER,ALPHA=α

$K_1^{f,t}$,$K_1^{f,c}$,K_4^f

Use the following option to define the allowable stresses for a unidirectional fiber-reinforced composite:

**ALLOWABLE STRESS*,TYPE=UD

Use the following option to define the allowable stresses for a woven composite:

**ALLOWABLE STRESS*,TYPE=WOVEN

Matrix Failure Criterion

Assuming the longitudinal failure of a unidirectional fiber-reinforced composite is fiber dominated, K_1^m can be set to zero, which simplifies the matrix failure criterion to:

Matrix tension ($\widehat{\sigma}_{22}^{m} + \widehat{\sigma}_{33}^{m} \geq 0$): $$F_m^t = K_2^{m,t} I_2^2 + K_3^m I_3 + K_4^m I_4.$$

Matrix compression ($\widehat{\sigma}_{22}^m + \widehat{\sigma}_{33}^m < 0$): $$F_m^c = K_2^{m,c} I_2^2 + K_3^m I_3 + K_4^m I_4.$$

$\widehat{\sigma}_{22}^m, \widehat{\sigma}_{33}^m$ are the components of the effective stress tensor, $\widehat{\boldsymbol{\sigma}}$, that is used to evaluate the initiation criteria and is computed from

$$\widehat{\sigma} = \frac{1}{1-d_m}\boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the true stress of the matrix and d_m is the damage variable for the matrix.

Input File Usage: Use the following option to define the matrix damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=HSNMATRIX

$K_2^{m,t}$,$K_2^{m,c}$,K_3^m,K_4^m

Measured Unidirectional Composite Strength

The coefficients in the matrix failure criterion can be determined with measured unidirectional composite strengths for pure transverse tension/compression, in-plane shear, and transverse shear loading. In the case of pure in-plane shear $\sigma_{22}^m = 0, \sigma_{33}^m = 0, \sigma_{12}^m \neq 0, \sigma_{23}^m = 0$. We can determine the coefficient K_4^m by

$$K_4^m = \left(\frac{1}{S_{12}^m}\right)^2.$$

In the case of pure transverse shear $\sigma_{22}^m = 0, \sigma_{33}^m = 0, \sigma_{23}^m \neq 0, \sigma_{12}^m = 0.$, we can determine the coefficient K_3^m by

$$K_3^m = \frac{1}{2}\left(\frac{1}{S_{23}^m}\right)^2.$$

In the case of pure transverse tension and compression, $\sigma_{22}^m + \sigma_{33}^m \neq 0, \sigma_{12}^m = 0, \sigma_{23}^m = 0.$ The coefficients $K_2^{m,t}, K_2^{m,c}$ can be determined by

$$K_2^{m,t} = \frac{1}{\left(S_{22}^{m,22+}+S_{33}^{m,22+}\right)^2}\left(1 - K_3^m\left[\left(S_{22}^{m,22+}\right)^2 + \left(S_{33}^{m,22+}\right)^2\right]\right)$$

$$K_2^{m,c} = \frac{1}{\left(S_{22}^{m,22-}+S_{33}^{m,22-}\right)^2}\left(1 - K_3^m\left[\left(S_{22}^{m,22-}\right)^2 + \left(S_{33}^{m,22-}\right)^2\right]\right),$$

where $S_{22}^{m,22+}$ and $S_{33}^{m,22+}$ denote the matrix stress components at transverse tension failure, and $S_{22}^{m,22-}$ and $S_{33}^{m,22-}$ denote the matrix stress components at transverse compression failure.

In the case of woven composites, measured composite strength is usually limited to pure tension/compression in the fiber direction and in-plane shear. *Schultz (2013)* proposed a principal stress-based damage criterion for the matrix material:

Matrix tension ($\widehat{\sigma}_{22}^m + \widehat{\sigma}_{33}^m \geq 0$): $$F_m^t = A_2^{m,t} I_t^2 + K_4^m I_4.$$

Matrix compression ($\widehat{\sigma}_{22}^m + \widehat{\sigma}_{33}^m < 0$): $$F_m^c = A_2^{m,c} I_c^2 + K_4^m I_4.$$

$I_{t,c}$ are the maximum/minimum principal stress of the matrix in the plane normal to the fiber direction:

$$I_{t,c} = \frac{\widehat{\sigma}_{22}^m+\widehat{\sigma}_{33}^m}{2} \pm \left[\left(\frac{\widehat{\sigma}_{22}^m-\widehat{\sigma}_{33}^m}{2}\right)^2 + \left(\widehat{\sigma}_{23}^m\right)^2\right]^{\frac{1}{2}}.$$

$\widehat{\sigma}_{22}^m, \widehat{\sigma}_{33}^m$ are the components of the effective stress tensor, $\widehat{\boldsymbol{\sigma}}$, that is used to evaluate the initiation criteria and is computed from

$$\widehat{\boldsymbol{\sigma}} = \frac{1}{1-d_m}\boldsymbol{\sigma},$$

where $\boldsymbol{\sigma}$ is the true stress of the matrix and d_m is the damage variable for the matrix.

The coefficient K_4^m can be determined by

$$K_4^m = \left(\frac{1}{S_{12}^m}\right)^2.$$

The coefficients $A_2^{m,tc}$ can be determined using composite strengths for pure tension and compression loading conditions ($\sigma_{22}^m + \sigma_{33}^m \neq 0, \sigma_{12}^m = 0, \sigma_{23}^m = 0$).

Input File Usage: Use the following option to define the matrix damage initiation criterion proposed by *Schultz (2013)*:

**DAMAGE INITIATION*,CRITERION=TSINVMATRIX

$A_2^{m,t}, A_2^{m,c}, K_4^m$

Allowable Stress

You can choose not to specify the coefficients directly in the definition of the constituent damage criterion. In that case you need to specify the allowable stress of the composite in the multiscale material definition. Then Abaqus/Standard computes the volume-averaged constituent stresses using the homogenization method specified, and the failure criterion coefficients using the equations above. When you specify the allowable stresses at the composite level, Abaqus/Standard ignores failure coefficients specified at the constituent level.

In a woven composite matrix, failure can occur before the ultimate failure of the composite. In addition to the ultimate composite strengths, you must specify the strengths of the composite at the initial failure of the matrix inside the tows.

Input File Usage: Use the following option to define the allowable stresses for a unidirectional fiber-reinforced composite:

**ALLOWABLE STRESS*,TYPE=UD

$X^T, X^C Y^T Y^C S^L S^T$

Use the following option to define the allowable stresses for a woven composite (X^T, X^C, S^L) including the stresses at the onset of matrix failure ($X_0^T X_0^C S_0^L$):

**ALLOWABLE STRESS*,TYPE=WOVEN

$X^T, X^C, S^L, X_0^T X_0^C S_0^L$

Elements

The multiscale damage model can be used with any elements in Abaqus/Standard that include mechanical behavior; that is, elements that have displacement degrees of freedom.

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables relate specifically to damage initiation at a material point in the fiber-reinforced composite damage model:

DMICRT All damage initiation criteria components.

HSNFTCRT Maximum value of the fiber tensile initiation criterion experienced during the analysis when the HSNFIBER criterion is specified.

HSNFCCRT Maximum value of the fiber compressive initiation criterion experienced during the analysis when the HSNFIBER criterion is specified.

HSNMTCRT Maximum value of the matrix tensile initiation criterion experienced during the analysis when the HSNMATRIX criterion is specified.

HSNMCCRT Maximum value of the matrix compressive initiation criterion experienced during the analysis when the HSNMATRIX criterion is specified.

TSINVMTCRT Maximum value of the matrix tensile initiation criterion experienced during the analysis when the TSINVMATRIX criterion is specified.

TSINVMCCRT Maximum value of the matrix compressive initiation criterion experienced during the analysis when the TSINVMATRIX criterion is specified.

For the variables above associated with a damage initiation criterion, a value that is less than 1.0 indicates that the criterion has not been satisfied, while a value of 1.0 or higher indicates that the criterion has been satisfied. If you define a damage evolution model, the maximum value of this variable does not exceed 1.0; if you do not define a damage evolution model, this variable can have values higher than 1.0, which indicates the degree to which the criterion has been exceeded.

References

- Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, vol. 47, pp. 329–334, 1980.
- Mayes, J. S., and A. C. Hansen, "Multicontinuum Failure Analysis of Composite Structural Laminates," *Mechanics of Composite Materials and Structures*, vol. 8, pp. 249–262, 2001.
- Shultz, J. A., and M. R. Garnich, "Meso-scale and Multicontinuum Modeling of a Triaxial Braided Textile Composite," *Journal of Composite Materials*, vol. 47, pp. 303–314, 2013.

Damage Evolution for Fiber-Reinforced Composites Using Multiscale Modeling

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling*
- **DAMAGE EVOLUTION*
- **DAMAGE STABILIZATION*

Overview

The damage evolution capability for multiscale fiber-reinforced materials in Abaqus/Standard:

- assumes that damage is characterized by progressive degradation of the material stiffness, leading to material failure;
- models the degradation of the composite at the constituent level, leading to composite-level failure through homogenization;
- requires the undamaged response of the material to be linear elastic (see *Linear Elastic Behavior*);
- offers options to model energy dissipation in each constituent during the damage process in a unidirectional fiber-reinforced composite;
- offers options to model instantaneous degradation of material stiffness with user-specified scaling factors (this method is the only option for modeling the progressive damage of a woven composite); and
- can be used in combination with a viscous regularization scheme to improve the convergence rate in the softening regime.

Damage Evolution

Damage Initiation for Fiber-Reinforced Composites Using Multiscale Modeling discussed the constituent-level damage initiation in fiber-reinforced composites. This section discusses the post-damage initiation behavior for cases in which a damage evolution model has been specified. The damage evolution is modeled at the constituent level, leading to the damage process of a composite through homogenization. Prior to damage initiation, the constituent material is linearly elastic, with the stiffness matrix of an isotropic or a transversely isotropic material. Thereafter, the response of the constituent material is computed from

$$\boldsymbol{\sigma} = \mathbf{C}_\mathbf{d}\boldsymbol{\varepsilon}^{el},$$

where $\boldsymbol{\varepsilon}^{el}$ is the elastic strain, and $\mathbf{C}_\mathbf{d}$ is the damaged elasticity matrix, which has the form

$$\mathbf{C}_\mathbf{d} = (1 - D)\,\mathbf{C},$$

where D is the damage variable and $\mathbf{C}$ is the undamaged stiffness matrix of the constituent.

The damage variables, D, are derived from damage variables in each constituent based on tension or compression modes, as follows:

Fiber:

$$D = \begin{cases} d_f^t \text{ if } \quad \hat{\sigma}_{11} \geq 0, \\ d_f^c \text{ if } \quad \hat{\sigma}_{11} < 0. \end{cases}$$

Matrix:

$$D = \begin{cases} d_m^t \text{ if } \quad \widehat{\sigma}_{22} + \widehat{\sigma}_{33} \geq 0, \\ d_m^c \text{ if } \quad \widehat{\sigma}_{22} + \widehat{\sigma}_{33} < 0. \end{cases}$$

$\widehat{\sigma}_{11}$, $\widehat{\sigma}_{22}$, and $\widehat{\sigma}_{33}$ are components of the effective stress tensor at the constituent level.

Evolution of Damage Variables for Each Mode

To alleviate mesh dependency during material softening, Abaqus introduces a characteristic length into the formulation so that the constitutive law is expressed in a stress-displacement relation. The damage variable evolves such that the stress-displacement behaves as shown in *Figure 1* in each of the two failure modes (tension and compression). The positive slope of the stress-displacement curve before damage initiation corresponds to linear elastic material behavior; the negative slope after damage initiation is achieved by evolution of the respective damage variables according to the equations shown below.

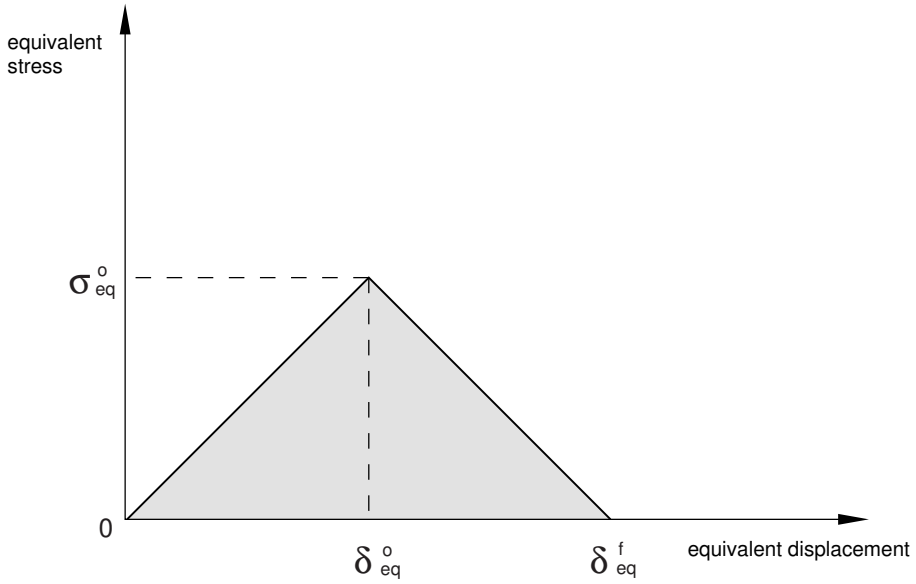


Figure 1: Equivalent stress versus equivalent displacement.

The equivalent displacement and stress for the tension and compression damage modes are defined as follows:

Tension $(\widehat{\sigma}_{11} \geq 0)$ with the HSNFIBER damage criterion:

$$\delta_{eq}^{ft} = L^c \sqrt{\langle \varepsilon_{11} \rangle^2 + \alpha \left(\varepsilon_{12}^2 + \varepsilon_{13}^2 \right)},$$

$$\sigma_{eq}^{ft} = \frac{\langle \sigma_{11} \rangle \langle \varepsilon_{11} \rangle + \alpha (\tau_{12}\varepsilon_{12} + \tau_{13}\varepsilon_{13})}{\delta_{eq}^{ft}/L^c},$$

Compression $(\widehat{\sigma}_{11} < 0)$ with the HSNFIBER damage criterion:

$$\delta_{eq}^{fc} = L^c \sqrt{\langle -\varepsilon_{11} \rangle^2 + \alpha \left(\varepsilon_{12}^2 + \varepsilon_{13}^2 \right)},$$

$$\sigma_{eq}^{fc} = \frac{\langle -\sigma_{11} \rangle \langle -\varepsilon_{11} \rangle + \alpha (\tau_{12}\varepsilon_{12} + \tau_{13}\varepsilon_{13})}{\delta_{eq}^{fc}/L^c},$$

Tension $(\widehat{\sigma}_{22} + \widehat{\sigma}_{33} \geq 0)$ with the HSNMATRIX or TSINVMATRIX damage criterion:

$$\delta_{eq}^{mt} = L^c \sqrt{\langle I_{t,\varepsilon} \rangle^2 + \varepsilon_{12}^2 + \varepsilon_{13}^2},$$

$$\sigma_{eq}^{mt} = \frac{\langle I_t \rangle \langle I_{t,\varepsilon} \rangle + \tau_{12}\varepsilon_{12} + \tau_{13}\varepsilon_{13}}{\delta_{eq}^{mt}/L^c}.$$

where $I_t = \frac{\hat{\sigma}_{22}+\hat{\sigma}_{33}}{2} + \left[\left(\frac{\hat{\sigma}_{22}-\hat{\sigma}_{33}}{2}\right)^2 + (\hat{\sigma}_{23})^2\right]^{\frac{1}{2}}$ and $I_{t,\varepsilon} = \frac{\hat{\varepsilon}_{22}+\hat{\varepsilon}_{33}}{2} + \left[\left(\frac{\hat{\varepsilon}_{22}-\hat{\varepsilon}_{33}}{2}\right)^2 + (\hat{\varepsilon}_{23})^2\right]^{\frac{1}{2}}$ are the maximum principal stress/strain in the plane normal to the fiber direction.

Compression $(\hat{\sigma}_{22} + \hat{\sigma}_{33} < 0)$ with the HSNMATRIX or TSINVMATRIX damage criterion:

$$\delta_{eq}^{mc} = L^c \sqrt{\langle I_{c,\varepsilon} \rangle^2 + \varepsilon_{12}^2 + \varepsilon_{13}^2},$$

$$\sigma_{eq}^{mc} = \frac{\langle I_c \rangle \langle I_{c,\varepsilon} \rangle + \tau_{12}\varepsilon_{12} + \tau_{13}\varepsilon_{13}}{\delta_{eq}^{mc}/L^c}.$$

where $I_c = \frac{\hat{\sigma}_{22}+\hat{\sigma}_{33}}{2} - \left[\left(\frac{\hat{\sigma}_{22}-\hat{\sigma}_{33}}{2}\right)^2 + (\hat{\sigma}_{23})^2\right]^{\frac{1}{2}}$ and $I_{c,\varepsilon} = \frac{\hat{\varepsilon}_{22}+\hat{\varepsilon}_{33}}{2} - \left[\left(\frac{\hat{\varepsilon}_{22}-\hat{\varepsilon}_{33}}{2}\right)^2 + (\hat{\varepsilon}_{23})^2\right]^{\frac{1}{2}}$ are the minimum principal stress/strain in the plane normal to the fiber direction.

The characteristic length, L^c, is based on the element geometry and formulation. L^c is a typical length of a line across an element for a first-order element and half of the same typical length for a second-order element. For membranes and shells, L^c is a characteristic length in the reference surface, computed as the square root of the area. The symbol $\langle\rangle$ in the equations above represents the Macaulay bracket operator, which is defined for every $\alpha \in \Re$ as $\langle\alpha\rangle = (\alpha + |\alpha|)/2$.

After damage initiation (that is, $\delta_{eq} \geq \delta_{eq}^0$) for the behavior shown in *Figure 1*, the damage variable for a particular mode is given by the following expression:

$$d = \frac{\delta_{eq}^f \left(\delta_{eq} - \delta_{eq}^0\right)}{\delta_{eq} \left(\delta_{eq}^f - \delta_{eq}^0\right)},$$

where δ_{eq}^0 is the initial equivalent displacement at which the initiation criterion for that mode was met, and δ_{eq}^f is the displacement at which the material is completely damaged in this failure mode. The relation above is presented graphically in *Figure 2*.

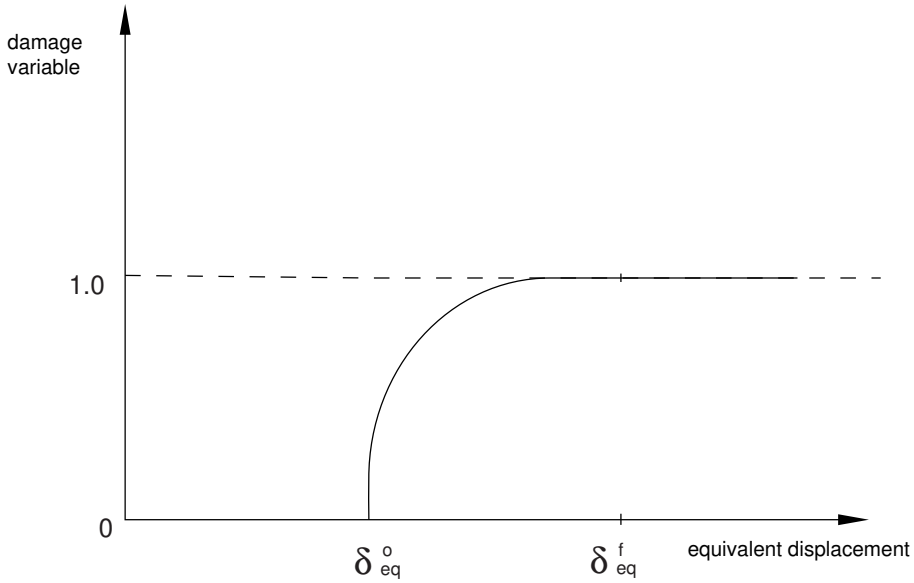


Figure 2: Damage variable as a function of equivalent displacement.

The values of δ_{eq}^{0} for the various modes depend on the elastic stiffness and the strength parameters specified as part of the damage initiation definition (see *Damage Initiation for Fiber-Reinforced Composites*). For each failure mode, you must specify the energy dissipated by failure, G^c, which corresponds to the area of the triangle OAC in *Figure 3*.

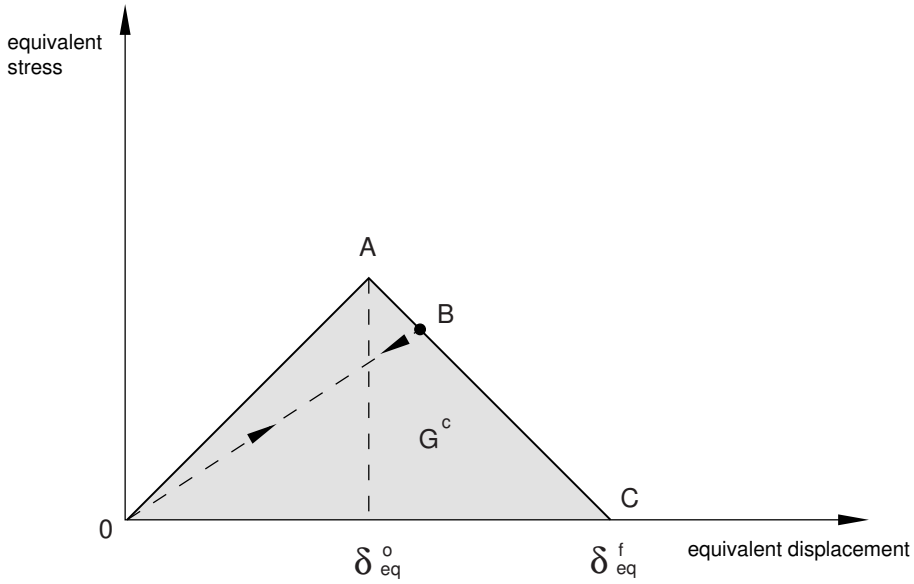


Figure 3: Linear damage evolution.

The values of δ_{eq}^{f} for the various modes depend on the respective G^c values.

Unloading from a partially damaged state, such as point B in *Figure 3*, occurs along a linear path toward the origin in the plot of equivalent stress versus equivalent displacement. This same path is followed back to point B on reloading as shown in the figure.

Input File Usage: Use the following option to define the damage evolution law:

*DAMAGE EVOLUTION, TYPE=ENERGY, SOFTENING=LINEAR

G_t^c, G_c^c

where G_t^c, G_c^c are energies dissipated during damage for tension and compression failure modes, respectively.

You can specify the post-damage residual stiffness scaling factor directly for the tension and compression failure modes. The scaling factor is related to the damage variable as

$$SF = 1 - D.$$

When a small scaling factor is specified at the onset of damage initiation, the sudden stiffness degradation often leads to convergence difficulties in implicit analyses. Discontinuous solutions also require a larger number of iterations to reach convergence (see *Commonly Used Control Parameters* for information about how to modify solution control parameters to achieve better convergence). You can also use the viscous regularization scheme described below (*Viscous Regularization*) to overcome these difficulties.

Input File Usage: Use the following option to define the post-damage residual stiffness scaling factor directly for damage evolution:

*DAMAGE EVOLUTION, TYPE=DISCONTINUOUS

SF_t, SF_c

where SF_t and SF_c are the post-damage scaling factor for tension and compression failure modes, respectively.

Maximum Degradation

When a fracture energy is specified for damage evolution, Abaqus/Standard limits the damage variable at a constituent to a maximum value of 0.99 in both tension and failure modes. You can modify this upper bound as discussed in *Controlling Element Deletion and Maximum Degradation for Materials with Damage Evolution*.

Abaqus/Standard does not delete elements after failure with this multiscale damage model.

Viscous Regularization

Material models exhibiting softening behavior and stiffness degradation often lead to severe convergence difficulties in implicit analysis programs such as Abaqus/Standard. You can overcome some of these convergence difficulties using the viscous regularization scheme, which slows down the rate of increase of damage.

In this regularization scheme, a viscous damage variable is defined by the evolution equation:

$$\dot{d}_v = \frac{1}{\eta}\left(d - d_v\right),$$

where η is the viscosity coefficient representing the relaxation time of the viscous system, and d is the damage variable evaluated in the inviscid backbone model. The damaged response of the viscous material is given as

$$\sigma = \mathbf{C_d}\epsilon,$$

where the damaged elasticity matrix, $\mathbf{C_d}$, is computed using viscous values of damage variables. Using viscous regularization with a small value of the viscosity parameter (small compared to the characteristic time increment) usually helps improve the rate of convergence of the model in the softening regime without compromising results.

The basic idea is that the solution of the viscous system relaxes to that of the inviscid case as $t/\eta \to \infty$, where t represents time.

In Abaqus/Standard, the approximate amount of energy associated with viscous regularization over the whole model or over an element set is available using the output variable ALLCD.

Defining Viscous Regularization Coefficients

You can specify different values of viscous coefficients for different failure modes.

Input File Usage: Use the following option to define viscous coefficients:

**DAMAGE STABILIZATION*

η_t, η_c

where η_t and η_c are viscosity coefficients for tension and compression failure modes, respectively.

Elements

The multiscale damage model can be used with any elements in Abaqus/Standard that include mechanical behavior; that is, elements that have displacement degrees of freedom.

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables relate specifically to damage evolution in the multiscale fiber-reinforced composite damage model (the integration point output variables are available only when you request micro-level output in your output request):

DAMAGEFT Fiber tensile damage variable when the HSNFIBER damage criterion is specified.

DAMAGEFC Fiber compressive damage variable when the HSNFIBER damage criterion is specified.

DAMAGEMT Matrix tensile damage variable when the HSNMATRIX or TSINVMATRIX damage criterion is specified.

DAMAGEMC Matrix compressive damage variable when the HSNMATRIX or TSINVMATRIX damage criterion is specified.

EDMDDEN Energy dissipated per unit volume in the element by damage.

ELDMD Total energy dissipated in the element by damage.

DMENER Energy dissipated per unit volume by damage.

ALLDMD Energy dissipated in the whole (or partial) model by damage.

ECDDEN Energy per unit volume in the element that is associated with viscous regularization.

ELCD

Total energy in the element that is associated with viscous regularization.

CENER

Energy per unit volume that is associated with viscous regularization.

ALLCD

The approximate amount of energy over the whole model or over an element set that is associated with viscous regularization.

Generalized Damage and Failure for Anisotropic Materials

In this section:

- *About Generalized Stress- and Strain-Based Damage and Failure for Anisotropic Materials*
- *Generalized Damage Initiation Criteria for Anisotropic Materials*

About Generalized Stress- and Strain-Based Damage and Failure for Anisotropic Materials

Products: Abaqus/Standard Abaqus/Explicit

References:

- *About Progressive Damage and Failure*
- **DAMAGE INITIATION*

Overview

Damage evolution laws are not supported for these general stress- and strain-based damage initiation criteria. Therefore, Abaqus evaluates the initiation criterion for output purposes only, and the definition has no effect on the material response. You can specify more than one damage initiation criterion for a given material. If you specify multiple damage initiation criteria for the same material, the solver treats them independently. The output is in the form of a failure index that provides information about the proximity to failure.

The generalized stress- and strain-based damage initiation criteria provide failure measures that:

- are indicators of material failure;
- are postprocessed output requests and do not cause any material degradation;
- can be used with all stress/displacement elements; and
- can be strain rate–dependent in Abaqus/Explicit.

Damage Initiation Criterion

Abaqus provides the following damage initiation criteria based on different failure theories that can be used as indicators of material failure:

- Maximum stress theory and maximum strain theory.
- Stress- and strain-based Tsai-Wu theory (*Tsai (1971)*).

For each criterion, stresses and strains are measured in the local material coordinate system.

Elements

The general stress- and strain-based damage initiation criteria can be used with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

References

- Tsai, S. W., and E. M. Wu, "A General Theory of Strength for Anisotropic Materials," *Journal of Composite Materials*, vol. 5(1), pp. 58–80, 1971.

Generalized Damage Initiation Criteria for Anisotropic Materials

Products: Abaqus/Standard Abaqus/Explicit

References:

- *About Progressive Damage and Failure*
- **DAMAGE INITIATION*

Overview

Maximum Strain Criterion

The input data for the maximum strain failure theory are:

- tensile and compressive strain limits, $X_{\epsilon t}$ and $X_{\epsilon c}$, in the local X-direction;
- tensile and compressive strain limits, $Y_{\epsilon t}$ and $Y_{\epsilon c}$, in the local Y-direction;
- tensile and compressive strain limits, $Z_{\epsilon t}$ and $Z_{\epsilon c}$, in the local Z-direction;
- shear strain limits (maximum shear strains), $S_{\epsilon_{xy}}$, $S_{\epsilon_{xz}}$ and $S_{\epsilon_{yz}}$, in the local X–Y, X–Z, and Y–Z planes, respectively

All strain limits must be positive.

If $\epsilon_{11} > 0$, $X_\epsilon = X_{\epsilon_t}$; otherwise, $X_\epsilon = X_{\epsilon_c}$. If $\epsilon_{22} > 0$, $Y_\epsilon = Y_{\epsilon_t}$; otherwise, $Y_\epsilon = Y_{\epsilon_c}$. If $\epsilon_{33} > 0$, $Z_\epsilon = Z_{\epsilon_t}$; otherwise, $Z_\epsilon = Z_{\epsilon_c}$.

The maximum strain-based damage initiation criterion is met when the following condition is satisfied for a three-dimensional strain state, $\boldsymbol{\epsilon}$:

$$I_F = \max\left(\left\lfloor \frac{\epsilon_{11}}{X_\epsilon} \right\rfloor, \left\lfloor \frac{\epsilon_{22}}{Y_\epsilon} \right\rfloor, \left\lfloor \frac{\epsilon_{33}}{Z_\epsilon} \right\rfloor, \left\lfloor \frac{\epsilon_{12}}{S_{\epsilon_{xy}}} \right\rfloor, \left| \frac{\epsilon_{13}}{S_{\epsilon_{xz}}} \right|, \left| \frac{\epsilon_{23}}{S_{\epsilon_{yz}}} \right|\right) = 1.0\,.$$

Input File Usage: Use the following option to define the maximum strain-based damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=MAX STRAIN

Maximum Stress Criterion

The input data for the maximum stress failure theory are:

- tensile and compressive stress limits, X_t and X_c in the local X-direction;
- tensile and compressive stress limits, Y_t and Y_c in the local Y-direction;
- tensile and compressive stress limits, Z_t and Z_c in the local Z-direction;
- shear strengths (maximum shear stresses), S_{xy}, S_{xz} and S_{yz} in the local X–Y, X–Z, and Y–Z planes, respectively.

All stress limits must be positive.

If $\sigma_{11} > 0, X = X_t$; otherwise, $X = X_c$. If $\sigma_{22} > 0, Y = Y_t$; otherwise, $Y = Y_c$. If $\sigma_{33} > 0, Z = Z_t$; otherwise, $Z = Z_c$

The maximum stress-based damage initiation criterion is met when the following condition is satisfied for a three-dimensional stress state, $\boldsymbol{\sigma}$:

$$I_F = \max\left(\left\lfloor \frac{\sigma_{11}}{X} \right\rfloor, \left\lfloor \frac{\sigma_{22}}{Y} \right\rfloor, \left\lfloor \frac{\sigma_{33}}{Z} \right\rfloor, \left\lfloor \frac{\sigma_{12}}{S_{xy}} \right\rfloor, \left| \frac{\sigma_{13}}{S_{xz}} \right|, \left| \frac{\sigma_{23}}{S_{yz}} \right|\right) = 1.0\,.$$

Input File Usage: Use the following option to define the maximum stress-based damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=MAX STRESS

Strain-Based Tsai-Wu Criterion

The strain-based Tsai-Wu damage initiation criterion is based on the theory of *Tsai (1971)*. The strain-based Tsai-Wu damage initiation criterion is met when the following condition is satisfied for a three-dimensional strain state $\boldsymbol{\epsilon}$:

$$I_F = G_1\epsilon_{11} + G_2\epsilon_{22} + G_3\epsilon_{33} + G_{11}\epsilon_{11}^2 + G_{22}\epsilon_{22}^2 + G_{33}\epsilon_{33}^2 + 2G_{12}\epsilon_{11}\epsilon_{22} + 2G_{13}\epsilon_{11}\epsilon_{33} + 2G_{23}\epsilon_{22}\epsilon_{33} + G_{44}\epsilon_{12}^2 + G_{55}\epsilon_{13}^2 + G$$

where $G_1, G_2, G_3, G_{11}, G_{22}, G_{33}, G_{12}, G_{13}, G_{23}, G_{44}, G_{55}, \text{and } G_{66}$ are material properties related to material maximum strain limits.

Input File Usage: Use the following option to define the Tsai-Wu strain-based damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=TSAIWUE

Stress-Based Tsai-Wu Criterion

The stress-based Tsai-Wu damage initiation criterion is based on the theory of *Tsai (1971)*. The stress-based Tsai-Wu damage initiation criterion is met when the following condition is satisfied for a three-dimensional stress state $\boldsymbol{\sigma}$:

$$I_F = F_1\sigma_{11} + F_2\sigma_{22} + F_3\sigma_{33} + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{33}\sigma_{33}^2 + 2F_{12}\sigma_{11}\sigma_{22} + 2F_{13}\sigma_{11}\sigma_{33} + 2F_{23}\sigma_{22}\sigma_{33} + F_{44}\sigma_{12}^2 + F_{55}\sigma_{13}^2 +$$

where $F_1, F_2, F_3, F_{11}, F_{22}, F_{33}, F_{12}, F_{13}, F_{23}, F_{44}, F_{55}, \text{and } F_{66}$ are material properties related to material stress strengths.

Input File Usage: Use the following option to define the Tsai-Wu stress-based damage initiation criterion:

**DAMAGE INITIATION*,CRITERION=TSAIWU

Strain Rate Dependency in Abaqus/Explicit

In Abaqus/Explicit, each damage initiation criterion can depend either on the equivalent strain rate, $\dot{\bar{\epsilon}}$, or on the strain rate components, $\dot{\epsilon}_{ij}$. The equivalent strain rate is defined as

$$\dot{\bar{\epsilon}} = \frac{\{\Delta\bar{\epsilon}\}}{\Delta t}, \bar{\epsilon} = \sqrt{\frac{2}{3}\boldsymbol{\epsilon} : \boldsymbol{\epsilon}},$$

and the strain rate components are defined as

$$\dot{\epsilon}_{ij} = \frac{\langle \Delta | \epsilon_{ij} | \rangle}{\Delta t}.$$

In the above definitions <> represents the Macaulay bracket operator, and $|\epsilon_{ij}|$ represents the absolute value of ϵ_{ij}. These definitions ensure that strain rate dependency is taken into consideration during monotonic loading only (increasing equivalent strain or absolute value of strain components).

Input File Usage: Use the following option to define strain rate-dependent damage initiation criterion:

**DAMAGE INITIATION*,RATE DEPENDENT

Output Failure Indices

Each damage initiation criterion defines a failure surface surrounding the origin. Failure occurs whenever a state of stress or strain is either on or outside this surface. The failure index, R, is used to measure the proximity to the failure surface. R is defined as the (dimensionless) scaling factor such that, for a given stress state σ or a given strain state ϵ,

$$I_F\left(\frac{\sigma}{R}\right) = 1, \text{or} \quad I_F\left(\frac{\epsilon}{R}\right) = 1;$$

that is, $\frac{1}{R}$ is the scaling factor by which we need to multiply all the stress or strain components simultaneously to lie on the failure surface. Values $R < 1.0$ indicate that the state of stress is within the failure surface, while values $R > 1.0$ indicate failure.

Elements

You can use the general stress- and strain-based damage initiation criteria with any elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*), the following variables relate specifically to damage initiation at a material point in the general stress/strain-based damage model:

DMICRT All damage initiation criteria components.

EREQ Strain rate used to evaluate strain-rate-dependent damage initiation criteria in Abaqus/Explicit.

MSTRAINCRT Maximum value of the maximum strain damage initiation criterion experienced during the analysis.

MSTRESSCRT Maximum value of the maximum stress damage initiation criterion experienced during the analysis.

TSAIWUCRT Maximum value of the stress-based Tsai-Wu damage initiation criterion experienced during the analysis.

TSAIWUECRT	Maximum value of the strain-based Tsai-Wu damage initiation criterion experienced during the analysis.
DMIFI	All failure index components.
MSTRAINFI	Maximum value of the maximum strain failure index experienced during the analysis.
MSTRESSFI	Maximum value of the maximum stress failure index experienced during the analysis.
TSAIWUFI	Maximum value of the stress-based Tsai-Wu failure index experienced during the analysis.
TSAIWUEFI	Maximum value of the strain-based Tsai-Wu failure index experienced during the analysis.

References

- Tsai, S. W., and E. M. Wu, "A General Theory of Strength for Anisotropic Materials," *Journal of Composite Materials*, vol. 5(1), pp. 58–80, 1971.

Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis

In this section:

- *About Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis*
- *Damage Initiation for Ductile Materials in Low-Cycle Fatigue*
- *Damage Evolution for Ductile Materials in Low-Cycle Fatigue*

About Damage and Failure for Ductile Materials in Low-Cycle Fatigue Analysis

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- *Damage Initiation for Ductile Materials in Low-Cycle Fatigue*
- *Damage Evolution for Ductile Materials in Low-Cycle Fatigue*
- *Low-Cycle Fatigue Analysis Using the Direct Cyclic Approach*
- **DAMAGE INITIATION*
- **DAMAGE EVOLUTION*

Overview

Abaqus/Standard offers a general capability for modeling progressive damage and failure of ductile materials due to stress reversals and the accumulation of inelastic strain energy in a low-cycle fatigue analysis using the direct cyclic approach. In the most general case this requires the specification of the following:

- the undamaged ductile materials in any elements (including cohesive elements based on a continuum approach) whose response is defined in terms of a continuum-based constitutive model (*About the Material Library*);
- a damage initiation criterion (*Damage Initiation for Ductile Materials in Low-Cycle Fatigue*); and
- a damage evolution response (*Damage Evolution for Ductile Materials in Low-Cycle Fatigue*).

A summary of the general framework for progressive damage and failure in Abaqus is given in *About Progressive Damage and Failure*. This section provides an overview of the damage initiation criteria and damage evolution law for ductile materials in a low-cycle fatigue analysis using the direct cyclic approach.

General Concepts of Damage of Ductile Materials in Low-Cycle Fatigue

Accurately and effectively predicting the fatigue life for an inelastic structure, such as a solder joint in an electronic chip packaging, subjected to sub-critical cyclic loading is a challenging problem. Cyclic thermal or mechanical loading often leads to stress reversals and the accumulation of inelastic strain, which may in turn lead to the initiation and propagation of a crack. The low-cycle fatigue analysis capability in Abaqus/Standard uses a direct cyclic approach (*Low-Cycle Fatigue Analysis Using the Direct Cyclic Approach*) to model progressive damage and failure based on a continuum damage approach. The damage initiation (*Damage Initiation for Ductile Materials in Low-Cycle Fatigue*) and evolution (*Damage Evolution for Ductile Materials in Low-Cycle Fatigue*) are characterized by the stabilized accumulated inelastic hysteresis strain energy per cycle proposed by *Darveaux (2002)* and *Lau (2002)*.

The damage evolution law describes the rate of degradation of the material stiffness per cycle once the corresponding initiation criterion has been reached. For damage in ductile materials Abaqus/Standard assumes that the degradation of the stiffness can be modeled using a scalar damage variable, D. At any given cycle during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\,\overline{\boldsymbol{\sigma}},$$

where $\overline{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor that would exist in the material in the absence of damage computed in the current increment. The material has lost its load-carrying capacity when $D = 1$.

Elements

The failure modeling capability for ductile materials can be used with any elements (including cohesive elements based on a continuum approach) in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom).

References

- Darveaux, R., "Effect of Simulation Methodology on Solder Joint Crack Growth Correlation and Fatigue Life Prediction," *Journal of Electronic Packaging*, vol. 124, pp. 147–154, 2002.
- Lau, J., S. Pan, and C. Chang, "A New Thermal-Fatigue Life Prediction Model for Wafer Level Chip Scale Package (WLCSP) Solder Joints," *Journal of Electronic Packaging*, vol. 124, pp. 212–220, 2002.

Damage Initiation for Ductile Materials in Low-Cycle Fatigue

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- **DAMAGE INITIATION*

Overview

The material damage initiation capability for ductile materials based on inelastic hysteresis energy:

- is intended as a general capability for predicting initiation of damage in ductile materials in a low-cycle fatigue analysis;
- can be used in combination with the damage evolution law for ductile materials described in *Damage Evolution for Ductile Materials in Low-Cycle Fatigue*; and
- can be used only in a low-cycle fatigue analysis using the direct cyclic approach (*Low-Cycle Fatigue Analysis Using the Direct Cyclic Approach*).

Damage Initiation Criteria for Ductile Materials

The damage initiation criterion is a phenomenological model for predicting the onset of damage due to stress reversals and the accumulation of inelastic strain in a low-cycle fatigue analysis. It is characterized by the accumulated inelastic hysteresis energy density per cycle, Δw, in a material point for a stabilized cycle. The cycle number in which damage is initiated is given by

$$N_0 = c_1(\Delta w/\Delta w_0)^{c_2},$$

where c_1 and c_2 are material constants, and Δw_0 is a reference value of the accumulated inelastic hysteresis energy density per cycle.

The use of the reference energy density, Δw_0, makes the above form of the damage initiation law different from the more standard Coffin-Manson type relations. It is recommended that for Δw_0 (which has units of energy density), you choose a value of 1.0 in the unit system in which the initiation law is calibrated to the available test data. If a different unit system is then used, you must convert the material constant c_1 based on its dimension of cycle. In this case, you must also convert Δw_0 appropriately to the new system to ensure that the physical results are invariant to the choice of the unit system.

The initiation criterion can be used in conjunction with any ductile material.

Input File Usage: **DAMAGE INITIATION*, CRITERION=HYSTERESIS ENERGY

Use the following option to specify a reference accumulated inelastic hysteresis energy density per cycle:

**DAMAGE INITIATION*, CRITERION=HYSTERESIS ENERGY, REF ENERGY=Δw_0

Elements

The damage initiation criteria for ductile materials can be used with any elements in Abaqus/Standard that include mechanical behavior (elements that have displacement degrees of freedom). This includes cohesive elements based on a continuum approach (*Modeling of an Adhesive Layer of Finite Thickness*).

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variable has special meaning when a damage initiation criterion is specified:

CYCLEINI Number of cycles to initialize the damage at the material point.

Damage Evolution for Ductile Materials in Low-Cycle Fatigue

Products: Abaqus/Standard

References:

- *About Progressive Damage and Failure*
- **DAMAGE EVOLUTION*

Overview

The damage evolution capability for ductile materials based on inelastic hysteresis energy:

- assumes that damage is characterized by the progressive degradation of the material stiffness, leading to material failure;
- must be used in combination with a damage initiation criterion for ductile materials in low-cycle fatigue analysis (*Damage Initiation for Ductile Materials in Low-Cycle Fatigue*);
- uses the inelastic hysteresis energy per stabilized cycle to drive the evolution of damage after damage initiation; and
- must be used in conjunction with the linear elastic material model (*Linear Elastic Behavior*), the porous elastic material model (*Elastic Behavior of Porous Materials*), or the hypoelastic material model (*Hypoelastic Behavior*).

Damage Evolution Based on Accumulated Inelastic Hysteresis Energy Density

Once the damage initiation criterion (*Damage Initiation for Ductile Materials in Low-Cycle Fatigue*) is satisfied at a material point, the damage state is calculated and updated based on the accumulated inelastic hysteresis energy density per cycle, Δw, for a stabilized cycle. The rate of the damage in a material point per cycle is given by

$$\frac{dD}{dN} = \frac{c_3(\Delta w/\Delta \overline{w}_0)^{c_4}}{L},$$

where c_3 and c_4 are material constants, L is the characteristic length associated with an integration point, and $\Delta \overline{w}_0$ is a reference value of the accumulated inelastic hysteresis energy density per cycle.

The use of the reference energy density, $\Delta \overline{w}_0$, makes the above form of the damage evolution law different from the more standard Coffin-Manson type relations. It is recommended that for $\Delta \overline{w}_0$ (which has units of energy density), you choose a value of 1.0 in the unit system in which the evolution law is calibrated to the available test data. If a different unit system is then used, you must convert the material constant c_3 based on its dimension of length per cycle. In this case, you must also convert $\Delta \overline{w}_0$ appropriately to the new system to ensure that the physical results are invariant to the choice of the unit system.

For damage in ductile materials Abaqus/Standard assumes that the degradation of the elastic stiffness can be modeled using the scalar damage variable, D. At any given loading cycle during the analysis the stress tensor in the material is given by the scalar damage equation

$$\boldsymbol{\sigma} = (1 - D)\,\overline{\boldsymbol{\sigma}},$$

where $\overline{\boldsymbol{\sigma}}$ is the effective (or undamaged) stress tensor that would exist in the material in the absence of damage computed in the current increment. The material has completely lost its load carrying capacity when $D = 1$. You can remove the element from the mesh if all of the section points at all integration locations have lost their loading carrying capability.

Input File Usage: **DAMAGE EVOLUTION*, TYPE=HYSTERESIS ENERGY

Use the following option to specify a reference accumulated inelastic hysteresis energy density per cycle:

**DAMAGE EVOLUTION*, TYPE=HYSTERESIS ENERGY, REF ENERGY=$\Delta\overline{w}_0$

Mesh Dependency and Characteristic Length

The implementation of the damage evolution model requires the definition of a characteristic length associated with an integration point. The characteristic length is based on the element geometry and formulation: it is a typical length of a line across an element for a first-order element; it is half of the same typical length for a second-order element. For beams and trusses it is a characteristic length along the element axis. For membranes and shells it is a characteristic length in the reference surface. For axisymmetric elements it is a characteristic length in the *r*–*z* plane only. For cohesive elements it is equal to the constitutive thickness. This definition of the characteristic length is used because the direction in which fracture occurs is not known in advance. Therefore, elements with large aspect ratios will have rather different behavior depending on the direction in which the damage occurs: some mesh sensitivity remains because of this effect, and elements that are as close to square as possible are recommended. However, since the damage evolution law is energy based, mesh dependency of the results may be alleviated.

Maximum Degradation and Element Removal

You can control how Abaqus/Standard treats elements with severe damage.

Defining the Upper Bound to the Damage Variable

By default, the upper bound to all damage variables at a material point is $D_{max} = 1.0$. You can reduce this upper bound as discussed in *Controlling Element Deletion and Maximum Degradation for Materials with Damage Evolution*.

Input File Usage: **SECTION CONTROLS*, MAX DEGRADATION=D_{max}

Controlling Element Removal for Damaged Elements

A material point is considered to fail once D reaches D_{max}. By default, an element is deleted from a mesh upon material failure. Details for element deletion driven by material failure are discussed in *Material Failure and Element Deletion*. The status of an element can be determined by requesting output variable STATUS.

Alternatively, you can specify that an element should remain in the model even after all of the damage variables reach D_{max}. In this case, once all of the damage variables reach the maximum value, the stiffness remains constant.

Input File Usage: Use the following option to delete failed elements from the mesh (default):

**SECTION CONTROLS*, ELEMENT DELETION=YES

Use the following option to keep failed elements in the mesh computations:

*SECTION CONTROLS, ELEMENT DELETION=NO

Elements

Damage evolution for ductile materials can be defined for any element that can be used with the damage initiation criteria for a low-cycle fatigue analysis in Abaqus/Standard (*Damage Initiation for Ductile Materials in Low-Cycle Fatigue*).

Output

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning when damage evolution is specified:

STATUS Status of the element (the status of an element is 1.0 if the element is active, 0.0 if the element is not).

SDEG Overall scalar stiffness degradation, D.

Hydrodynamic Properties

In this section:

- *Hydrodynamic Behavior*
- *Equation of State*

Hydrodynamic Behavior

Overview

The material library in Abaqus/Explicit includes several equation of state models to describe the hydrodynamic behavior of materials.
An equation of state is a constitutive equation that defines the pressure as a function of the density and the internal energy (*Equation of State*).

The following equations of state are supported in Abaqus/Explicit:

Mie-Grüneisen equation of state

The Mie-Grüneisen equation of state (*Mie-Grüneisen Equations of State*) is used to model materials at high pressure. It is linear in energy and assumes a linear relationship between the shock velocity and the particle velocity.

Tabulated equation of state

The tabulated equation of state (*Tabulated Equation of State*) is used to model the hydrodynamic response of materials that exhibit sharp transitions in the pressure-density relationship, such as those induced by phase transformations. It is linear in energy.

P–alpha equation of state

The $P-\alpha$ equation of state (*P–alpha Equation of State*) is designed for modeling the compaction of ductile porous materials. The constitutive model captures the irreversible compaction behavior at low stresses and predicts the correct thermodynamic behavior at high pressures for the fully compacted solid material. It is used in combination with either the Mie-Grüneisen equation of state or the tabulated equation of state to describe the solid phase.

JWL high explosive equation of state

The Jones-Wilkins-Lee (or JWL) equation of state (*JWL High Explosive Equation of State*) models the pressure generated by the release of chemical energy in an explosive. This model is implemented in a form referred to as a programmed burn, which means that the reaction and initiation of the explosive is not determined by shock in the material. Instead, the initiation time is determined by a geometric construction using the detonation wave speed and the distance of the material point from the detonation points.

Ideal gas equation of state

The ideal gas equation of state (*Ideal Gas Equation of State*) is an idealization to real gas behavior and can be used to model any gases approximately under appropriate conditions (e.g., low pressure and high temperature).

References:

Deviatoric Behavior

The material modeled by an equation of state may have no deviatoric strength or may have either isotropic elastic or viscous (both Newtonian and non-Newtonian) deviatoric behavior (*Deviatoric Behavior*). The elastic model can be used by itself or in conjunction with the Mises, the Johnson-Cook, or the extended Drucker-Prager plasticity models to model hydrodynamic materials with elastic-plastic deviatoric behavior.

Thermal Strain

Thermal expansion cannot be introduced for any of the equation of state models.

Equation of State

Products: Abaqus/Explicit Abaqus/CAE

References:

- *Hydrodynamic Behavior*
- *About the Material Library*
- *VUEOS*
- **EOS*
- **EOS COMPACTION*
- **ELASTIC*
- **VISCOSITY*
- **DETONATION POINT*
- **GAS SPECIFIC HEAT*
- **REACTION RATE*
- **TENSILE FAILURE*
- *Defining equations of state*

Overview

Equations of state:

- provide a hydrodynamic material model in which the material's volumetric strength is determined by an equation of state;
- determine the pressure (positive in compression) as a function of the density, ρ, and the specific energy (the internal energy per unit mass), E_m: $p = f(\rho, E_m)$;
- are available as Mie-Grüneisen equations of state (thus providing the linear $U_s - U_p$ Hugoniot form);
- are available as tabulated equations of state linear in energy;
- are available as $P - \alpha$ equations of state for the compaction of ductile porous materials and must be used in conjunction with either the Mie-Grüneisen or the tabulated equation of state for the solid phase;
- are available as JWL high explosive equations of state;
- are available as ignition and growth equations of state;
- are available in the form of an ideal gas;
- are available in the form of user-defined equations of state (*VUEOS*);
- assume an adiabatic condition unless a dynamic fully coupled temperature-displacement analysis is used;
- can be used to model a material that has only volumetric strength (the material is assumed to have no shear strength) or a material that also has isotropic elastic or viscous deviatoric behavior;
- can be used with the Mises (*Classical Metal Plasticity*) or the Johnson-Cook (*Johnson-Cook Plasticity*) plasticity models;
- can be used with the extended Drucker-Prager (*Extended Drucker-Prager Models*) plasticity models (without plastic dilation); and

- can be used with the tensile failure model (*Dynamic Failure Models*) to model dynamic spall or a pressure cutoff.

Energy Equation and Hugoniot Curve

The equation for conservation of energy equates the increase in internal energy per unit mass, E_m, to the rate at which work is being done by the stresses and the rate at which heat is being added. In the absence of heat conduction the energy equation can be written as

$$\rho \frac{\partial E_m}{\partial t} = (p + p_{bv}) \frac{1}{\rho} \frac{\partial \rho}{\partial t} + \mathbf{S} : \dot{\mathbf{e}} + \rho \dot{Q},$$

where p is the pressure stress defined as positive in compression, p_{bv} is the pressure stress due to the bulk viscosity, $\mathbf{S}$ is the deviatoric stress tensor, $\dot{\mathbf{e}}$ is the deviatoric part of strain rate, and $\dot{Q}$ is the heat rate per unit mass.

The equation of state is assumed for the pressure as a function of the current density, ρ, and the internal energy per unit mass, E_m:

$$p = f(\rho, E_m),$$

which defines all the equilibrium states that can exist in a material. The internal energy can be eliminated from the above equation to obtain a p versus V relationship (where V is the current volume) or, equivalently, a p versus $1/\rho$ relationship that is unique to the material described by the equation of state model. This unique relationship is called the Hugoniot curve and is the locus of p–V states achievable behind a shock (see *Figure 1*).

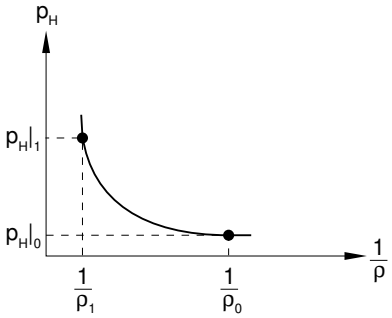


Figure 1: A schematic representation of a Hugoniot curve.

The Hugoniot pressure, p_H, is a function of density only and can be defined, in general, from fitting experimental data.

An equation of state is said to be linear in energy when it can be written in the form

$$p = f + gE_m,$$

where $f(\rho)$ and $g(\rho)$ are functions of density only and depend on the particular equation of state model.

Mie-Grüneisen Equations of State

A Mie-Grüneisen equation of state is linear in energy. The most common form is

$$p - p_H = \Gamma \rho (E_m - E_H),$$

where p_H and E_H are the Hugoniot pressure and specific energy (per unit mass) and are functions of density only, and Γ is the Grüneisen ratio defined as

$$\Gamma = \Gamma_0 \frac{\rho_0}{\rho},$$

where Γ_0 is a material constant and ρ_0 is the reference density.

The Hugoniot energy, E_H, is related to the Hugoniot pressure by

$$E_H = \frac{p_H \eta}{2\rho_0},$$

where $\eta = 1 - \rho_0/\rho$ is the nominal volumetric compressive strain. Elimination of Γ and E_H from the above equations yields

$$p = p_H \left(1 - \frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 E_m.$$

The equation of state and the energy equation represent coupled equations for pressure and internal energy. Abaqus/Explicit solves these equations simultaneously at each material point.

Linear U_s – U_p Hugoniot Form

A common fit to the Hugoniot data is given by

$$p_H = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2},$$

where c_0 is the bulk speed of sound and s is the slope of the linear $U_s - U_p$ Hugoniot form of the equation of state. Material parameters c_0 and s define the linear relationship between the shock velocity, U_s, and the particle velocity, U_p, as follows:

$$U_s = c_0 + sU_p.$$

With the above assumptions the linear $U_s - U_p$ Hugoniot form is written as

$$p = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \left(1 - \frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 E_m,$$

where $\rho_0 c_0^2$ is equivalent to the elastic bulk modulus at small nominal strains.

There is a limiting compression given by the denominator of this form of the equation of state

$$\eta_{lim} = \frac{1}{s}$$

or

$$\rho_{lim} = \frac{s\rho_0}{s-1} \quad .$$

At this limit there is a tensile minimum; thereafter, negative sound speeds are calculated for the material.

Input File Usage: Use both of the following options:

**DENSITY (to specify the reference density ρ_0)*
**EOS*, TYPE=USUP *(to specify the variables c_0, s, and Γ_0)*

Abaqus/CAE Usage: Property module: material editor:
General->**Density** *(to specify the reference density ρ_0)*
Mechanical->**Eos**: **Type: Us - Up** *(to specify the variables c_0, s, and Γ_0)*

Initial State

The initial state of the material is determined by the initial values of specific energy, E_m, and pressure stress, p. Abaqus/Explicit will automatically compute the initial density, ρ, that satisfies the equation of state, $p = f(\rho, E_m)$. You can define the initial specific energy and initial stress state (see *Initial Conditions*). The initial pressure used by the equation of state is inferred from the specified stress states. If no initial conditions are specified, Abaqus/Explicit will assume that the material is at its reference state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \rho &= \rho_0. \end{aligned}$$

Input File Usage: Use either or both of the following options, as required:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY
**INITIAL CONDITIONS*, TYPE=STRESS

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Tabulated Equation of State

The tabulated equation of state provides flexibility in modeling the hydrodynamic response of materials that exhibit sharp transitions in the pressure-density relationship, such as those induced by phase transformations. The tabulated equation of state is linear in energy and assumes the form

$$p = f_1(\varepsilon_{\text{vol}}) + \rho_0 f_2(\varepsilon_{\text{vol}}) E_m,$$

where $f_1(\varepsilon_{\text{vol}})$ and $f_2(\varepsilon_{\text{vol}})$ are functions of the logarithmic volumetric strain ε_{vol} only, with $\varepsilon_{\text{vol}} = \ln(\rho_0/\rho)$, and ρ_0 is the reference density.

You can specify the functions $f_1(\varepsilon_{\text{vol}})$ and $f_2(\varepsilon_{\text{vol}})$ directly in tabular form. The tabular entries must be given in descending values of the volumetric strain (that is, from the most tensile to the most compressive states). Abaqus/Explicit will use a piecewise linear relationship between data points. Outside the range of specified values of volumetric strains, the functions are extrapolated based on the last slope computed from the data.

Input File Usage: Use both of the following options:

**DENSITY (to specify the reference density ρ_0)*
**EOS*, TYPE=TABULAR *(to specify f_1 and f_2 as functions of ε_{vol})*

Abaqus/CAE Usage: Property module: material editor:
General->**Density** *(to specify the reference density ρ_0)*
Mechanical->**Eos**: **Type: Tabular** *(to specify f_1 and f_2 as functions of ε_{vol})*

Initial State

The initial state of the material is determined by the initial values of specific energy, E_m, and pressure stress, p. Abaqus/Explicit automatically computes the initial density, ρ, that satisfies the equation of state. You can define the initial specific energy and initial stress state (see *Initial Conditions*). The initial pressure used by the equation of state is inferred from the specified stress states. If no initial conditions are specified, Abaqus/Explicit assumes that the material is at its reference state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \rho &= \rho_0 \qquad (\varepsilon_{vol} = 0). \end{aligned}$$

Input File Usage: Use either or both of the following options, as required:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY
**INITIAL CONDITIONS*, TYPE=STRESS

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

User-Defined Equation of State

The user-defined equation of state provides a general capability for modeling the volumetric response of materials through user subroutine *VUEOS* (see *VUEOS*). The equation of state defines the pressure as a function of the current density, ρ, and the internal energy per unit mass, E_m: $p = f(\rho, E_m)$. Abaqus/Explicit solves the energy equation together with the equation of state using an iterative method. The pressure stress, p, and the derivatives of the pressure with respect to the internal energy and to the density, $\partial p / \partial E_m$ and $\partial p / \partial \rho$, must be provided by user subroutine *VUEOS*. The latter is needed for the evaluation of the effective bulk modulus of the material, which is necessary for the stable time increment calculation.

Optionally, you can also specify the number of property values needed as data in the user subroutine as well as the number of solution-dependent variables (see *About User Subroutines and Utilities*).

Input File Usage: Use the following option:

**EOS*, TYPE=USER, PROPERTIES=*n*

Abaqus/CAE Usage: The user-defined equation of state is not supported in Abaqus/CAE.

Initial State

You need to make sure that the initial specific energy, the initial stress, and the initial density satisfy the equation of state. If you do not specify the initial conditions, Abaqus/Explicit assumes that the material is at its reference state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \rho &= \rho_0. \end{aligned}$$

Input File Usage: Use either or both of the following options to define the initial specific energy and/or initial pressure stress:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY
**INITIAL CONDITIONS*, TYPE=STRESS

Use the following option to define the initial density:

**DENSITY*

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

P–alpha Equation of State

The $P-\alpha$ equation of state is designed for modeling the compaction of ductile porous materials. The implementation in Abaqus/Explicit is based on the model proposed by Hermann (1968) and Carroll and Holt (1972). The constitutive model provides a detailed description of the irreversible compaction behavior at low stresses and predicts the correct thermodynamic behavior at high pressures for the fully compacted solid material. In Abaqus/Explicit the solid phase is assumed to be governed by either the Mie-Grüneisen equation of state or the tabulated equation of state. The relevant properties of the porous material in the virgin state, to be discussed later, and the material properties of the solid phase are specified separately.

The porosity of the material, *n*, is defined as the ratio of pore volume, V_p, to total volume, $V = V_s + V_p$, where V_s is the solid volume. The porosity remains in the range $0 \le n < 1$, with *0* indicating full compaction. It is convenient to introduce a scalar variable α, sometimes referred to as "distension," defined as the ratio of the density of the solid material, ρ_s, to the density of the porous material, ρ, both evaluated at the same temperature and pressure:

$$\alpha = \frac{\rho_s}{\rho} \ge 1.$$

For a fully compacted material $\alpha = 1$; otherwise, α is greater than *1*. Assuming that the density of the pores is negligible compared to that of the solid phase, α can be expressed in terms of the porosity *n* as

$$\alpha = \frac{\rho_s}{\rho} = \frac{V}{V_s} = \frac{V}{V-V_p} = \frac{1}{1-V_p/V} = \frac{1}{1-n}.$$

An equation of state is assumed for the pressure of the porous material as a function of α; current density, ρ; and internal energy per unit mass, E_m, in the form

$$p = p\left(\alpha, \rho, E_m\right).$$

Assuming that the pores carry no pressure, it follows from equilibrium considerations that when a pressure p is applied to the porous material, it gives rise to a volume-average pressure in the solid phase equal to $p_s = \alpha p$. Assuming that the specific internal energies of the porous material and the solid matrix are the same (that is, neglecting the surface energy of the pores), the equation of state of the porous material can be expressed as

$$p\left(\alpha, \rho, E_m\right) = \frac{1}{\alpha} p_s\left(\alpha\rho, E_m\right) = \frac{1}{\alpha} p_s\left(\rho_s, E_m\right),$$

where $p_s\left(\rho_s, E_m\right)$ is the equation of state of the solid material. For the fully compacted material (that is, when $\alpha = 1$), the $P - \alpha$ equation of state reduces to that of the solid phase, therefore predicting the correct thermodynamic behavior at high pressures.

The $P - \alpha$ equation of state must be supplemented by an equation that describes the behavior of α as a function of the thermodynamic state. This equation takes the form

$$\alpha = A\left(p, \alpha_{min}\right),$$

where α_{min} is a state variable corresponding to the minimum value attained by α during plastic (irreversible) compaction of the material. The state variable is initialized to the elastic limit α_e for a material that is at its virgin state. The specific form of the function $A\left(p, \alpha_{min}\right)$ used by Abaqus/Explicit is illustrated in *Figure 2* and is discussed next.

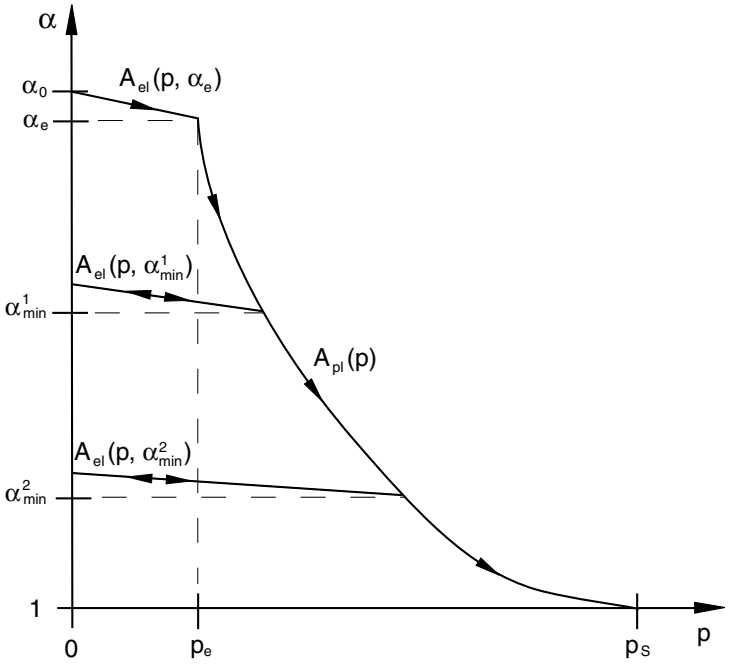


Figure 2: $P - \alpha$ elastic and plastic curves for the description of compaction of ductile porous materials.

The function $A\left(p, \alpha_{min}\right)$ captures the general behavior to be expected in a ductile porous material. The unloaded virgin state corresponds to the value $\alpha_0 = 1/\left(1 - n_0\right)$, where n_0 is the reference porosity of the material. Initial compression of the porous material is assumed to be elastic. Recall that decreasing porosity corresponds to a reduction in α. As the pressure increases beyond the elastic limit, p_e, the pores in the material start to crush, leading to irreversible compaction and permanent (plastic) volume change. Unloading from a partially compacted state follows a new elastic curve that depends on the maximum compaction (or, alternatively, the minimum

value of α) ever attained during the deformation history of the material. The absolute value of the slope of the elastic curve decreases as α_{min} decreases, as will be quantified later. The material becomes fully compacted when the pressure reaches the compaction pressure p_S; at that point $\alpha = \alpha_{min} = 1$, a value that is retained forever. The function $A(p, \alpha_{min})$ therefore has multiple branches: a plastic branch, $A_{pl}(p)$, and multiple elastic branches, $A_{el}(p, \alpha_{min})$, corresponding to elastic unloading from partially compacted states. The appropriate branch of A is selected according to the following rule:

$$\alpha = A(p, \alpha_{min}) = \begin{cases} A_{pl}(p) & \text{if } A_{pl}(p) \le \alpha_{min} \\ A_{el}(p, \alpha_{min}) & \text{if } A_{pl}(p) > \alpha_{min} \end{cases}$$

These expressions can be inverted to solve for p:

$$p = P(\alpha, \alpha_{min}) = \begin{cases} P_{pl}(\alpha) & \text{if } \alpha \le \alpha_{min} \\ P_{el}(\alpha, \alpha_{min}) & \text{if } \alpha > \alpha_{min} \end{cases}$$

The equation for the plastic curve takes the form

$$A_{pl}(p) = 1 + (\alpha_e - 1)\left(\frac{p_S - p}{p_S - p_e}\right)^2$$

or, alternatively,

$$P_{pl}(\alpha) = p_S - (p_S - p_e)\left(\frac{\alpha - 1}{\alpha_e - 1}\right)^{\frac{1}{2}}.$$

The elastic curve originally proposed by Hermann (1968) is given by the differential equation

$$\frac{d\overline{A}_{el}}{dp}(\alpha) = \frac{\alpha^2}{K_0}\left(1 - \frac{1}{h^2(\alpha)}\right), \qquad h(\alpha) = 1 + \frac{(c_e - c_s)(\alpha - 1)}{c_s(\alpha_0 - 1)},$$

where $K_0 = \rho_{s0} c_s^2$ is the elastic bulk modulus of the solid material at small nominal strains; ρ_{s0} is the reference density of the solid; and c_s and c_e are the reference sound speeds in the solid and virgin (porous) materials, respectively.

If the solid phase is modeled using the Mie-Grüneisen equation of state, c_s is given directly by the reference sound speed, c_0. On the other hand, if the solid phase is modeled using the tabulated equation of state, c_s is computed from the initial bulk modulus and reference density of the solid material, $c_s = \sqrt{K_0/\rho_{s0}}$. In this case the reference density is required to be constant; it cannot be a function of temperature or field variables.

Following Wardlaw et al. (1996), the above equation for the elastic curve in Abaqus/Explicit is simplified and replaced by the linear relations

$$A_{el}(p, \alpha_{min}) = \alpha_{min} + (p - P_{pl}(\alpha_{min}))\left.\frac{d\overline{A}_{el}}{dp}\right|_{\alpha = \alpha_{min}}$$

and

$$P_{el}(\alpha, \alpha_{min}) = P_{pl}(\alpha_{min}) + \frac{(\alpha - \alpha_{min})}{\left.\frac{d\overline{A}_{el}}{dp}\right|_{\alpha = \alpha_{min}}}.$$

Input File Usage: Use the following option to specify the reference density of the solid phase, ρ_{s0}:

**DENSITY*

Use one of the following options to specify additional material properties for the solid phase:

**EOS*, TYPE=USUP *(if the solid phase is modeled using the Mie-Grüneisen equation of state)*
**EOS*, TYPE=TABULAR *(if the solid phase is modeled using the tabulated equation of state)*

Use the following option to specify the properties of the porous material (the reference sound speed, c_e; the reference porosity, n_0; the elastic limit, p_e; and the compaction pressure, p_s):

**EOS COMPACTION*

Abaqus/CAE Usage: Property module: material editor:
General->Density *(to specify the reference density ρ_0)*

Use one of the following options to specify additional material properties for the solid phase:

Mechanical->Eos: **Type: Us - Up** *(if the solid phase is modeled using the Mie-Grüneisen equation of state)*
Mechanical->Eos: **Type: Tabular** *(if the solid phase is modeled using the tabulated equation of state)*

Use the following option to specify the properties of the porous material:

Mechanical->Eos: **Suboptions-> Eos Compaction** *(to specify the reference sound speed, c_e; the porosity of the unloaded material, n_0; the pressure required to initialize plastic behavior, p_e; and the pressure at which all pores are crushed, p_s)*

Initial State

The initial state of the porous material is determined from the initial values of porosity, $n = (\alpha - 1)/\alpha$; specific energy, E_m; and pressure stress, p. Abaqus/Explicit automatically computes the initial density, ρ, that satisfies the equation of state, $p = f(\alpha, \rho, E_m)$. You can define the initial porosity, initial specific energy, and initial stress state (see *Initial Conditions*). If no initial conditions are given, Abaqus/Explicit assumes that the material is at its virgin state:

$$\begin{aligned} E_m &= 0, \\ p &= 0, \\ \alpha &= \alpha_0, \qquad (n = n_0), \\ \rho &= \rho_{s0}/\alpha_0. \end{aligned}$$

Abaqus/Explicit will issue an error message if the initial $p - \alpha$ state lies outside the region of allowed states (see *Figure 2*). When initial conditions are specified only for p (or for α), Abaqus/Explicit will compute α (or p) assuming that the $p - \alpha$ state lies on the primary (monotonic loading) curve.

Input File Usage: Use some or all of the following options, as required:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY
**INITIAL CONDITIONS*, TYPE=STRESS
**INITIAL CONDITIONS*, TYPE=POROSITY

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy and initial porosity are not supported in Abaqus/CAE.

JWL High Explosive Equation of State

The Jones-Wilkins-Lee (or JWL) equation of state models the pressure generated by the release of chemical energy in an explosive. This model is implemented in a form referred to as a programmed burn, which means that the reaction and initiation of the explosive is not determined by shock in the material. Instead, the initiation time is determined by a geometric construction using the detonation wave speed and the distance of the material point from the detonation points.

The JWL equation of state can be written in terms of the internal energy per unit mass, E_m, as

$$p = A\left(1 - \frac{\omega\rho}{R_1\rho_0}\right)\exp\left(-R_1\frac{\rho_0}{\rho}\right) + B\left(1 - \frac{\omega\rho}{R_2\rho_0}\right)\exp\left(-R_2\frac{\rho_0}{\rho}\right) + \omega\rho E_m,$$

where $A, B, R_1, R_2,$ and ω are user-defined material constants; ρ_0 is the user-defined density of the explosive; and ρ is the density of the detonation products.

Input File Usage: Use both of the following options:

**DENSITY (to specify the density of the explosive ρ_0)*
**EOS*, TYPE=JWL *(to specify the material constants $A, B, R_1, R_2,$ and ω)*

Abaqus/CAE Usage: Property module: material editor:
General->Density *(to specify the density of the explosive ρ_0)*
Mechanical->Eos: **Type: JWL** *(to specify the material constants $A, B, R_1, R_2,$ and ω)*

Arrival Time of Detonation Wave

Abaqus/Explicit calculates the arrival time of the detonation wave at a material point $\left(t_d^{mp}\right)$ as the distance from the material point to the nearest detonation point divided by the detonation wave speed:

$$t_d^{mp} = \min\left[t_d^N + \sqrt{\left(\mathbf{x}^{mp} - \mathbf{x}_d^N\right)\cdot\left(\mathbf{x}^{mp} - \mathbf{x}_d^N\right)}\Big/ C_d\right],$$

where $\mathbf{x}^{mp}$ is the position of the material point, $\mathbf{x}_d^N$ is the position of the *N*th detonation point, t_d^N is the detonation delay time of the *N*th detonation point, and C_d is the detonation wave speed of the explosive material. The minimum in the above formula is over the *N* detonation points that apply to the material point.

Burn Fraction

To spread the burn wave over several elements, a burn fraction, F_b, is computed as

$$F_b = \min\left[1, \frac{(t - t_d^{mp})C_d}{B_s l_e}\right],$$

where B_s is a constant that controls the width of the burn wave (set to a value of 2.5) and l_e is the characteristic length of the element. If the time is less than t_d^{mp}, the pressure is zero in the explosive; otherwise, the pressure is given by the product of F_b and the pressure determined from the JWL equation above.

Defining Detonation Points

You can define any number of detonation points for the explosive material. Coordinates of the points must be defined along with a detonation delay time. Each material point responds to the first detonation point that it sees. The detonation arrival time at a material point is based on the time that it takes a detonation wave (traveling at the detonation wave speed C_d) to reach the material point plus the detonation delay time for the detonation point. If there are multiple detonation points, the arrival time is based on the minimum arrival time for all the detonation points. In a body with curved surfaces care should be taken that the detonation arrival times are meaningful. The detonation arrival times are based on the straight line of sight from the material point to the detonation point. In a curved body the line of sight might pass outside of the body.

Input File Usage: Use both of the following options to define the detonation points:

**EOS*, TYPE=JWL
**DETONATION POINT*

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Eos**: **Type: JWL**: **Suboptions**->**Detonation Point**

Initial State

Explosive materials generally have some nominal volumetric stiffness before detonation. It might be useful to incorporate this stiffness when elements modeled with a JWL equation of state are subjected to stress before initiation of detonation by the arriving detonation wave. You can define the pre-detonation bulk modulus, K_{pd}. The pressure will be computed from the volumetric strain and K_{pd} until detonation, at which time the pressure will be determined by the procedure outlined above. The initial relative density (ρ/ρ_0) used in the JWL equation is assumed to be unity. The initial specific energy E_{m_0} is assumed to be equal to the user-defined detonation energy E_0.

If you specify a nonzero value of K_{pd}, you can also define an initial stress state for the explosive materials.

Input File Usage: Use the following option to define the initial stress:

**INITIAL CONDITIONS*, TYPE=STRESS

Optionally, you can also define the initial specific energy directly:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Ignition and Growth Equation of State

The ignition and growth equation of state models shock initiation and detonation wave propagation of solid high explosives. The heterogeneous explosive is modeled as a homogeneous mixture of two phases: the unreacted solid explosive and the reacted gas products. Separate JWL equations of state are prescribed for each phase:

$$p_s = \widetilde{F}_{1s}\left(\rho_s\right) - \widetilde{F}_{1s}\left(\rho_0\right) + \widetilde{F}_{2s}\left(\rho_s\right) E_{ms},$$
$$p_g = \widetilde{F}_{1g}\left(\rho_g\right) + \widetilde{F}_{2g}\left(\rho_g\right)\left(E_{mg} + E_d\right),$$

where

$$\widetilde{F}_{1i}\left(\rho_i\right) = A_i\left(1 - \frac{\omega_i\rho_i}{R_{1i}\rho_0}\right)\exp\left(-R_{1i}\frac{\rho_0}{\rho_i}\right) + B_i\left(1 - \frac{\omega_i\rho_i}{R_{2i}\rho_0}\right)\exp\left(-R_{2i}\frac{\rho_0}{\rho_i}\right)$$

and

$$\widetilde{F}_{2i}\left(\rho_i\right) = \omega_i\rho_i, \qquad (i = s, g).$$

The subscript *s* refers to the unreacted solid explosive, and *g* refers to the reacted gas products. A_i, B_i, R_{1i}, R_{2i}, and ω_i are user-defined material constants used in the JWL equations; E_d is the detonation energy; ρ_0 is the user-defined reference density of the explosive, and ρ_i is the density of the unreacted explosive or the reacted products.

Input File Usage: Use both of the following options:

**DENSITY(to specify the density of the explosive ρ_0)*
**EOS*, TYPE=IGNITION AND GROWTH, DETONATION ENERGY=E_d
(to specify the material constants A, B, R_1, R_2, and ω of the unreacted solid explosive and the reacted gas product)

Abaqus/CAE Usage: Property module: material editor:
General->Density *(to specify the density of the explosive ρ_0)*
Mechanical->Eos: **Type: Ignition and growth**: **Detonation energy**: E_d; **Solid Phase** tabbed page and **Gas Phase** tabbed page
(to specify the material constants A, B, R_1, R_2, and ω of the unreacted solid explosive and the reacted gas product)

Mass Fraction

The mixture of unreacted solid explosive and reacted gas products is defined by the mass fraction

$$F_i = \frac{m_i}{\left(m_s + m_g\right)},$$

where m_s is the mass of the unreacted explosive, and m_g is the mass of the reacted products. It is assumed that the two phases are in thermomechanical equilibrium:

$$p_s = p_g \qquad \text{and} \qquad T_s = T_g.$$

It is also assumed that the volumes are additive:

$$V = V_s + V_g \text{ or } \frac{1}{\rho} = (1 - F)\frac{1}{\rho_s} + F\frac{1}{\rho_g}.$$

Similarly, the internal energy is assumed to be additive:

$$(m_s + m_g) E_m = m_s E_{ms} + m_g E_{mg},$$

where

$$E_m = E_{m_0} + \int_{\theta_0 - \theta^Z}^{\theta - \theta^Z} c_v(T)\, dT.$$

Hence, the specific heat of the mixture is given by

$$c_v = (1 - F) c_{vs} + F c_{vg}.$$

Input File Usage: Use the following options to define the specific heat of the unreacted solid explosive:

**EOS*, TYPE=IGNITION AND GROWTH
**SPECIFIC HEAT*, DEPENDENCIES=*n*

Use the following options to define the specific heat of the reacted gas product:

**EOS*, TYPE=IGNITION AND GROWTH
**GAS SPECIFIC HEAT*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Use the following options to define the specific heat of the unreacted solid explosive:

Property module: material editor:
Mechanical->Eos: **Type: Ignition and GrowthThermal->Specific Heat**

Use the following options to define the specific heat of the reacted gas product:

Property module: material editor:
Mechanical->Eos: **Type: Ignition and growth**:
Gas Specific tabbed page: **Specific Heat**

You can toggle on **Use temperature-dependent data** to define the specific heat as a function of temperature or select the **Number of field variables** to define the specific heat as a function of field variables.

Reaction Rate

The conversion of unreacted solid explosive to reacted gas products is governed by the reaction rate. The reaction rate equation in the ignition and growth model is a pressure-driven rule, which includes three terms:

$$\frac{dF}{dt} = \dot{F}_{ig} + \dot{F}_{G_1} + \dot{F}_{G_2}.$$

These three terms are defined as follows:

$$\dot{F}_{ig} = I(1-F)^b\left(\frac{\rho_s}{\rho_0}-1-a\right)^x,$$
$$\dot{F}_{G_1} = G_1(1-F)^c F^d\left(\frac{p}{p_{ref}}\right)^y,$$
$$\dot{F}_{G_2} = G_2(1-F)^e F^g\left(\frac{p}{p_{ref}}\right)^z,$$

where I, G_1, G_2, a, b, c, d, e, g, x, y, and z are reaction rate constants; and p_{ref} is a reference pressure.

The first term, $\dot{F}_{ig}$, describes hot spot ignition by igniting some of the material relatively quickly but limiting it to a small proportion of the total solid F_{ig}^{max}. The second term, $\dot{F}_{G_1}$, represents the growth of reaction from the hot spot sites into the material and describes the inward and outward grain burning phenomena; this term is limited to a proportion of the total solid $F_{G_1}^{max}$. The third term, $\dot{F}_{G_2}$, is used to describe the rapid transition to detonation observed in some energetic materials.

$$\dot{F}_{ig} = 0 \text{ if } F \geq F_{ig}^{max}$$
$$\dot{F}_{G_1} = 0 \text{ if } F \geq F_{G_1}^{max}$$
$$\dot{F}_{G_2} = 0 \text{ if } F \leq F_{G_2}^{min}$$

Input File Usage: Use both of the following options to define the reaction rate:

**EOS*, TYPE=IGNITION AND GROWTH
**REACTION RATE*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Eos: **Type: Ignition and growth**:
Reaction Rate tabbed page

Initial State

The initial mass fraction of the unreacted solid explosive is assumed to be one. The initial relative density (ρ/ρ_0) used in the ignition and growth equation is assumed to be unity. The initial specific energy can be defined for the unreacted explosive.

Input File Usage: Use the following option to define the initial specific energy:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Initial specific energy is not supported in Abaqus/CAE.

Ideal Gas Equation of State

An ideal gas equation of state can be written in the form of

$$p + p_A = \rho R\left(\theta - \theta^Z\right),$$

where p_A is the ambient pressure, R is the gas constant, θ is the current temperature, and θ^Z is the absolute zero on the temperature scale being used. It is an idealization to real gas behavior and can be used to model any gases approximately under appropriate conditions (for example, low pressure and high temperature).

One of the important features of an ideal gas is that its specific energy depends only on its temperature; therefore, the specific energy can be integrated numerically as

$$E_m = E_{m_0} + \int_{\theta_0-\theta^Z}^{\theta-\theta^Z} c_v\,(T)\,dT,$$

where E_{m_0} is the initial specific energy at the initial temperature θ_0 and c_v is the specific heat at constant volume (or the constant volume heat capacity), which depends only on temperature for an ideal gas.

Modeling with an ideal gas equation of state is typically performed adiabatically; the temperature increase is calculated directly at the material integration points according to the adiabatic thermal energy increase caused by the work pdv, where v is the specific volume (the volume per unit mass, $v = 1/\rho$). Therefore, unless a fully coupled temperature-displacement analysis is performed, an adiabatic condition is always assumed in Abaqus/Explicit.

When performing a fully coupled temperature-displacement analysis, the pressure stress and specific energy are updated based on the evolving temperature field. The energy increase due to the change in state will be accounted for in the heat equation and will be subject to heat conduction.

For the ideal gas model in Abaqus/Explicit you define the gas constant, R, and the ambient pressure, p_A. For an ideal gas R can be determined from the universal gas constant, $\tilde{R}$, and the molecular weight, MW, as follows:

$$R = \frac{\tilde{R}}{MW}.$$

In general, the value R for any gas can be estimated by plotting $pv/\left(\theta - \theta^Z\right)$ as a function of state (for example, pressure or temperature). The ideal gas approximation is adequate in any region where this value is constant. You must specify the specific heat at constant volume, c_v. For an ideal gas c_v is related to the specific heat at constant pressure, c_p, by

$$R = c_p - c_v.$$

Input File Usage: Use both of the following options:

**EOS*, TYPE=IDEAL GAS
**SPECIFIC HEAT*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Eos: **Type: Ideal Gas**
Thermal->Specific Heat

Initial State

There are different methods to define the initial state of the gas. You can specify the initial density, ρ, and either the initial pressure stress, p_0, or the initial temperature, θ_0. The initial value of the unspecified field (temperature or pressure) is determined from the equation of state. Alternatively, you can specify both the initial pressure stress and the initial temperature. In this case the user-specified initial density is replaced by that derived from the equation of state in terms of initial pressure and temperature.

By default, Abaqus/Explicit automatically computes the initial specific energy, E_{m_0}, from the initial temperature by numerically integrating the equation

$$E_{m_0} = \int_0^{\theta_0 - \theta^Z} c_v(T)\, dT.$$

Optionally, you can override this default behavior by defining the initial specific energy for the ideal gas directly.

Input File Usage: Use some or all of the following options, as required:

**DENSITY*, DEPENDENCIES=*n*
**INITIAL CONDITIONS*, TYPE=STRESS
**INITIAL CONDITIONS*, TYPE=TEMPERATURE

Use the following option to specify the initial specific energy directly:

**INITIAL CONDITIONS*, TYPE=SPECIFIC ENERGY

Abaqus/CAE Usage: Property module: material editor: **General->Density**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Temperature** for the **Types for Selected Step**

Load module: **Create Predefined Field**: **Step: Initial**: choose **Mechanical** for the **Category** and **Stress** for the **Types for Selected Step**

Initial specific energy is not supported in Abaqus/CAE.

Absolute Zero Value

When a non-absolute temperature scale is used, you must specify the value of absolute zero temperature.

Input File Usage: **PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Any module: **Model->Edit Attributes->*model_name*: Absolute zero temperature**

Special Case

In the case of an adiabatic analysis with constant specific heat (both c_v and c_p are constant), the specific energy is linear in temperature

$$E_m = c_v\left(\theta - \theta^Z\right).$$

The pressure stress can, therefore, be recast in the common form of

$$p + p_A = (\gamma - 1)\,\rho E_m,$$

where $\gamma = c_p/c_v$ is the ratio of specific heats and can be defined as

$$\gamma = \frac{n+2}{n},$$

where

$n = 3$

for a monatomic;

$n = 5$

for a diatomic; and

$n = 6$

for a polyatomic gas.

Comparison with the Hydrostatic Fluid Model

The ideal gas equation of state can be used to model wave propagation effects and the dynamics of a spatially varying state of a gaseous region. For cases in which the inertial effects of the gas are not important and the state of the gas can be assumed to be uniform throughout a region, the hydrostatic fluid model (*About Surface-Based Fluid Cavities*) is a simpler, more computationally efficient alternative.

Deviatoric Behavior

The equation of state defines only the material's hydrostatic behavior. It can be used by itself, in which case the material has only volumetric strength (the material is assumed to have no shear strength). Alternatively, Abaqus/Explicit allows you to define deviatoric behavior, assuming that the deviatoric and volumetric responses are uncoupled. Three options are available for the deviatoric response: a linear isotropic elastic model, a viscous model, and a nonlinear viscoelastic model. The material's volumetric response is governed then by the equation of state model, while its deviatoric response is governed by either the linear isotropic elastic model, the viscous fluid model or the nonlinear viscoelastic model.

Elastic Shear Behavior

For the elastic shear behavior the deviatoric stress is related to the deviatoric strain as

$$\mathbf{S} = 2\mu\mathbf{e}^{el},$$

where $\mathbf{S}$ is the deviatoric stress and $\mathbf{e}^{el}$ is the deviatoric elastic strain. See *Defining Isotropic Shear Elasticity for Equations of State in Abaqus/Explicit* for more details.

Input File Usage: Use both of the following options to define elastic shear behavior:

**EOS*
**ELASTIC*, TYPE=SHEAR

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Elasticity->Elastic**; **Type: Shear**; **Shear Modulus**

Viscous Shear Behavior

For the viscous shear behavior the deviatoric stress is related to the deviatoric strain rate as

$$\mathbf{S} = 2\eta\dot{\mathbf{e}} = \eta\dot{\gamma},$$

where $\mathbf{S}$ is the deviatoric stress, $\dot{\mathbf{e}}$ is the deviatoric part of the strain rate, η is the viscosity, and $\dot{\boldsymbol{\gamma}} = 2\dot{\mathbf{e}}$ is the engineering shear strain rate.

Abaqus/Explicit provides a wide range of viscosity models to describe both Newtonian and non-Newtonian fluids. These are described in *Viscosity*.

Input File Usage: Use both of the following options to define viscous shear behavior:

**EOS*
**VISCOSITY*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Viscosity**

Nonlinear Viscoelastic Shear Behavior

The combination of the elastic and viscous shear behaviors results in a nonlinear viscoelastic shear definition (based on a Maxwell viscoelastic model), such that the deviatoric stress and the deviatoric strain satisfy the following equation:

$$\frac{\dot{\mathbf{S}}}{2\mu} + \frac{\mathbf{S}}{2\eta} = \dot{\mathbf{e}},$$

where $\mathbf{S}$ is the deviatoric stress, $\dot{\mathbf{e}}$ is the deviatoric part of the strain rate, η is the viscosity, and μ is the elastic shear modulus.

In addition to modeling nonlinear viscoelastic shear behavior, you can also use this model to simulate purely viscous fluids with very large viscosity by carefully choosing a penalty value for the elastic shear modulus, μ, to achieve a reasonable stable time increment. You can combine elastic shear behavior with viscous shear behavior only; you cannot add any additional plastic shear behavior to the material definition.

Input File Usage: Use both of the following options to define the combined elastic and viscous shear behavior:

**EOS*,
**ELASTIC*, TYPE=SHEAR
**VISCOSITY*

Abaqus/CAE Usage: Property module: material editor:

Mechanical->Elasticity->Elastic; **Type: Shear**; **Shear Modulus**

Mechanical->Viscosity

Use with the Mises or the Johnson-Cook Plasticity Models

An equation of state model, with the exception of the ignition and growth model, can be used with the Mises (*Classical Metal Plasticity*) or the Johnson-Cook (*Johnson-Cook Plasticity*) plasticity models to model elastic-plastic behavior. In this case you must define the elastic part of the shear behavior. The material's volumetric response is governed by the equation of state model, while the deviatoric response is governed by the linear elastic shear and the plasticity model.

Input File Usage: Use the following options:

**EOS*
**ELASTIC*, TYPE=SHEAR
**PLASTIC*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Elasticity->Elastic; **Type: Shear**
Mechanical->Plasticity->Plastic

Initial Conditions

You can specify initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ (*Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use with the Extended Drucker-Prager Plasticity Models

An equation of state model, with the exception of the ignition and growth model, can be used in conjunction with the extended Drucker-Prager (*Extended Drucker-Prager Models*) plasticity models to model pressure-dependent plasticity behavior. This approach can be appropriate for modeling the response of ceramics and other brittle materials under high velocity impact conditions. In this case you must define the elastic part of the shear behavior. The material's deviatoric response is governed by the linear elastic shear and the pressure-dependent plasticity model, while the volumetric response is governed by the equation of state model. In particular, no plastic dilation effects are taken into account (if you specify a dilation angle other than zero, the value is ignored and Abaqus/Explicit issues a warning message).

High-velocity impact of a ceramic target illustrates the use of an equation of state model with the extended Drucker-Prager plasticity models.

Input File Usage: Use the following options:

**EOS*
**ELASTIC*, TYPE=SHEAR
**DRUCKER PRAGER*
**DRUCKER PRAGER HARDENING*

Abaqus/CAE Usage: Property module: material editor:
Mechanical->Elasticity->Elastic; **Type: Shear**
Mechanical->Plasticity->Drucker Prager: **Suboptions->Drucker Prager Hardening**

Initial Conditions

You can specify initial conditions for the equivalent plastic strain, $\bar{\varepsilon}^{pl}$ (*Initial Conditions*).

Input File Usage: **INITIAL CONDITIONS*, TYPE=HARDENING

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**, choose **Mechanical** for the **Category** and **Hardening** for the **Types for Selected Step**

Use with the Tensile Failure Model

An equation of state model (except the ideal gas equation of state) can also be used with the tensile failure model (*Dynamic Failure Models*) to model dynamic spall or a pressure cutoff. The tensile failure model uses the hydrostatic pressure stress as a failure measure and offers a number of failure choices. You must provide the hydrostatic cutoff stress.

You can specify that the deviatoric stresses should fail when the tensile failure criterion is met. In the case where the material's deviatoric behavior is not defined, this specification is meaningless and is, therefore, ignored.

The tensile failure model in Abaqus/Explicit is designed for high-strain-rate dynamic problems in which inertia effects are important. Therefore, it should be used only for such situations. Improper use of the tensile failure model may result in an incorrect simulation.

Input File Usage: Use the following options:

**EOS*
**TENSILE FAILURE*

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Eos**: **Suboptions->Tensile Failure**

Adiabatic Assumption

An adiabatic condition is always assumed for materials modeled with an equation of state unless a dynamic coupled temperature-displacement procedure is used. The adiabatic condition is assumed irrespective of whether an adiabatic dynamic stress analysis step has been specified. The temperature increase is calculated directly at the material integration points according to the adiabatic thermal energy increase caused by the mechanical work

$$\rho c_v(\theta)\frac{\partial\theta}{\partial t} = (p - p_{bv})\frac{1}{\rho}\frac{\partial\rho}{\partial t} + \mathbf{S} : \dot{\mathbf{e}},$$

where c_v is the specific heat at constant volume. Specifying temperature as a predefined field has no effect on the behavior of this model.

When performing a fully coupled temperature-displacement analysis, the specific energy is updated based on the evolving temperature field using

$$\rho\frac{\partial E_m}{\partial t} = \rho c_v(\theta)\frac{\partial\theta}{\partial t}.$$

Modeling Fluids

A linear $U_s - U_p$ equation of state model can be used to model incompressible viscous and inviscid laminar flow governed by the Navier-Stokes equation of motion. The volumetric response is governed by the equations of state, where the bulk modulus acts as a penalty parameter for the incompressible constraint.

To model a viscous laminar flow that follows the Navier-Poisson law of a Newtonian fluid, use the Newtonian viscous deviatoric model and define the viscosity as the real linear viscosity of the fluid. To model non-Newtonian viscous flow, use one of the nonlinear viscosity models available in Abaqus/Explicit. Appropriate initial conditions for velocity and stress are essential to get an accurate solution for this class of problems.

To model an incompressible inviscid fluid such as water in Abaqus/Explicit, it is useful to define a small amount of shear resistance to suppress shear modes that can otherwise tangle the mesh. Here the shear stiffness or shear

viscosity acts as a penalty parameter. The shear modulus or viscosity should be small because flow is inviscid; a high shear modulus or viscosity will result in an overly stiff response. To avoid an overly stiff response, the internal forces arising due to the deviatoric response of the material should be kept several orders of magnitude below the forces arising due to the volumetric response. This can be done by choosing an elastic shear modulus that is several orders of magnitude lower than the bulk modulus. If the viscous model is used, the shear viscosity specified should be on the order of the shear modulus, calculated as above, scaled by the stable time increment. The expected stable time increment can be obtained from a data check analysis of the model. This method is a convenient way to approximate a shear resistance that will not introduce excessive viscosity in the material.

If a shear model is defined, the hourglass control forces are calculated based on the shear resistance of the material. Thus, in materials with extremely low or zero shear strengths such as inviscid fluids, the hourglass forces calculated based on the default parameters are insufficient to prevent spurious hourglass modes. Therefore, a sufficiently high hourglass scaling factor is recommended to increase the resistance to such modes.

Elements

In general, the equations of state can be used with any solid (continuum) elements in Abaqus/Explicit except plane stress elements. The ignition and growth equation of state cannot be used with plane strain, plane stress, and axisymmetric elements. For three-dimensional applications exhibiting high confinement, the default kinematic formulation is recommended with reduced-integration solid elements (see *Section Controls*).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Explicit Output Variable Identifiers*), the following variables have special meaning for the equation of state models:

PALPH

Distension, α, of the $P-\alpha$ porous material. The current porosity is equal to one minus the inverse of α: $n = 1 - \alpha^{-1}$.

PALPHMIN

Minimum value, α_{min}, of the distension attained during plastic compaction of the $P-\alpha$ porous material.

PEEQ

Equivalent plastic strain, $\bar{\varepsilon}^{pl} = \bar{\varepsilon}^{pl}\big|_0 + \int_0^t \sqrt{\frac{2}{3}\dot{\varepsilon}^{pl} : \dot{\varepsilon}^{pl}}\, dt$, where $\bar{\varepsilon}^{pl}\big|_0$ is the initial equivalent plastic strain (zero or user-specified; see *Initial Conditions*). This is relevant only if the equation of state model is used in combination with the Mises, Johnson-Cook, or extended Drucker-Prager plasticity models.

References

- Carroll, M., and A. C. Holt, "Suggested Modification of the $P-\alpha$ Model for Porous Materials," *Journal of Applied Physics*, vol. 43, no. 2, pp. 759–761, 1972.
- Dobratz, B. M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," *UCRL-52997, Lawrence Livermore National Laboratory, Livermore, California*, January 1981.
- Herrmann, W., "Constitutive Equation for the Dynamic Compaction of Ductile Porous Materials," *Journal of Applied Physics*, vol. 40, no. 6, pp. 2490–2499, 1968.
- Lee, E., M. Finger, and W. Collins, "JWL Equation of State Coefficients for High Explosives," *UCID-16189, Lawrence Livermore National Laboratory, Livermore, California*, January 1973.

- Wardlaw, A. B., R. McKeown, and H. Chen, “Implementation and Application of the $P - \alpha$ Equation of State in the DYSMAS Code,” *Naval Surface Warfare Center, Dahlgren Division, Report Number: NSWCDD/TR-95/107*, May 1996.

Multiscale Properties

In this section:

- *Multiscale Material Modeling*
- *Mean-Field Homogenization*

Multiscale Material Modeling

Overview

The multiscale material model in Abaqus uses the mean-field homogenization approach to capture the behavior of heterogeneous materials that contain a matrix material and one or more inclusion materials.

The goal of multiscale material modeling is to predict material behaviors at the macro-level using information at smaller scales. The information can be communicated between scales from the bottom up; the microstructure of the composite material is represented by a Representative Volume Element (RVE), then a homogenization scheme is applied to the RVE, which provides a prediction of macro-constitutive behavior at a material point, as shown in *Figure 1*.

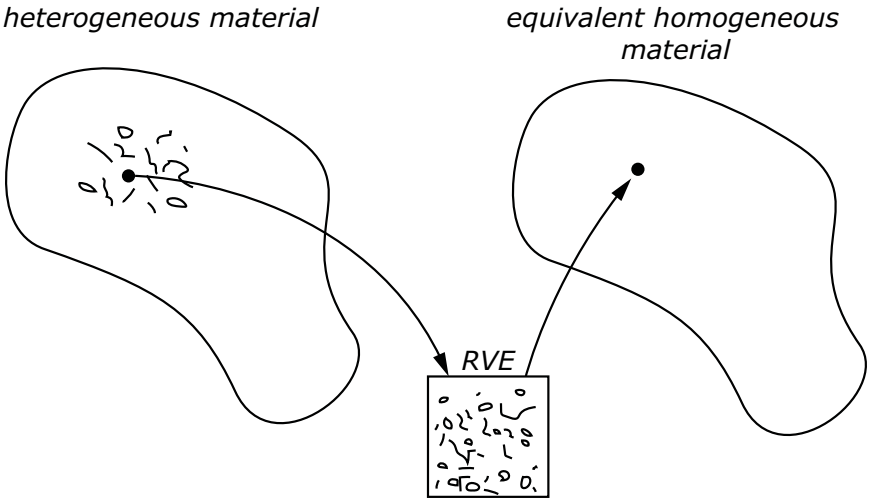


Figure 1: Homogenization of a heterogeneous material.

The information can also be communicated from a macro scale down to smaller scales by applying the macro-strain as a far-field strain to the RVE. The state of the constituents on the micro-level, such as strain and stress, are then updated, and the averaged quantities are communicated back to the material point at the macro-level.

Mean-field approaches such as Mori-Tanaka can provide analytical solutions of the averaged stresses and strains in each constituent of the composite. The averaged strain of each constituent is determined based on the specific homogenization method, which is based on assumptions of the interaction between inclusions and the matrix. Since only volume-averaged quantities are considered, this approach is unable to capture local stress concentration inside the RVE and the size effects of the inclusions; however, it is computationally efficient compared to the FE-RVE approach in which the RVE is modeled with a finite element model.

The mean-field homogenization approach gives good predictions for linear elastic composites. Since the homogenization is performed during the analysis, nonlinear and history-dependent behavior can be captured. However, in the case of highly nonlinear constituent materials, calibration is usually necessary for better predictions. If experimental results are not available, the results of an FE-RVE that represent details of the microstructure can be used as an alternative for calibration.

References:

Mean-Field Homogenization

Products: Abaqus/Standard Abaqus/Explicit

References:

- *Multiscale Material Modeling*
- **MEAN FIELD HOMOGENIZATION*

Overview

Mean-field homogenization:

- is a semianalytical homogenization approach that can be used to compute the thermal and mechanical properties of a composite material; and
- can also predict volume-averaged microlevel solutions in each constituent of the composite.

Specifying Constituent Properties for a Composite Material

In multiscale material modeling, each composite material consists of one matrix material and one or more inclusions including voids. In Abaqus/Explicit, the composite material must consist of one matrix material and only one inclusion; the inclusion material must be linear elastic with constant properties, and the matrix material can be only linear elastic or elastic-plastic with isotropic hardening.

Input File Usage: Use the following option to define the properties of the matrix in the composite:

**CONSTITUENT*, TYPE=MATRIX

Use the following option to define the properties of a nonvoid inclusion in the composite:

**CONSTITUENT*, TYPE=INCLUSION

Use the following option to define the properties of a void inclusion in the composite:

**CONSTITUENT*, TYPE=VOID

Mean-Field Homogenization for Linear Elastic Composites

The mean-field strain and stress in each phase I is defined as

$$\overline{\sigma}_I = \langle \sigma \rangle_I$$
$$\overline{\epsilon}_I = \langle \epsilon \rangle_I,$$

where

$$\langle f \rangle = \frac{1}{V} \int_V f dV.$$

The averaged macro fields can be written as

$$\langle \bullet \rangle = v_f^M \langle \bullet \rangle_M + \sum_{I=1}^{N} v_f^I \langle \bullet \rangle_I,$$

where v_f^M is the volume fraction of the matrix phase and v_f^I is the volume fraction of the inclusion phase. For the single-inclusion case, if all the constituents are linear elastic, the strain in the inclusion is related to the strain in the matrix through a concentration tensor A.

$$\langle \epsilon \rangle_I = A \langle \epsilon \rangle_M .$$

The strain and stress in the inclusion/matrix can also be related to the macro strain/stress through the strain/stress concentration tensor $B^{\epsilon}_{I,M}$ and $B^{\sigma}_{I,M}$:

$$\langle \epsilon \rangle_{I,M} = B^{\epsilon}_{I,M} \langle \epsilon \rangle .$$
$$\langle \sigma \rangle_{I,M} = B^{\sigma}_{I,M} \langle \sigma \rangle .$$

The form of the concentration tensor defines the homogenization methods below.

Voigt and Reuss Models

The simplest mean-field homogenization formulations are the Voigt and Reuss models. These models do not take into account the shape or the orientation of the inclusions; however, they provide upper and lower bounds of the macro stiffness modulus and, therefore, can be used for validation. The Voigt model is also used in a two-step approach to model composites with multiple inclusions.

The Voigt model assumes uniform strain in the composite $\langle \epsilon \rangle_I = \langle \epsilon \rangle_M$, which gives

$$A = I.$$

The Reuss model assumes uniform stress in the composite $\langle \sigma \rangle_I = \langle \sigma \rangle_M$, which gives

$$A = C_I^{-1} C_M ,$$

where C_M is the stiffness of the matrix and C_I is the stiffness of the inclusion.

Input File Usage: Use the following option to specify the Voigt model as the homogenization method:

**MEAN FIELD HOMOGENIZATION*, FORMULATION=VOIGT

Use the following option to specify the Reuss model as the homogenization method:

**MEAN FIELD HOMOGENIZATION*, FORMULATION=REUSS

Micromechanical Models for Unidirectional Composites

The Chamis model (*Chamis, 1989*) and Composite Cylinder Assemblage (CCA) model (*Christensen, 1979*) are micromechanical homogenization models developed and validated for unidirectional composites.

With the composite stiffness known from the micromechanical models, the concentration tensor can be computed using

$$A = \frac{v_f^M}{v_f^I} (C_C - C_I)^{-1} \Big(C_C - C_M \Big),$$

where C_C is the composite stiffness, C_M is the stiffness of the matrix, and C_I is the stiffness of the inclusion.

Input File Usage: Use the following option to specify the Chamis model as the homogenization method:

**MEAN FIELD HOMOGENIZATION*, FORMULATION=CHAMIS

Use the following option to specify the CCA model as the homogenization method:

**MEAN FIELD HOMOGENIZATION*, FORMULATION=CCA

Eshelby-Based Models

The more sophisticated mean-field homogenizations are based on Eshelby's solution. In his 1957 paper (*Eshelby, 1957*), Eshelby solved a single-inclusion problem described in *Figure 1*.

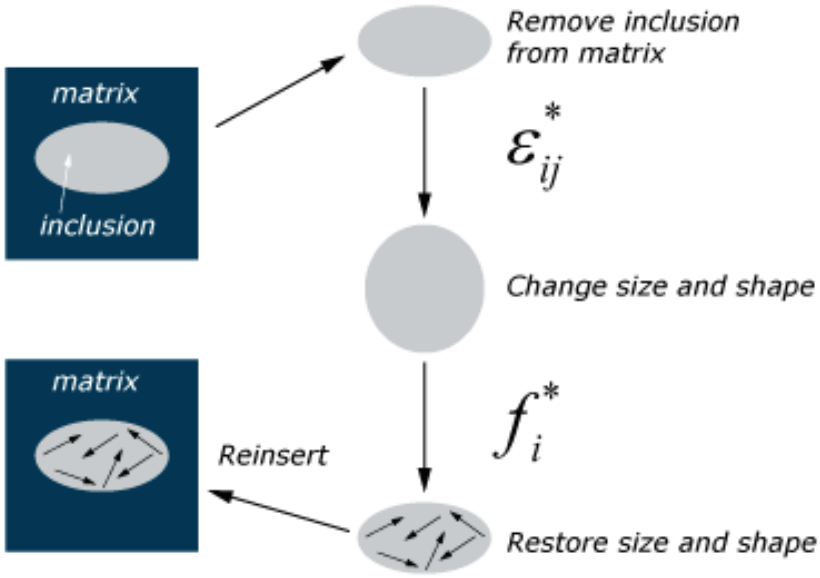


Figure 1: Eshelby's problem: an ellipsoid is cut out of an infinite matrix, undergoes an eigenstrain, and is inserted back into the matrix.

Eshelby found that the strain inside the constrained ellipsoid is uniform and given by

$$\epsilon(x) = E : \epsilon^*,$$

where x represents a point inside the inclusion and ϵ^* is the eigenstrain. The Eshelby tensor (E) is a function of the stiffness of the matrix and also the shape and orientation of the inclusion. To calculate Eshelby's tensor usually involves numerical integration over the surface of the ellipsoidal inclusion; however, analytical solutions are available (*Mura, 1987*) for simple shapes such as a sphere, cylinder, penny, oblate, and prolate. Based on Eshelby's solution, the following models are proposed based on different assumptions about the properties of the outer material surrounding the inclusions.

Mori-Tanaka Model

The Mori-Tanaka formulation assumes that each inclusion behaves like an isolated inclusion and the strain in the matrix is considered as the far-field strain. Therefore, the concentration tensor can be written as

$$A = \left[E : \left(C_M^{-1} C_I - I\right) + I\right]^{-1}.$$

Input File Usage: **MEAN FIELD HOMOGENIZATION*, FORMULATION=MT

Inverse Mori-Tanaka Model

The inverse Mori-Tanaka formulation assumes a high volume fraction of the inclusion and permutes the properties of the matrix and inclusion in a single inclusion problem, which gives

$$\boldsymbol{A} = \left[\boldsymbol{E} : \left(\boldsymbol{C}_I^{-1}\boldsymbol{C}_M - \boldsymbol{I}\right) + \boldsymbol{I}\right]^{-1}.$$

Input File Usage: **MEAN FIELD HOMOGENIZATION*, FORMULATION=INVERSED MT

Balanced Model

The Mori-Tanaka and inverse Mori-Tanaka formulations give the upper and lower bounds of the composite stiffness for low and high volume fractions. The balanced formulation is the interpolative formulation, which can be written as

$$\boldsymbol{H}_1 = \left[\boldsymbol{E} : \left(\boldsymbol{C}_M^{-1}\boldsymbol{C}_I - \boldsymbol{I}\right) + \boldsymbol{I}\right]^{-1},$$

$$\boldsymbol{H}_2 = \left[\boldsymbol{E} : \left(\boldsymbol{C}_I^{-1}\boldsymbol{C}_M - \boldsymbol{I}\right) + \boldsymbol{I}\right]^{-1},$$

$$\boldsymbol{A} = \left[\left(1-\varphi\right)\boldsymbol{H}_1^{-1} + \varphi\boldsymbol{H}_2^{-1}\right]^{-1},$$

where φ is the smooth interpolation function proposed by Lielens (*1999*)

$$\varphi = \tfrac{1}{2}v_f^I\left(1 + v_f^I\right).$$

Input File Usage: **MEAN FIELD HOMOGENIZATION*, FORMULATION=BALANCED

User-Defined Concentration Tensor

You can define the concentration tensor $\boldsymbol{A}$ directly by specifying all of its components. The user-defined concentration tensor must be defined in the local coordinate system of the inclusion. You can use homogenization models other than those listed above, calibration from experimental results, or FE-RVE models to obtain the value of the concentration tensor.

Input File Usage: Use the following option to define the fourth-order strain concentration tensor used to calculate the strain in a constituent:

**CONCENTRATION TENSOR*, TYPE=STRAIN (default)

Use the following option to define the second-order conductivity concentration tensor used to calculate the temperature gradient in a constituent:

**CONCENTRATION TENSOR*, TYPE=CONDUCTIVITY

Mean-Field Homogenization for Thermal Properties

Mean-field homogenization can also be used to compute effective thermal properties and micro-level thermal fields of the constituents.

Conductivity

The effective conductivity of composite materials can also be computed using Eshelby's single-inclusion method described in *Mean-Field Homogenization for Linear Elastic Composites*. For example, Mori-Tanaka method gives the following conductivity concentration tensor

$$A^C = \left[E^C : \left(k_M^{-1} k_I - I\right) + I\right]^{-1}.$$

where E^C is the modified 3x3 second-order Eshelby's tensor and $k_{I,M}$ are second-order conductivity tensor of the constituents.

The relationship between the temperature gradient in the constituents is given by

$$\langle \nabla\theta \rangle_I = A^C \langle \nabla\theta \rangle_M.$$

and the macro-level temperature gradient is the volume-averaged temperature gradient of the constituents. Chamis and CCA formulations are not supported for conductivity.

Specific Heat and Latent Heat

The specific heat and latent heat of the composite can be compute using the mass-averaged value of the constituents.

Incremental Mean-Field Homogenization for Nonlinear Composites

An incremental linearization approach is used to extend the homogenization models to composites with nonlinear behavior. The tangent stiffness matrix is used in place of the linear elastic modulus to compute the concentration tensor A

$$\Delta\epsilon_I = A\Delta\epsilon_M.$$

Since the tangent matrix is not constant in the nonlinear case and is usually a function of the strain, iterations are performed to guarantee a converged solution of the strain increment in each constituent.

Isotropization Methods

Analytical expression of Eshelby's tensor is available only if the inclusions are ellipsoidal and the matrix material is isotropic or transversely isotropic. For matrix materials that are anisotropic, using the stiffness directly to compute the Eshelby's tensor can result in stiffer prediction for the composite behavior (*Doghri, 2003*). The isotropic part of the tangent stiffness matrix can generally be written as

$$C^{iso} = 3KI^{vol} + 2GI^{dev},$$

where K is the bulk modulus and G is the shear modulus. I^{vol} and I^{dev} are the volumetric and deviatoric part of the fourth-order identity tensor. The method used to extract the isotropic part of the stiffness matrix is not unique, and the accuracy of the prediction varies depending on the properties of the matrix material.

General Method

The general method is the most common isotropization method, and it extracts the isotropic part using

$$K = \frac{1}{3} I^{vol} :: C^{ani},$$

$$G = \frac{1}{10} I^{dev} :: C^{ani}.$$

The general method can be used for any anisotropic tangent matrix. This method is the default isotropization method.

Input File Usage: **CONSTITUENT*

Spectral Method

The spectral isotropization method can be used if isotropic hardening is defined in the matrix material (*Doghri, 2003*). The isotropic part of the tangent matrix is given by

$$K = K_e,$$

$$G = G_e \left(1 - \frac{3G_e}{3G_e + \frac{d\sigma^0}{d\bar{\epsilon}^{pl}}\left(\bar{\epsilon}^{pl}\right)} \right),$$

where K_e is the elastic bulk modulus, G_e is the elastic shear modulus, $\sigma^0\left(\bar{\epsilon}^{pl}\right)$ is the yield stress, and $\bar{\epsilon}^{pl}$ is the equivalent plastic strain.

Input File Usage: **CONSTITUENT*, ISOTROPIZATION COEFFICIENT=1.0

Modified Spectral Method

For matrix materials with significant softening after yielding, both the general method and the spectral method might give predictions that are too stiff. The modified spectral method proposed by Selmi et. al. (*Selmi, 2011*) improves the accuracy of the spectral method by evaluating the derivative of the yield stress at a larger equivalent plastic strain

$$K = K_e,$$

$$G = k_G G_e \left(1 - \frac{3G_e}{3G_e + \frac{d\sigma^0}{d\bar{\epsilon}^{pl}}\left(k_p \bar{\epsilon}^{pl} + k_s\right)} \right),$$

where k_G, k_p, and k_s are the isotropization parameters. You can calibrate these parameters by comparing the mean-field solution to experimental data or to FE-RVE solutions. The isotropization parameters are irrelevant if the constituent does not have plastic behavior.

Input File Usage:

Use the following option when all the parameters are constant and k_p is the only parameter with a nondefault value:

**CONSTITUENT*, ISOTROPIZATION COEFFICIENT=*isotropization coefficient*

Use the following options to define temperature- or field-dependent isotropization parameters:

**CONSTITUENT*
**ISOTROPIZATION PARAMETERS*

Isotropic Projection

Options are available to use the isotropic projection of the tangent matrix to compute different parts of the concentration tensor. For better predictions, Doghri et al. (*Doghri, 2010*) recommended the following:

- If the general method is used, only Eshelby's tensor is computed with the isotropic projection.
- If the spectral/modified spectral method is used, Hill's tensor given by $P = E : C_M^{-1}$ is computed with the isotropic projection.

It is possible that some choices can lead to unphysical prediction of microlevel strain resulting in unstable material behavior in the composite. In addition, in Abaqus/Standard analyses, the tangent stiffness matrix for the composite might lose its symmetry after homogenization; therefore, you must use the unsymmetric solver to achieve convergence.

Input File Usage: Use the following option to use the isotropic projection of the matrix stiffness to compute all parts of the concentration tensor by default:

**MEAN FIELD HOMOGENIZATION*, ISOTROPIZATION=ALLISO

Use the following option to use the isotropic projection of the matrix stiffness to compute only Eshelby's tensor:

**MEAN FIELD HOMOGENIZATION*, ISOTROPIZATION=E-ISO

Use the following option to use the isotropic projection of the matrix stiffness to compute Hill's tensor:

**MEAN FIELD HOMOGENIZATION*, ISOTROPIZATION=P-ISO

Multistep Homogenization

For composites with multiple inclusions, a multistep homogenization approach is used, as shown in *Figure 2*. The composite is decomposed into "grains," with each grain containing one inclusion family and the matrix. The inclusions in each family have the same material properties, aspect ratio, and orientation. In the first step homogenization is performed in each grain using the user-specified formulation; in the second step, the Voigt formulation is used to compute the properties of the overall composite.

An alternative approach is to use the Mori-Tanaka scheme in both the first and the second step, assuming the average strain in the matrix is uniform across all grains. This approach is equivalent to the direct Mori-Tanaka approach proposed by Benveniste (*Benveniste et al., 1991*). In an Abaqus/Standard analysis, the second approach has the drawback that it might result in an unsymmetric effective modulus when the inclusions are misaligned and nonisotropic. In such cases, use of the unsymmetric solver might be necessary.

Abaqus/Explicit analyses only support one matrix and one inclusion material. For composites with dispersed fiber directions (see *Specifying a Second-Order Orientation Tensor in Abaqus/Explicit*), the direct Mori-Tanaka approach is used.

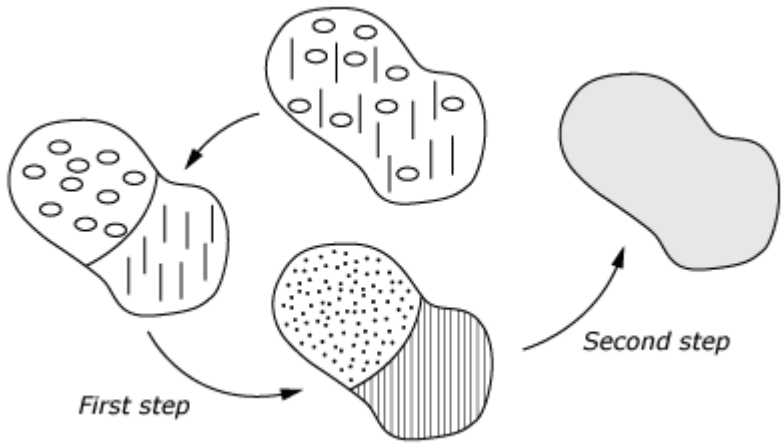


Figure 2: Multistep homogenization method for multiphase composites.

Input File Usage: Use the following option to use the Voigt formulation in the second step homogenization for the multiphase composite:

**MEAN FIELD HOMOGENIZATION*, UNIFORM MATRIX STRAIN=NO

Use the following option to use the direct Mori-Tanaka formulation for the multiphase composite, assuming the average strain in the matrix is uniform across all grains:

**MEAN FIELD HOMOGENIZATION*, UNIFORM MATRIX STRAIN=YES

Specifying the Microstructure of the Composite

You can specify the microstructure of the composite by giving the shape, volume fraction, aspect ratio, and fiber orientation of each constituent. You can use a distribution to define the volume fraction, aspect ratio, and the orientation tensor (see *Distribution Definition*).

The orientation tensor and random fiber orientation can be used only with three-dimensional solid elements.

Specifying the Shape, Volume Fraction, and Aspect Ratio

The volume fraction and aspect ratio can be specified as spatial distributions. The aspect ratio is given by L/D as shown in *Figure 3*.

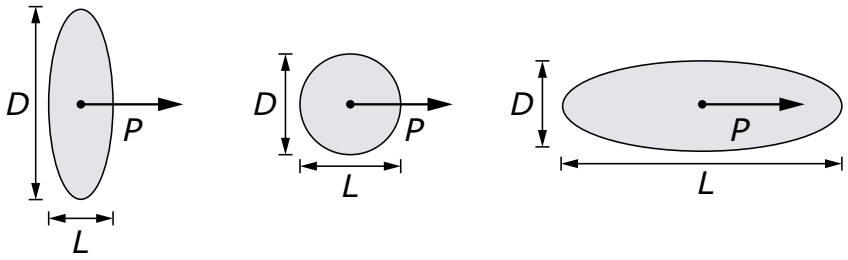


Figure 3: Different shapes of the inclusion.

The aspect ratio of a spherical inclusion is 1.0, and the aspect ratio of a cylindrical inclusion is infinity; therefore, you do not need to specify the aspect ratio in these cases.

Input File Usage: Use the following option to specify a prolate-shaped inclusion ($L > D$):

**CONSTITUENT*, SHAPE=PROLATE
volume fraction, *aspect ratio*,

Use the following option to specify an oblate-shaped inclusion ($L < D$):

**CONSTITUENT*, SHAPE=OBLATE
volume fraction, *aspect ratio*,

Use the following option to specify a penny-shaped inclusion ($L \ll D$):

**CONSTITUENT*, SHAPE=PENNY
volume fraction, *aspect ratio*,

Use the following option to specify a spherical inclusion:

*CONSTITUENT, SHAPE=SPHERE
volume fraction,

Use the following option to specify a continuous cylindrical inclusion with a circular cross-section:

*CONSTITUENT, SHAPE=CYLINDER
volume fraction,

Use the following option to specify a continuous cylindrical inclusion with an elliptical cross-section:

*CONSTITUENT, SHAPE=ELLIPTIC CYLINDER

volume fraction, a_2/a_3

Here a_2 and a_3 are the radii of the elliptical cross-section along the axis in the local 2- and 3-directions.

Specifying a Fixed Orientation for the Inclusion and Void Phases

By default, the orientation of the inclusion or void phase is fixed and is aligned with the x-axis of the local coordinate system specified in the section definition. You can specify a different orientation by giving the three components of the orientation vector in the local coordinate system, as shown in *Figure 4*. The components of the orientation unit vector, $\boldsymbol{p}$, are

$$\begin{aligned} p_1 &= \sin\theta\cos\varphi, \\ p_2 &= \sin\theta\sin\varphi, \\ p_3 &= \cos\theta. \end{aligned}$$

The local 3-direction of the inclusion (void included) is the projection of the local z-axis defined in the section onto the surface normal to the inclusion orientation (1-direction). If the local z-axis is within 0.1 degree of being parallel to the 1-direction, the local 3-direction is the projection of the local y-axis onto the surface. The local 2-direction is then at right angles to the local 3-direction, so that the local 2-direction, local 3-direction, and the inclusion orientation (1-direction) form a right-handed set. The local 2 and 3 directions are relevant only if the shape of the inclusion (void included) is an elliptical cylinder.

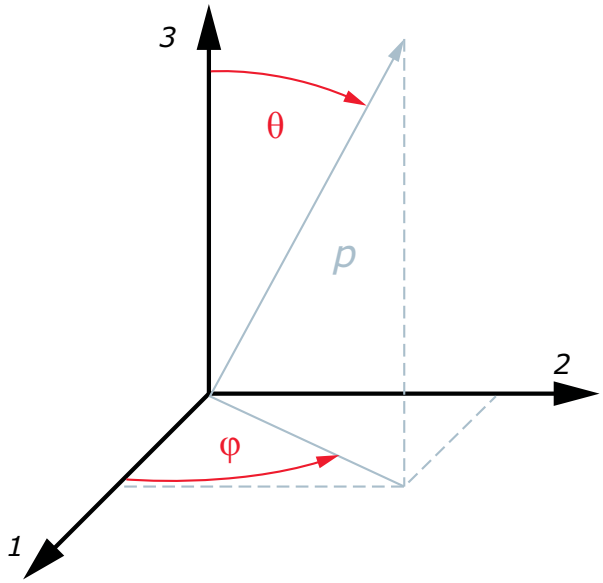


Figure 4: Orientation of the inclusion (void included).

Input File Usage: Use the following option to specify a fixed fiber orientation:

**CONSTITUENT*, DIRECTION=FIXED

, , p_1, p_2, p_3

Specifying a Second-Order Orientation Tensor in Abaqus/Standard

For composites containing short fibers, a second-order orientation tensor, $\boldsymbol{a}$, is usually used to describe the dispersion of the fiber orientation

$$a_{ij} = \oint p_i p_j \Psi(p)\, dp,$$

where $\oint dp = \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} \sin\theta d\theta d\varphi$ is the integral over the surface of the unit sphere (all possible directions of p) and $\Psi(p)$ is the orientation distribution function (ODF). The spatial distribution of the orientation tensor is usually given by injection molding software and can be used directly to compute the effective moduli of the composite material that is linear elastic. If the material is nonlinear, the average fields given by

$\langle f \rangle = \oint f(p)\Psi(p)dp$ must be computed during the analysis, and the ODF is required to calculate the integral. When only the second-order orientation is available, Abaqus recovers the ODF using the approach proposed by *Onat and Leckie (1988)*

$$\Psi(p) \simeq \frac{1}{4\pi} + \frac{15}{8\pi} b_{ij} f_{ij}(p) + \frac{315}{32\pi} B_{ijkl} F_{ijkl}(p),$$

for three-dimensional orientations, where f_{ij} and F_{ijkl} are the tensor basis functions

$$f_{ij}(p) = p_i p_j - \frac{1}{3}\delta_{ij},$$

$$F_{ijkl} = p_i p_j p_k p_l - \frac{1}{7}\left(p_i p_j \delta_{kl} + p_i p_k\, \delta_{jl} + p_i p_l \delta_{jk} + p_k p_l \delta_{ij} + p_j p_l \delta_{ik} + p_j p_k \delta_{il}\right) + \frac{1}{35}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}),$$

and b_{ij} and B_{ijkl} are the deviatoric versions of the orientation tensors

$$b_{ij} = a_{ij} - \frac{1}{3}\delta_{ij},$$

$$B_{ijkl} = A_{ijkl} - \frac{1}{7}\Big(a_{ij}\delta_{kl} + a_{ik}\delta_{jl} + a_{il}\delta_{jk} + a_{kl}\delta_{ij} + a_{jl}\delta_{ik} + a_{jk}\delta_{il}\Big) + \frac{1}{35}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}).$$

The ODF reconstructed using the approach above can have negative values. When Abaqus detects negative values, it switches to the von Mises distribution function to reconstruct the ODF. However, the algorithm that computes the parameters of the von Mises distribution can fail to converge when the fibers are almost fully aligned. Abaqus issues a warning message in this case. Alternatively, you can specify that Abaqus construct the ODF using the original approach proposed by *Onat and Leckie (1988)* without additional adjustment.

The fourth-order orientation tensor is required during the ODF reconstruction. It is computed using a closure approximation, which is a formula that approximates the fourth-order tensor in terms of the known second-order tensor. A popular closure approximation is the hybrid closure approximation proposed by Advani and Tucker (*1987*), which is constructed using the interpolation of linear and quadratic closure approximation

$$A^h_{ijkl} = \Big(1 - \alpha\Big)A^l_{ijkl} + \alpha A^q_{ijkl},$$

where the linear closure is given by

$$A^l_{ijkl} = -\frac{1}{35}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \frac{1}{7}(a_{ij}\delta_{kl} + a_{ik}\delta_{jl} + a_{il}\delta_{jk} + a_{kl}\delta_{ij} + a_{jl}\delta_{ik} + a_{jk}\delta_{il}),$$

and the quadratic closure is given by

$$A^q_{ijkl} = a_{ij}a_{kl}.$$

The weight function is defined as $\alpha = 1 - 27\det(a)$ for three-dimensional orientations.

Other available closure approximations are the orthotropic smooth approximation (*Advani and Tucker, 1987*), the orthotropic fitted (ORF) approximation (*Cintra and Tucker, 1995*), and the orthotropic equivalent of the natural (ORE) approximation (*VerWeyst, 1998*).

Only one constituent in the composite is allowed to have orientations specified with the second-order orientation tensor.

Taking into account the symmetry of the ODF$\Psi(p) = \Psi(-p)$, we compute the integral $\oint f(p)\Psi(p)dp$ numerically as follows:

$$\oint f(p)\Psi(p)\,dp \simeq 2\sum_{I=1}^{N}\sum_{J=1}^{N} f(\theta_I, \varphi_J)\,\Psi(\theta_I, \varphi_J)\sin(\theta_I)\Delta\theta\Delta\varphi,$$

where $\Delta\theta = \Delta\varphi = \frac{\pi}{N}$ and $\theta_I = I\Delta\theta, \varphi_J = J\Delta\varphi$. The numerical integral is computed by subdividing the macro point into $N \times N$ subdomains, with each subdomain having the same fiber orientation $p = p(\theta_I, \varphi_J)$. Then a multistep homogenization (as described in *Figure 2*) is used to compute the macrolevel solutions. The ODF is more accurately recovered if a large value of *N* is used; however, the computational cost and memory usage also increases substantially.

When the fiber direction is random in the three-dimensional space, the second-order orientation tensor is

$a_{ij} = \frac{1}{3}\,\delta_{ij}$, and the ODF is constant across all subdomains.

When the direct Mori-Tanaka homogenization approach is used, the fiber response can be calculated either individually in each subdomain or just once using the averaged state over all subdomains. In both cases only the averaged fiber state is available for output. The maximum principal direction of the second-order orientation tensor is used as the 1-direction of the local coordinate system for the inclusion.

In composites where the fiber material is linear elastic with uniform moduli across all pseudo-grains, if we assume uniform matrix strain inside the composite, homogenization and strain partitioning can be simplified.

Instead of computing the integrals numerically, Abaqus/Standard uses the following equation to compute the orientation averaging terms, $C^{avg} = \oint C(p)\,\Psi(p)\,dp$:

$$C^{avg}_{ijkl} = C_1 a_{ijkl} + C_2\left(a_{ij}\delta_{kl} + a_{kl}\delta_{ij}\right) + C_3\left(a_{ik}\delta_{jl} + a_{il}\delta_{jk} + a_{jl}\delta_{ik} + a_{jk}\delta_{il}\right) + C_4\delta_{ij}\delta_{kl} + C_5\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right),$$

where C_1, C_2, C_3, C_4, and C_5 are the five constants of the tensor that is transversely isotropic. ODF reconstruction is not necessary and is not performed.

Input File Usage: Use the following option to specify the fiber orientation using a second-order tensor:

**CONSTITUENT*, DIRECTION=ORIENTATION TENSOR
, , a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23}

Use the following option to specify a random fiber orientation:

**CONSTITUENT*, DIRECTION=RANDOM3D

Use the following option to specify the number of subdivisions used to compute the surface integral:

**MEAN FIELD HOMOGENIZATION*, ANGLE SUBDIVISIONS=*N*

Use the following option to construct the ODF using the original approach proposed by *Onat and Leckie (1988)* without additional adjustment:

**MEAN FIELD HOMOGENIZATION*, ODF ADJUST=NO

Use the following option to compute the fiber response individually in each subdomain:

**CONSTITUENT*, RESPONSE=GRANULAR (default)

Use the following options to compute the average response of the fiber directly when using the direct Mori-Tanaka homogenization approach:

**MEAN FIELD HOMOGENIZATION*, UNIFORM MATRIX STRAIN=YES
**CONSTITUENT*, RESPONSE=AVERAGE

Use the following option to compute the ODFs using the orthotropic fitted (ORF) closure approximation:

**MEAN FIELD HOMOGENIZATION*, CLOSURE=FITTED

Use the following option to compute the ODFs using the orthotropic equivalent of the natural (ORE) closure approximation:

**MEAN FIELD HOMOGENIZATION*, CLOSURE=ORE

Use the following option to compute the ODFs using the hybrid closure approximation:

**MEAN FIELD HOMOGENIZATION*, CLOSURE=HYBRID

Use the following option to compute the ODFs using the quadratic closure approximation:

**MEAN FIELD HOMOGENIZATION*, CLOSURE=QUADRATIC

Use the following option to compute the ODFs using the orthotropic smooth closure approximation:

**MEAN FIELD HOMOGENIZATION*, CLOSURE=SMOOTH

Use the following option to compute the orientation averaging directly:

**MEAN FIELD HOMOGENIZATION*, ORIENTATION AVERAGING=DIRECT, UNIFORM MATRIX STRAIN=YES

Specifying a Second-Order Orientation Tensor in Abaqus/Explicit

In Abaqus/Explicit, when a second-order orientation tensor, $\boldsymbol{a}$, is used to specify the dispersion of the fiber material, the direct Mori-Tanaka approach is used. Because the fiber material can be only linear elastic with uniform moduli across all pseudo-grains, ODF reconstruction is not performed and orientation averaging terms are computed directly.

The second-order orientation tensor is specified in the orientation definition of the section where the multiscale material is assigned.

Input File Usage: Use the following option to specify the fiber orientation using a second-order tensor:

**CONSTITUENT*, DIRECTION=ORIENTATION TENSOR

Use the following option to specify the components of a second-order tensor in the orientation definition:

**ORIENTATION*, DISPERSION

Composites with Thermal Expansion

Composites with thermal expansion are supported only in Abaqus/Standard.

The effective thermal expansion coefficient tensor can be computed using Levin's theorem (*2*):

$$\{\alpha\} = \sum_{I=1}^{N} v_f^I \alpha_I B_I^\sigma + v_f^M \alpha_M B_M^\sigma,$$

where $\boldsymbol{\alpha_{I,M}}$ are the thermal expansion coefficients of the constituents, $\boldsymbol{N}$ is the total number of inclusions, and $\boldsymbol{B_{I,M}^\sigma}$ are the stress concentration tensors defined in *Mean-Field Homogenization for Linear Elastic Composites*.

In the presence of thermal strains due to a temperature change, the averaged strain/stress in each constituent can be written as:

$$\{\epsilon\}_{I,M} = B_{I,M}^\epsilon \{\epsilon\} + B_{I,M}^T \Delta T.$$

The relationship between $\boldsymbol{B_{I,M}^\epsilon}$ and $\boldsymbol{B_{I,M}^T}$ can be established through superposition of the two problems to achieve the condition $\{\epsilon\} = \mathbf{0}$ with $\boldsymbol{\Delta T \neq 0}$ (*3*).

Problem I solves a uniform stress field $\{\sigma\}_I = \{\sigma\}_M = \{\sigma\}$ under the condition $\boldsymbol{\Delta T \neq 0}$ while assuming $\epsilon' = \{\epsilon'\}_I = \{\epsilon'\}_M$. The solution to problem I gives

$$\epsilon' = (C_I - C_M)^{-1}(\beta_I - \beta_M)\Delta T,$$

where $\beta_{I,M} = C_{I,M}\alpha_{I,M}$.

Problem II is purely mechanical assuming $\Delta T = 0$. Solving $\langle \epsilon \rangle = 0$ gives the solution

$$\langle \epsilon'' \rangle_{I,M} = -B^{\epsilon}_{I,M} \epsilon'.$$

By superimposing the constituent strain solutions from problem I and problem II, we get

$$\langle \epsilon \rangle_{I,M} = \langle \epsilon' \rangle_{I,M} + \langle \epsilon'' \rangle_{I,M} = \left(I - B^{\epsilon}_{I,M}\right) (C_I - C_M)^{-1} (\beta_I - \beta_M) \Delta T,$$

which results in the relationship between $B^{T}_{I,M}$ and $B^{\epsilon}_{I,M}$ below

$$B^{T}_{I,M} = \left(I - B^{\epsilon}_{I,M}\right) (C_I - C_M)^{-1} (\beta_I - \beta_M).$$

Thermal expansion is supported only for plane strain and three-dimensional solid elements.

Composites with Viscoelasticity

Composites with viscoelasticity are supported only in Abaqus/Standard.

When one or more constituents have viscoelasticity defined in the time domain, the homogenization approach described above is no longer appropriate because it does not take into account the relaxation of stress with time. However, similar homogenization can be performed in the frequency domain, and homogenized properties of the composite in the time domain can be computed using homogenized properties at different frequencies. This homogenization approach is applicable only when the viscoelasticity in the constituents is defined in the time domain using a Prony series expansion and includes the following steps:

1. The shear and bulk moduli of all the constituents are transformed to the frequency domain using Fourier transforms (see details in *Prony Series Parameters*).
2. The mean-field homogenization is performed in the frequency domain the same way as discussed above, with the exception that the moduli are complex numbers instead of real numbers. The real part of the composite modulus is the storage modulus, $\mathbf{C}^{s}$, and the imaginary part is the loss modulus, $\mathbf{C}^{l}$.
3. Assuming the time response of the composite can be written in a similar form as that in the constituents where viscoelasticity is defined using a Prony series expansion

   $$g_{ij}(t) = 1 - \sum_{k=1,N} g_k \left(1 - e^{-t/\tau_k}\right),$$

   where $g_{ij}(t) = C_{ij}(t) / C^{0}_{ij}$ and C^{0}_{ij} is the instantaneous elastic modulus of the composite, the homogenized storage modulus of the composite, $\mathbf{C}^{s}$, can be written in the frequency domain as follows:

   $$C^{s}_{ij}(\omega) = C^{0}_{ij} \left[1 - \sum_{k=1,N} \frac{g_k}{1+\tau_k^2 \omega^2}\right].$$

   The parameters τ_k are selected to be equally spaced numbers on a logarithmic scale in the range $[\tau_{\min}, \tau_{\max}]$. $\tau_{\min}$ and $\tau_{\max}$ are the minimum and maximum frequencies used to define the viscoelasticity in all constituents.

   The parameters g_k are computed through a least-squares curve fit at frequencies $\omega_l = 1/\tau_l \ (l = 1, N)$.

You can only specify the thermorheologically simple temperature effect in the matrix material. The composite is assumed to inherit the same temperature effect as the matrix material.

Other limitations include:

1. The inverse Mori-Tanaka and balanced homogenization formulations are not supported when you use the orientation tensor to define the direction of the inclusion.
2. When you use the orientation tensor to define the direction of the inclusion, the elasticity specified in the inclusion material must satisfy the transversely isotropic condition, and no other inclusions can be defined in the composite.
3. You cannot use the orientation tensor to define the direction of the inclusion in which viscoelasticity is defined.

4. Distributions other than distributions of the orientation tensor are not allowed in the material definition.
5. Temperature and field variable dependencies are allowed to define the elasticity of the matrix material only, and only one inclusion is allowed in this case.
6. Other than viscoelasticity, only the following properties are supported in the inclusion material:
   a. Conductivity
   b. Density
   c. Elasticity
   d. Thermal expansion
   e. Specific heat
7. Other than viscoelasticity, only the following properties are supported in the matrix material:
   a. Conductivity
   b. Density
   c. Elasticity
   d. Thermal expansion
   e. Specific heat
   f. Cure properties

Because the multiscale material is replaced by a composite with prehomogenized properties, the microlevel output is not available. Other than viscoelasticity, Abaqus/Standard also computes the homogenized values of all the other supported properties.

Composites Curing

Mean-field homogenization can be used to model composite behavior during a curing process (*Curing Processes in Polymers*). Similar to composites with viscoelasticity, a multiscale material is replaced by a composite with prehomogenized properties. The same limitations described in *Composites with Viscoelasticity* also apply.

The composite inherits the following properties from the matrix material definition:

- Cure kinetics
- Cure max conversion
- Cure glass transition temperature

Mean-field homogenization is used to compute the following cure properties for the composite:

- Cure heat generation. The specific heat of the reaction is computed using mass-averaged value of the constituents.
- Cure shrinkage. Levin's theorem is used to compute the effective thermal expansion coefficient (*Composites with Thermal Expansion*) and to compute the effective cure shrinkage coefficient of the composite.

Composites with Damage

When damage and failure are defined at the constituent level, the damage in each constituent contributes to the overall damage in an indirect way through stress averaging and strain partitioning. As the stiffness of the constituent decreases, the strain increment in this constituent is likely to increase based on the formulation of the homogenization. In reality, the damage behavior of each constituent is likely to interact with each other and affect the overall damage behavior of the composite; therefore, you might need to specify an additional damage variable, $\boldsymbol{D}$, for the composite. At any time during the analysis, the stress tensor of the composite is given by

$$\boldsymbol{\sigma} = (1 - D)\overline{\boldsymbol{\sigma}},$$

where $\overline{\boldsymbol{\sigma}} = \sum_i v_i^f \boldsymbol{\sigma}_i$, and $\boldsymbol{\sigma}_i = (1 - D_i)\overline{\boldsymbol{\sigma}}_i$ is the effective stress of the constituent.

You can choose to combine the damage variables, D_i, in all constituents to get the overall damage behavior of the composite. The overall damage variable, D, can be computed as the maximum of D_i:

$$D = \max\{D_i\},$$

or you can choose to combine the damage variables in a multiplicative sense:

$$D = 1 - \prod_i (1 - D_i).$$

You can also specify D using user subroutine *UDAMAGEMF*. In addition, you can specify a separate damage variable, d_i^{int}, for each constituent. This variable is used to account for the additional contribution to the overall damage behavior due to interaction (for example, a fiber material might buckle more easily if cracks start to develop in the matrix material) as follows:

$$\boldsymbol{\sigma} = (1 - D) \sum_i v_i^f \left(1 - d_i^{\text{int}}\right) \boldsymbol{\sigma}_i.$$

Input File Usage:

Use the following option to specify that D_i contributes to D in a maximum sense:

**MEAN FIELD DAMAGE*, DEGRADATION=MAXIMUM

Use the following option to specify that D_i contributes to D in a multiplicative sense:

**MEAN FIELD DAMAGE*, DEGRADATION=MULTIPLICATIVE

Use the following option to specify a user-defined damage behavior for the composite:

**MEAN FIELD DAMAGE*, DEGRADATION=USER

Abaqus does not support damage and failure defined at the composite level, with the exception of the general stress/strain-based damage initiation criteria in Abaqus/Standard (see *Generalized Damage Initiation Criteria for Anisotropic Materials*). Instead of computing the damage criterion using the composite-level stress/strain data, Abaqus/Standard evaluates the damage initiation criterion in each pseudo-grain using homogenized stress/strain, then computes an weighted average at the composite level written as

$$F = \sum_{k=1}^{N} w_k F_k$$

where F is the averaged damage initiation criterion or failure index, F_k is the damage initiation criterion or failure index in each pseudo-grain, and N is the number of pseudo-grains in the multiscale model. This feature is not supported when an averaged response of the fiber is computed in a chopped fiber-reinforced composite, because the stress/strain data are not available in each pseudo-grain but rather computed as averaged quantities.

Example

This simple example illustrates how to define a multiscale material.

The name of the multiscale material is `COMPOSITE`. The material consists of one inclusion material `MAT2` embedded in the matrix material `MAT1`. The shape of the inclusion is prolate. The Mori-Tanaka homogenization method is used. The inclusion has a volume fraction of v_f and an aspect ratio of ar. The direction of the inclusion is fixed and defined with a vector (p_1, p_2, p_3). The name of the matrix is `MATRIX_MAT1`, and the name of the inclusion is `FIBER_MAT2`.

Input file template

```
*MATERIAL, NAME=MAT1
…
*MATERIAL, NAME=MAT2
…
*MATERIAL, NAME=COMPOSITE
*MEAN FIELD HOMOGENIZATION, FORMULATION=MT
*CONSTITUENT, TYPE=MATRIX, MATERIAL=MAT1, NAME=MATRIX_MAT1
*CONSTITUENT, TYPE=INCLUSION, MATERIAL=MAT2, NAME=FIBER_MAT2,
 SHAPE=PROLATE, DIRECTION=FIXED
v_f, ar, p_1, p_2, p_3
```

Elements

In Abaqus/Standard the multiscale material model can be used with

- three-dimensional solid elements, including stress/displacement elements, diffusive heat transfer elements, and coupled temperature-displacement elements;
- two-dimensional solid elements (stress/displacement only);
- membrane elements;
- three-dimensional conventional shell elements (stress/displacement only); and
- continuum shell elements (stress/displacement only).

In Abaqus/Standard additional plane stress iterations are included for two-dimensional plane stress elements and shell elements.

In Abaqus/Explicit the multiscale material model can be used only with three-dimensional solid stress/displacement elements.

Hybrid elements and incompatible mode elements are not supported.

Output

By default, the output variables are macrolevel results. In addition, you can also request microlevel output. The name of the constituent is appended to the end of the output variable.

For composites reinforced with dispersed short fibers, the fiber output variables are computed in the local system with the local fiber direction as the maximum principal direction of the orientation tensor. The matrix output is available only if the direct Mori-Tanaka homogenization approach is used. Otherwise, only averaged stress and strain are available. If plasticity is specified in the matrix material, averaged plastic strains are also available. The same matrix output limitation also applies to composites reinforced with multiple fiber families.

Microlevel strain output is not available in Abaqus/Explicit.

Input File Usage: Use the following options to request field output for the constituents:

**OUTPUT*, FIELD
**ELEMENT OUTPUT*, MICROMECHANICS

Use the following options to request history output for the constituents:

**OUTPUT*, HISTORY
**ELEMENT OUTPUT*, MICROMECHANICS

References

- Advani, S., and C. Tucker, "The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites," *Journal of Rheology*, vol. 31 (8), pp. 751–784, 1987.
- Levin, V.M., "On the Coefficients of Thermal Expansion of Heterogeneous Materials," *Mechanics of Solids*, vol. 2, pp. 58–61, 1967.
- Benveniste, Y., and G. J. Dvorak, "On a Correspondance Between Mechanical and Thermal Effects in Two-Phase Composites," *Micromechanics and Inohomogeneity*, pp. 65–81, 1990.
- Benveniste, Y., G. J. Dvorak, and T. Chen, "On Diagonal and Elastic Symmetry of the Approximate Effective Stiffness Tensor of Heterogeneous Media," *Journal of the Mechanics and Physics of Solids*, vol. 39(7), pp. 927–946, 1991.
- Chamis, C.C., "Mechanics of Composite Materials: Past, Present and Future," *Journal of Composites Technology and Research*, vol. 11, pp. 3–14, 1989.
- Christensen, R.M., "Solutions for Effective Shear Properties in Three Phase Sphere and Cylinder Models," *Journal of the Mechanics and Physics of Solids*, vol. 27, pp. 315–330, 1979.
- Cintra, J., and C. Tucker, "Orthotropic Closure Approximations for Flow-Induced Fiber Orientation," *Journal of Rheology*, vol. 39, pp. 1095–1122, 1995.
- Doghri, I., I. Adam, and N. Bilger, "Mean-Field Homogenization of Elasto-Viscoplastic Composites Based on a General Incrementally Affine Linearization Method," *International Journal of Plasticity*, pp. 219–238, 2010.
- Doghri, I., and A. Ouaar, "Homogenization of Two-Phase Elasto-Plastic Composite Materials and Structures. Study of Tangent Operators, Cyclic Plasticity and Numerical Algorithms," *International Journal of Solids and Structures*, vol. 40(7), pp. 1681–1712, 2003.
- Eshelby, J. D., "The Determination of the Elastic Field of an Ellipsoidal Inclusion and Related Problems," *Proceedings of the Royal Society of London*, pp. 376–396, 1957.
- Kammoun, S., L. Brassart, G. Robert, I. Doghri, and L. Delannay, "Micromechanical Modeling of Short Glass-Fiber Reinforced Thermoplastics-Isotropic Damage of Pseudograins," *American Institute of Physics Conference Proceedings 1353*, pp. 974–977, 2011.
- Lielens, G., "Micro-Macro Modeling of Structured Materials," *PhD Thesis Universite Catholique de Louvain Belgium*, 1999.
- Mura, T., Micromechanics of Defects in Solids, MARTINUS NIJHOFF Publishers , 1987.
- Onat, E.T., and F.A. Leckie, "Representation of Mechanical Behavior in the Presence of Changing Internal Structure," *Journal of Applied Mechanics*, vol. 55(1), pp. 1–10, 1988.
- Pierard, O., and I. Doghri, "An Enhanced Affine Formulation and the Corresponding Numerical Algorithms for the Mean-Field Homogenization of Elasto-Viscoplastic Composites," *International Journal of Plasticity*, pp. 131–157, 2006.
- Selmi, A., I. Doghri, and L. Adam, "Micromechanical Simulations of Biaxial Yield, Hardening and Plastic Flow in Short Glass Fiber Reinforced Polyamide," *International Journal of Mechanical Sciences*, vol. 53(9), pp. 696–706, 2011.
- VerWeyst, B.E., "Numerical Predictions of Flow-Induced Fiber Orientation in 3-D Geometries," *Ph.D. thesis, University of Illinois at Urbana-Champaign*, 1998.

Other Material Properties

In this section:

- *Mechanical Properties*
- *Heat Transfer Properties*
- *Curing Processes in Polymers*
- *Acoustic Medium*
- *Poroelastic Acoustic Medium*
- *Mass Diffusion Properties*
- *Electromagnetic Properties*
- *Pore Fluid Flow Properties*
- *User Materials*

Mechanical Properties

In this section:

- *Fatigue Crack Growth Laws for Linear Elastic Materials*
- *Material Damping*
- *Thermal Expansion*
- *Field Expansion*
- *Pore Fluid Pressure*
- *Viscosity*
- *CZone Crush Stress*

Fatigue Crack Growth Laws for Linear Elastic Materials

Products: Abaqus/Standard Abaqus/CAE

References:

- *Linear Elastic Fatigue Crack Growth Analysis*
- **FATIGUE*
- **FRACTURE CRITERION*

Overview

A linear elastic fatigue crack growth criterion can be used to:

- model propagation of a discrete crack in a brittle material along an arbitrary, solution-dependent path, without any remeshing, and using the extended finite element method (XFEM);
- model fatigue crack growth along an arbitrary, solution-dependent path in a brittle material with the adaptive remeshing technique; and
- model progressive delamination growth along a predefined path at brittle material interfaces in laminated composites.

Linear Elastic Fatigue Crack Growth Behaviors

The onset and fatigue crack growth are both characterized by means of the Paris law, which relates either the onset or the rate of growth of a fatigue crack to the relative fracture energy release rate, as illustrated in *Figure 1*. The Paris law can also be expressed in terms of the equivalent relative stress intensity factor in some cases. An alternative form to the original Paris law, which better accounts for the mixed mode fatigue crack growth, and that depends on the maximum energy release rate (instead of the relative value) is also available. The alternative form is illustrated in *Figure 2*. The detailed equations for the different forms of the Paris law that are available in Abaqus are described later in this section.

The Paris regime is bounded by lower and upper limits, G_{thresh} and G_{pl}, respectively. The lower limit represents the energy release rate threshold, G_{thresh} (or the equivalent stress intensity factor threshold, K_{thresh}), below which there is no consideration of fatigue crack initiation or growth. The upper limit, G_{pl} (or the equivalent stress intensity factor limit, K_{pl}) , represents the energy release rate above which the fatigue crack grows at an accelerated rate. You can specify the ratio of $\frac{G_{thresh}}{G_C}$ and the ratio of $\frac{G_{pl}}{G_C}$ to define the lower and upper limits of the Paris regime. The quantity G_C in these ratios represents the critical equivalent strain energy release rate, calculated based on the pure mode fracture strengths of the bulk material and the user-specified mode-mix criterion. The default values of these bounds are $\frac{G_{thresh}}{G_C} = 0.01$ and $\frac{G_{pl}}{G_C} = 0.85$, respectively.

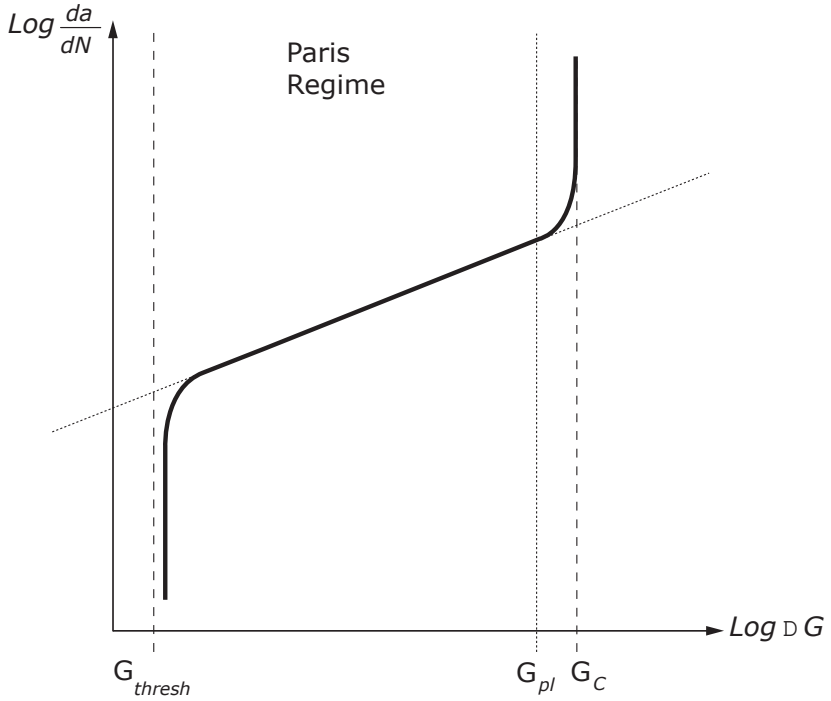


Figure 1: Fatigue crack growth governed by the Paris law.

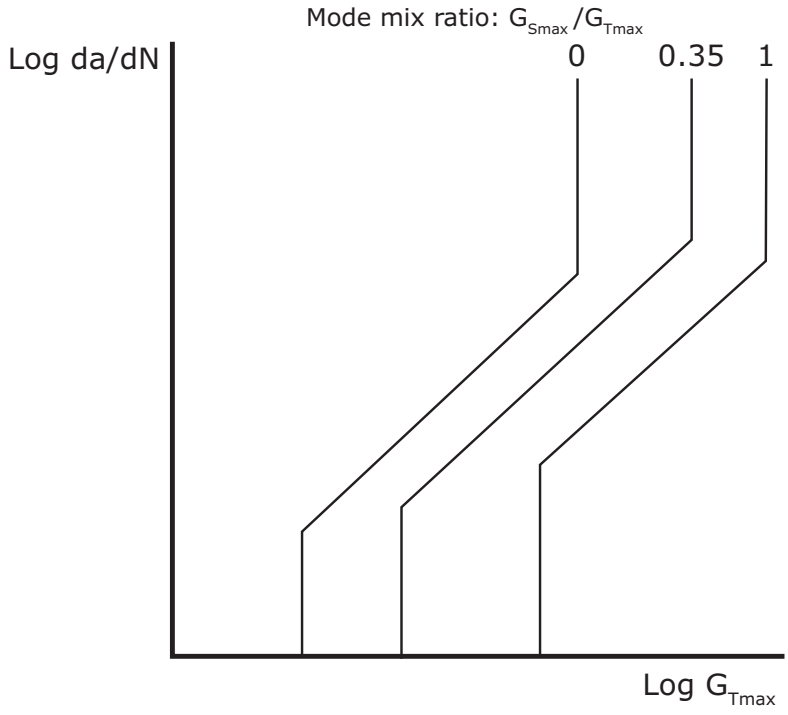


Figure 2: Mixed mode fatigue crack growth governed by the Paris law.

Input File Usage: Use the following option to define the fatigue onset and crack growth criterion:

**FRACTURE CRITERION*, TYPE=FATIGUE

Use the following option to define the PRX-fatigue onset and crack growth criterion:

**FRACTURE CRITERION*, TYPE=PRX-FATIGUE

Abaqus/CAE Usage: The onset and growth criterion for a fatigue crack are not supported in Abaqus/CAE.

Mixed-Mode Behavior

Abaqus provides three common mode-mix formulas for computing the equivalent fracture energy release rate, G_C: the BK law, the power law, and the Reeder law. The choice of model is not always clear in any given analysis; an appropriate model is best selected empirically.

BK Law

The BK law in Abaqus follows *Benzeggagh and Kenane (1996)* and is described by the following formula:

$$G_C = G_{IC} + (G_{IIC} - G_{IC}) \left(\frac{G_{II}+G_{III}}{G_I+G_{II}+G_{III}} \right)^{\eta}.$$

To define this model, you must provide G_{IC}, G_{IIC}, and η. This model provides a power law relationship combining the energy release rates in Mode I, Mode II, and Mode III into a single scalar fracture criterion.

Input File Usage: Use the following option to define the fatigue onset and crack growth criterion based on the original form of the Paris law and the BK law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=FATIGUE, MIXED MODE BEHAVIOR=BK

Use the following option to define the PRX-fatigue onset and crack growth criterion based on the original form of the Paris law and the BK law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=PRX-FATIGUE, MIXED MODE BEHAVIOR=BK

Abaqus/CAE Usage: Interaction module: **Create Interaction Property**: **Contact**, **Mechanical->Fracture Criterion**, **Type**: **Fatigue**, **Mixed mode behavior**: **BK**

Power Law

The power law model in Abaqus follows *Wu and Reuter (1965)* and is described by the following formula:

$$\frac{G_I+G_{II}+G_{III}}{G_C} = \left(\frac{G_I}{G_{IC}} \right)^{a_m} + \left(\frac{G_{II}}{G_{IIC}} \right)^{a_n} + \left(\frac{G_{III}}{G_{IIIC}} \right)^{a_o}.$$

To define this model, you must provide $G_{IC}, G_{IIC}, G_{IIIC}, a_m, a_n$, and a_o.

Input File Usage: Use the following option to define the fatigue onset and crack growth criterion based on original form of the Paris law and the POWER law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=FATIGUE, MIXED MODE BEHAVIOR=POWER

Use the following option to define the PRX-fatigue onset and crack growth criterion based on the original form of the Paris law and the POWER law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=PRX-FATIGUE, MIXED MODE BEHAVIOR=POWER

Abaqus/CAE Usage: Interaction module: **Create Interaction Property**: **Contact**, **Mechanical->Fracture Criterion**, **Type**: **Fatigue**, **Mixed mode behavior**: **Power**

Reeder Law

The Reeder law model in Abaqus follows *Reeder et al. (2002)* and is described by the following formula:

$$G_C = G_{IC} + (G_{IIC} - G_{IC}) \left(\frac{G_{II}+G_{III}}{G_I+G_{II}+G_{III}}\right)^{\eta} + (G_{IIIC} - G_{IIC}) \left(\frac{G_{III}}{G_{II}+G_{III}}\right) \left(\frac{G_{II}+G_{III}}{G_I+G_{II}+G_{III}}\right)^{\eta}.$$

To define this model, you must provide $G_{IC}, G_{IIC}, G_{IIIC}$, and η. The Reeder law is best applied when $G_{IIC} \neq G_{IIIC}$. When $G_{IIC} = G_{IIIC}$, the Reeder law reduces to the BK law. The Reeder law applies only to three-dimensional problems.

Input File Usage: Use the following option to define the fatigue onset and crack growth criterion based on the original form of the Paris law and the REEDER law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=FATIGUE, MIXED MODE BEHAVIOR=REEDER

Use the following options to define the PRX-fatigue onset and crack growth criterion based on the original form of the Paris law and the REEDER law to describe the mixed-mode behavior:

**FRACTURE CRITERION*, TYPE=PRX-FATIGUE, MIXED MODE BEHAVIOR=REEDER

Abaqus/CAE Usage: Interaction module: **Create Interaction Property**: **Contact**, **Mechanical->Fracture Criterion**, **Type**: **FATIGUE**, **Mixed mode behavior**: **Reeder**

Onset of Fatigue Crack Growth

The onset of fatigue crack growth is characterized by ΔG, which is the relative fracture energy release rate when the structure is loaded between its maximum and minimum values. The fatigue crack growth initiation criterion is defined as

$$f = \frac{N}{c_1 \Delta G^{c_2}} \geq 1.0,$$

where c_1 and c_2 are assumed to be material constants and N is the cycle number for onset. The crack front is not propagated unless the above equation is satisfied, and the maximum fracture energy release rate, $G_{\max}$ (which corresponds to the cyclic energy release rate when the structure is loaded up to its maximum value) is greater

than G_{thresh}. If you do not specify the onset criterion, Abaqus/Standard assumes that the onset of fatigue crack growth is satisfied automatically.

Fatigue Crack Growth Using the Paris Law

Once the onset criterion is satisfied at nodes along the crack front, the crack growth rate, da/dN, can be calculated based on the relative fracture energy release rate, ΔG. If $G_{thresh} < G_{max} < G_{pl}$,, the rate of crack growth per cycle is given by the Paris law:

$$\frac{da}{dN} = c_3 \Delta G^{c_4},$$

where c_3 and c_4 are assumed to be material constants.

Ratcliffe and Johnston (2014) and *Deobald et al. (2017)* proposed the following alternate form of the fatigue law, which better accounts for mixed-mode fatigue crack growth:

$$\frac{da}{dN} = c_3 G_{TMax}^{c_4}.$$

In the above expression, G_{TMax} is the total maximum strain energy release rate as opposed to the strain energy release rate range over a cycle, and the Paris law parameters, c_3 and c_4, are assumed to be material parameters that depend on mode-mix and stress ratios. Abaqus does not support the above form of the crack growth rate equation directly; instead, it allows you to specify da/dN as a tabular function of G_{TMax}, the mode-mix ratio, and the stress ratio.

In addition, you can utilize user subroutine *UMIXMODEFATIGUE* to implement a user-defined fatigue crack growth law.

For linear elastic materials, the fracture energy release rate is directly related to the stress intensity factors by the following relationship:

$$G = \frac{K_I^2}{E'} + \frac{K_{II}^2}{E'} + \frac{K_{III}^2}{2\mu},$$

where E' is the Young's modulus (which is equal to $\frac{E}{1-\nu^2}$ under plane strain and full three-dimensional conditions), μ is the shear modulus, and ν is the Poisson's ratio. A form based on the stress intensity factor, K that is equivalent to the fracture energy release rate–based form above, is also available:

$$\frac{da}{dN} = c_3 \Delta K_{eff}^{c_4},$$

where c_3 and c_4 are assumed to be material constants, and ΔK_{eff} is the effective stress intensity factor range of a load cycle.

Abaqus uses the *Irwin (1968)* definition of an effective stress intensity factor that is applicable under mixed-mode conditions and is given by:

$$\Delta K_{eff} = \sqrt{A * \Delta K_I^2 + B * \Delta K_{II}^2 + C * \frac{1}{1-\nu} \Delta K_{III}^2},$$

where *A*, *B*, and *C* are user-defined material constants.

For the fatigue crack growth criterion, the following forms based on the stress intensity factor are also available:

- A tabular form to support multiple piecewise linear log-log ($\frac{da}{dN}$ versus ΔK_{eff}) segments.
- A user-defined crack growth criterion using user subroutine *UMIXMODEFATIGUE*.

Input File Usage: Use one of the following options to define the fatigue onset and crack growth criterion based on the original form of the Paris law:

*FRACTURE CRITERION, TYPE=FATIGUE | PRX-FATIGUE, MIXED MODE BEHAVIOR=BK (default)

*FRACTURE CRITERION, TYPE=FATIGUE | PRX-FATIGUE, MIXED MODE BEHAVIOR=POWER

*FRACTURE CRITERION, TYPE=FATIGUE | PRX-FATIGUE, MIXED MODE BEHAVIOR=REEDER

Use the following option to define the fatigue onset and crack growth criterion based on the alternative form of the Paris law:

*FRACTURE CRITERION, TYPE=FATIGUE, MIXED MODE BEHAVIOR=TABULAR
*FRACTURE CRITERION, TYPE=PRX-FATIGUE, MIXED MODE BEHAVIOR=TABULAR

Use the following option to define a fatigue crack growth criterion using user subroutine *UMIXMODEFATIGUE*:

*FRACTURE CRITERION, TYPE=FATIGUE, MIXED MODE BEHAVIOR=USER
*FRACTURE CRITERION, TYPE=PRX-FATIGUE, MIXED MODE BEHAVIOR=USER

Use one of the following options to define a stress intensity factor–based fatigue crack growth criterion.

*FRACTURE CRITERION, TYPE=FATIGUE, MIXED MODE BEHAVIOR=IRWIN

*FRACTURE CRITERION, TYPE=FATIGUE, MIXED MODE BEHAVIOR=TABULAR, K-BASED

*FRACTURE CRITERION, TYPE=FATIGUE, MIXED MODE BEHAVIOR=USER, K-BASED

Abaqus/CAE Usage: The onset and growth criterion for a fatigue crack are not supported in Abaqus/CAE.

Elements

In general, any of the stress/displacement elements in Abaqus/Standard can be used in a fatigue crack growth analysis (see *Choosing the Appropriate Element for an Analysis Type*). However, when modeling fatigue crack growth with the extended finite element method based on the principles of linear elastic fracture mechanics, only first-order continuum stress/displacement elements and second-order stress/displacement tetrahedral elements can be associated with an enriched feature (see *Modeling Discontinuities as an Enriched Feature Using the Extended Finite Element Method*).

When modeling fatigue crack growth with the adaptive remeshing technique (see *Fatigue Crack Propagation with Adaptive Remeshing*), only second-order tetrahedral stress/displacement continuum elements can be used in the adaptive subdomain. However, you can use any of the stress-displacement elements available in the Abaqus/Standard element library in the nonadaptive subdomain (see *Choosing the Appropriate Element for an Analysis Type*).

Output Variables

In addition to the standard output identifiers available in Abaqus/Standard (see *Abaqus/Standard Output Variable Identifiers*), whole element and surface variables are available.

The following whole element variables are available with the extended finite element method:

CYCLEINIXFEM Number of cycles to initialize the fatigue crack growth at the enriched element.

CYCLEXFEM Number of cycles to fracture at the enriched element.

ENRRTXFEM All components of strain energy release rate at the enriched element.

The following additional surface output variables can also be requested along a predefined path at the interfaces:

CSDMG Overall value of the scalar damage variable.

BDSTAT Bond state. The bond state varies between 1.0 (fully bonded) and 0.0 (fully unbonded).

CYCLE Number of cycles to debond.

CRKLENGTH Accumulated crack length. It is measured starting from the unbonded nodes immediately behind the initial crack front. As the crack tip nodes debond, debonded element length will be added to the accumulated crack length.

ENRRT All components of strain energy release rate.

References

- Benzeggagh, M., and M. Kenane, "Measurement of Mixed-Mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-Mode Bending Apparatus," *Composite Science and Technology*, vol. 56 439, 1996.
- Deobald, L., G. Mabson, S. Engelstad, M. Rao, M. Gurvich, W. Seneviratne, S. Perera, T. O'Brien, G. Muri, J. Ratcliffe, C. Davila, N. Carvalho, and R. Krueger, "Guidelines for VCCT-Based Interlaminar Fatigue and Progressive Failure Finite Element Analysis," *NASA/TM-2017-219663*, 2017.
- Irwin, G. R., "Linear Fracture Mechanics Fracture Transition, and Fracture Control," *Engineering Fracture Mechanics*, vol. 1, pp. 241–257, 1968.
- Ratcliffe, J., and W. Johnston, "Influence of Mixed Mode I-Mode II Loading on Fatigue Delamination Growth Characteristics of a Graphite Epoxy Tape Laminate," *Proceedings of American Society for Composites 29th Technical Conference*, 2014.
- Reeder, J., S. Kyongchan, P. B. Chunchu, and D. R.. Ambur, "Postbuckling and Growth of Delaminations in Composite Plates Subjected to Axial Compression" 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Colorado, vol. 1746, p. 10, 2002.
- Wu, E. M., , and R. C. Reuter Jr., "Crack Extension in Fiberglass Reinforced Plastics," T and M Report, University of Illinois, vol. 275, 1965.

Material Damping

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Dynamic Analysis Procedures*
- *About the Material Library*
- **DAMPING*
- *Defining damping*

Overview

Material damping can be defined:

- for direct-integration (nonlinear, implicit, or explicit), subspace-based direct-integration, direct-solution steady-state, and subspace-based steady-state dynamic analysis; or
- for mode-based (linear) dynamic analysis in Abaqus/Standard.

Rayleigh Damping

In direct-integration dynamic analysis you very often define energy dissipation mechanisms—dashpots, inelastic material behavior, etc.—as part of the basic model. In such cases there is usually no need to introduce additional damping: it is often unimportant compared to these other dissipative effects. However, some models do not have such dissipation sources (an example is a linear system with chattering contact, such as a pipeline in a seismic event). In such cases it is often desirable to introduce some general damping. Abaqus provides “Rayleigh” damping for this purpose. It provides a convenient abstraction to damp lower (mass-dependent) and higher (stiffness-dependent) frequency range behavior.

Rayleigh damping can also be used in direct-solution steady-state dynamic analyses and subspace-based steady-state dynamic analyses to get quantitatively accurate results, especially near natural frequencies (see *Table 1* below).

To define material Rayleigh damping, you specify two Rayleigh damping factors: α_R for mass proportional damping and β_R for stiffness proportional damping. In general, damping is a material property specified as part of the material definition. For the cases of rotary inertia, point mass elements, and substructures, where there is no reference to a material definition, the damping can be defined in conjunction with the property references. Any mass proportional damping also applies to nonstructural features (see *Nonstructural Mass Definition*).

For a given mode *i* the fraction of critical damping, ξ_i, can be expressed in terms of the damping factors α_R and β_R as:

$$\xi_i = \frac{\alpha_R}{2\omega_i} + \frac{\beta_R \omega_i}{2},$$

where ω_i is the natural frequency at this mode. This equation implies that, generally speaking, the mass proportional Rayleigh damping, α_R, damps the lower frequencies and the stiffness proportional Rayleigh damping, β_R, damps the higher frequencies.

Mass Proportional Damping

The α_R factor introduces damping forces caused by the absolute velocities of the model and so simulates the idea of the model moving through a viscous "ether" (a permeating, still fluid, so that any motion of any point in the model causes damping). This damping factor defines mass proportional damping, in the sense that it gives a damping contribution proportional to the mass matrix for an element. If the element contains more than one material in Abaqus/Standard, the volume average value of α_R is used to multiply the element's mass matrix to define the damping contribution from this term. If the element contains more than one material in Abaqus/Explicit, the mass average value of α_R is used to multiply the element's lumped mass matrix to define the damping contribution from this term. α_R has units of (1/time).

Input File Usage: **DAMPING*, ALPHA=α_R

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Damping**: **Alpha**: α_R

Defining Variable Mass Proportional Damping

Mass proportional damping can vary during an analysis. You can define α_R as a tabular function of temperature and/or field variables.

Input File Usage: **DAMPING*, ALPHA=TABULAR

Abaqus/CAE Usage: Defining variable mass proportional damping is not supported in Abaqus/CAE.

Stiffness Proportional Damping

The β_R factor introduces damping proportional to the strain rate, which can be thought of as damping associated with the material itself. β_R defines damping proportional to the elastic material stiffness. Since the model might have quite general nonlinear response, the concept of "stiffness proportional damping" must be generalized, since it is possible for the tangent stiffness matrix to have negative eigenvalues (which would imply negative damping). To overcome this problem, β_R is interpreted as defining viscous material damping in Abaqus, which creates an additional "damping stress," $\boldsymbol{\sigma}_d$, proportional to the total strain rate:

$$\boldsymbol{\sigma}_d = \beta_R \mathbf{D}^{el} \dot{\boldsymbol{\varepsilon}},$$

where $\dot{\boldsymbol{\varepsilon}}$ is the strain rate. For hyperelastic (*Hyperelastic Behavior of Rubberlike Materials*) and hyperfoam (*Hyperelastic Behavior in Elastomeric Foams*) materials $\mathbf{D}^{el}$ is defined as the elastic stiffness in the strain-free state in Abaqus/Explicit and current elastic stiffness in Abaqus/Standard. For all other materials, $\mathbf{D}^{el}$ is the material's current elastic stiffness. $\mathbf{D}^{el}$ will be calculated based on the current temperature during the analysis.

This damping stress is added to the stress caused by the constitutive response at the integration point when the dynamic equilibrium equations are formed, but it is not included in the stress output. As a result, damping can be introduced for any nonlinear case and provides standard Rayleigh damping for linear cases; for a linear case stiffness proportional damping is exactly the same as defining a damping matrix equal to β_R times the (elastic) material stiffness matrix. Other contributions to the stiffness matrix (for example, hourglass, transverse shear, and drill stiffnesses) are not included when computing stiffness proportional damping. β_R has units of (time).

Input File Usage: **DAMPING*, BETA=β_R

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Damping**: **Beta**: β_R

Defining Variable Stiffness Proportional Damping

Stiffness proportional damping can vary during an analysis. You can define β_R as a tabular function of temperature and/or field variables.

Input File Usage: **DAMPING*, BETA=TABULAR

Abaqus/CAE Usage: Defining variable stiffness proportional damping is not supported in Abaqus/CAE.

Band-Limited Damping

For stiffness proportional damping, the fraction of critical damping is linearly proportional to the response frequency. Therefore, the desired damping ratio is achieved only at one target frequency. For practical purposes, it is useful to define the desired damping ratio uniformly over a reasonable frequency range. Band-limited damping provides this capability. It is available in Abaqus/Explicit and in Abaqus/Standard in implicit dynamic analysis (see *Table 1* below).

Unlike stiffness proportional damping, which creates an additional damping stress proportional to the total strain rate, band-limited damping creates an additional damping stress, $\boldsymbol{\sigma}_B$. This additional damping stress is proportional to the rate of filtered constitutive stress at the integration point:

$$\boldsymbol{\sigma}_B = \dot{L}\{\eta_a \sigma\},$$

where

η_a
: is the actual damping ratio calculated based on the desired damping ratio and the frequency range,

σ
: is the stress caused by the constitutive response at the integration point, and

$\dot{L}\{.\}$
: is the rate of linear transform $\mathbf{L}\{.\}$, which is the low-pass filter operator.

The frequency range is defined by the low-frequency cutoff, f_l, and the high-frequency cutoff, f_h. The actual damping ratio varies with frequency and achieves the desired damping ratio only within the frequency range. Outside the frequency range, the damping ratio is not exactly zero. The accuracy to achieve the desired damping ratio depends on the ratio of the high-frequency cutoff to the low-frequency cutoff, f_h / f_l.

For example, if the desired damping ratio is η_d, the normalized damping ratio is η_a / η_d. *Figure 1* shows the normalized damping ratio as a function of frequency for two cases: the red dotted line for the range $f_l = 10$ Hz with $f_h / f_l = 10$, and the blue solid line for the range $f_l = 8$ Hz with $f_h / f_l = 50$.

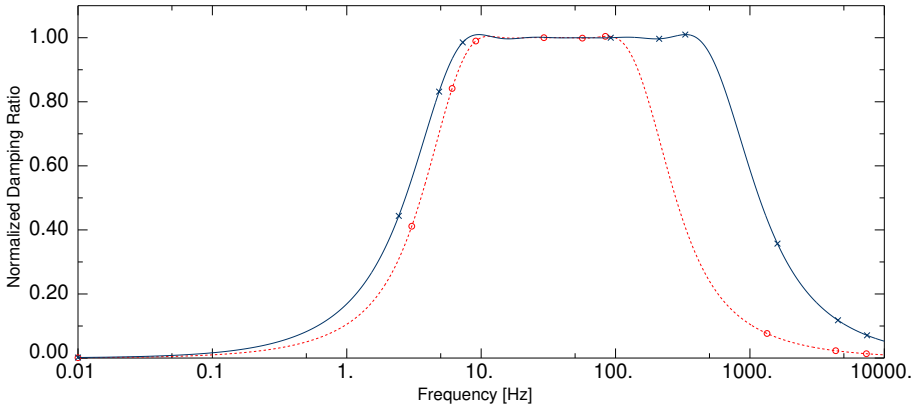


Figure 1: Normalized damping ratio.

It is well known that band-limited damping has a significant impact on the dynamic stiffness of the structure. The changes in the dynamic stiffness of the structure depend on the damping ratio and the frequency range. It increases the natural frequency of each mode, and the percentage change of the damped frequency could be large. For example, if $f_h / f_l = 100$, the maximum percentage change of the damped frequency to the natural frequency could reach approximately 1.65% (increased) with $\eta_d = 0.02$. Compared to stiffness proportional damping, the percentage change of the damped frequency is only 0.02% (reduced). For the fixed f_h / f_l, this percentage change is approximately linearly proportional to the damping ratio. Therefore, in practical applications, it is recommended that you use a low value for the damping ratio.

Similar to stiffness proportional damping, the band-limited damping stress is added to the stress caused by the constitutive response at the integration point when the dynamic equilibrium equations are formed, but it is not included in the stress output.

Input File Usage: **DAMPING*, BAND LIMITED=η_d, LOW FREQUENCY CUTOFF=f_l, HIGH FREQUENCY CUTOFF=f_h

Abaqus/CAE Usage: Defining band-limited damping is not supported in Abaqus/CAE.

Defining Variable Band-Limited Damping

You can define the desired damping ratio, η_d, as a tabular function of temperature and/or field variables to define band-limited damping.

Input File Usage: **DAMPING*, BAND LIMITED=TABULAR, LOW FREQUENCY CUTOFF=f_l, HIGH FREQUENCY CUTOFF=f_h

Abaqus/CAE Usage: Defining variable band-limited damping is not supported in Abaqus/CAE.

Energy Dissipation

The output of viscous energy dissipation associated with band-limited damping is computed by integrating the work done by the damping stresses on the rate of deformation. The output of energy dissipation computed this way is only approximate because it contains a small portion of stored energy due to the dynamic stiffness

contribution of band-limited damping, as mentioned above. For a material subjected to a harmonic excitation, the output of viscous energy dissipation can show an oscillatory behavior, with oscillation that is more pronounced at higher frequencies. An accurate value of the energy dissipated per cycle can be obtained by subtracting the value at the beginning of the cycle from the value at the end of the cycle.

Structural Damping

Structural damping assumes that the damping forces are proportional to the forces caused by stressing of the structure and are opposed to the velocity. Therefore, this form of damping can be used only when the displacement and velocity are exactly 90° out of phase. Structural damping is best suited for frequency domain dynamic procedures (see *Table 1* and *Modal Superposition Procedures* below). The damping forces are then

$$\boldsymbol{F_D^N = isI^N},$$

where $\boldsymbol{F_D^N}$ are the damping forces, $\boldsymbol{i = \sqrt{-1}}$, s is the user-defined structural damping factor, and $\boldsymbol{I^N}$ are the forces caused by stressing of the structure. The damping forces due to structural damping are intended to represent frictional effects (as distinct from viscous effects). Thus, structural damping is suggested for models involving materials that exhibit frictional behavior or where local frictional effects are present throughout the model, such as dry rubbing of joints in a multi-link structure.

Structural damping can be added to the model as mechanical dampers such as connector damping or as a complex stiffness on spring elements.

Input File Usage: Use the following option to define structural damping:

**DAMPING*, STRUCTURAL=$\boldsymbol{s}$

Abaqus/CAE Usage: Property module: material editor: **Mechanical**->**Damping**: **Structural**: $\boldsymbol{s}$

Defining Variable Structural Damping in Abaqus/Standard

Structural damping can vary during an Abaqus/Standard analysis. You can define $\boldsymbol{s}$ as a tabular function of temperature and/or field variables.

Input File Usage: **DAMPING*, STRUCTURAL=TABULAR

Abaqus/CAE Usage: Defining variable structural damping is not supported in Abaqus/CAE.

Defining Variable Frequency-Dependent Structural Damping in Abaqus/Standard

You can define $\boldsymbol{s}$ as a tabular function of frequency, temperature, and/or field variables.

Input File Usage: **DAMPING*, STRUCTURAL=TABULAR, FREQUENCY DEPENDENT

Abaqus/CAE Usage: Defining variable frequency-dependent structural damping is not supported in Abaqus/CAE.

Artificial Damping in Direct-Integration Dynamic Analysis

In Abaqus/Standard the operators used for implicit direct time integration introduce some artificial damping in addition to Rayleigh damping. Damping associated with the Hilber-Hughes-Taylor and hybrid operators is

usually controlled by the Hilber-Hughes-Taylor parameter α, which is not the same as the α_R parameter controlling the mass proportional part of Rayleigh damping. The β and γ parameters of the Hilber-Hughes-Taylor and hybrid operators also affect numerical damping. The α, β, and γ parameters are not available for the backward Euler operator. See *Implicit Dynamic Analysis Using Direct Integration* for more information about this other form of damping.

Artificial Damping in Explicit Dynamic Analysis

Rayleigh damping is meant to reflect physical damping in the actual material. In Abaqus/Explicit a small amount of numerical damping is introduced by default in the form of bulk viscosity to control high frequency oscillations; see *Explicit Dynamic Analysis* for more information about this other form of damping.

Effects of Damping on the Stable Time Increment in Abaqus/Explicit

As the fraction of critical damping for the highest mode ($\xi_{\max}$) increases, the stable time increment for Abaqus/Explicit decreases according to the equation

$$\Delta t \leq \frac{2}{\omega_{\max}} \left(\sqrt{1+\xi_{\max}^2} - \xi_{\max} \right),$$

where (by substituting $\omega_{\max}$, the frequency of the highest mode, into the equation for ξ_i given previously)

$$\xi_{\max} = \frac{\alpha_R}{2\omega_{\max}} + \frac{\beta_R \omega_{\max}}{2}.$$

These equations indicate a tendency for stiffness proportional damping to have a greater effect on the stable time increment than mass proportional damping.

To illustrate the effect that damping has on the stable time increment, consider a cantilever in bending modeled with continuum elements. The lowest frequency is $\omega_{\min} = 1$ rad/sec, while for the particular mesh chosen, the highest frequency is $\omega_{\max} = 1000$ rad/sec. The lowest mode in this problem corresponds to the cantilever in bending, and the highest frequency is related to the dilation of a single element.

With no damping the stable time increment is

$$\Delta t = \frac{2}{\omega_{\max}} = 2 \times 10^{-3} \text{ sec.}$$

If we use stiffness proportional damping to create 1% of critical damping in the lowest mode, the damping factor is given by

$$\beta_R = \frac{2 \times 0.01}{1} = 2 \times 10^{-2} \text{ sec.}$$

This corresponds to a critical damping factor in the highest mode of

$$\xi_{\max} = \frac{\omega_{\max} \beta_R}{2} = 10.$$

The stable time increment with damping is, thus, reduced by a factor of

$$\left(\sqrt{1+10^2} - 10 \right) \approx 0.05,$$

and becomes

$$\begin{aligned} \Delta t &\approx \left(2 \times 10^{-3}\right) \times 0.05 \\ &\approx 1 \times 10^{-4}. \end{aligned}$$

Thus, introducing 1% critical damping in the lowest mode reduces the stable time increment by a factor of twenty.

However, if we use mass proportional damping to damp out the lowest mode with 1% of critical damping, the damping factor is given by

$$\alpha_R = 2\omega_{\min}\xi = 2 \times 1 \times 10^{-2} = 2 \times 10^{-2}\,\text{sec}^{-1},$$

which corresponds to a critical damping factor in the highest mode of

$$\xi_{\max} = \frac{\alpha_R}{2\omega_{\max}} = \frac{2\times10^{-2}}{2\times1000} = 10^{-5}.$$

The stable time increment with damping is reduced by a factor of

$$\left(\sqrt{1+10^{-10}} - 10^{-5}\right) \approx 0.99999,$$

which is almost negligible.

This example demonstrates that it is generally preferable to damp out low-frequency response with mass proportional damping rather than stiffness proportional damping. However, mass proportional damping can significantly affect rigid body motion, so large α_R is often undesirable. To avoid a dramatic drop in the stable time increment, the stiffness proportional damping factor, β_R, should be less than or of the same order of magnitude as the initial stable time increment without damping. With $\beta_R = 2/\omega_{\max}$, the stable time increment is reduced by about 52%.

The above equation to calculate the stable time increment with critical damping is not suitable for band-limited damping. Because band-limited damping affects the dynamic stiffness of the structure, the stable time increment could have a dramatic drop if the damping ratio, η_d, or the ratio of high-frequency cutoff to low-frequency cutoff, f_h / f_l, is large. *Figure 2* illustrates the scale factor for the stable time increment with band-limited damping. The red dotted line shows the scale factor as a function of f_h / f_l with η_d=0.02, and the blue solid line shows the scale factor as a function of f_h / f_l with η_d=0.04. For comparison, with stiffness proportional damping, the scale factor is 0.9801 when $\xi_{\max}$=0.02 and is 0.9049 when $\xi_{\max}$=0.04, respectively.

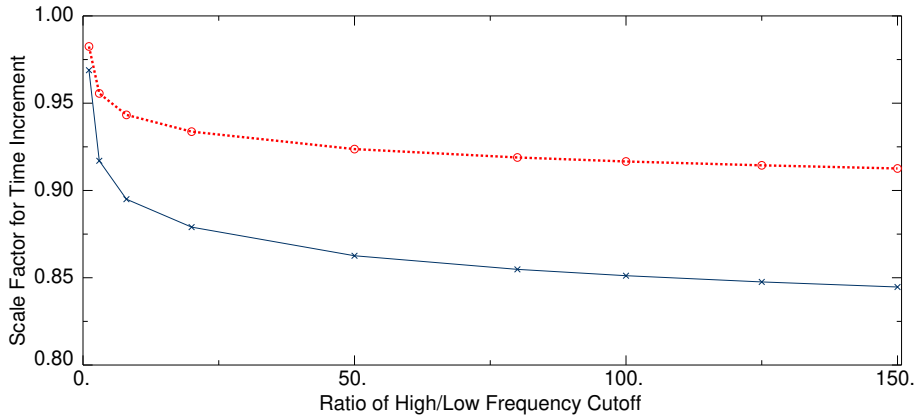


Figure 2: Scale factor for the stable time increment in band-limited damping.

Procedures

Table 1 describes the different types of material damping that you can use in different types of procedures.

Table 1: Compatible types of procedures and material damping.

Procedures	Material Damping Source			
	Alpha	Beta	Band-limited	Structural
**COMPLEX FREQUENCY*	√	√		√
**DYNAMIC*	√	√	√	
**DYNAMIC TEMPERATURE-DISPLACEMENT*,EXPLICIT	√	√	√	
**FREQUENCY*, DAMPING PROJECTION=ON	√	√		√
**MATRIX GENERATE*	√	√		√
**MODAL DYNAMIC*	√	√		√
**STEADY STATE DYNAMICS*, MODE BASED	√	√		√
**STEADY STATE DYNAMICS*, DIRECT	√	√		√
**STEADY STATE DYNAMICS*, SUBSPACE PROJECTION	√	√		√
**SUBSTRUCTURE GENERATE*	√	√		√

Modal Superposition Procedures

Damping can be specified as part of the step definition for modal superposition procedures. *Damping in a Linear Dynamic Analysis* describes the availability of damping types, which depends on the procedure type and the architecture used to perform the analysis. It also provides details on the following types of damping:

- Viscous modal damping (Rayleigh damping and fraction of critical damping)
- Structural modal damping
- Composite modal damping

Material Options

The β_R factor applies to all elements that use a linear elastic material definition (*Linear Elastic Behavior*) and to Abaqus/Standard beam and shell elements that use general sections. In the latter case, if a nonlinear beam section definition is provided, the β_R factor is multiplied by the slope of the force-strain (or moment-curvature) relationship at zero strain or curvature. In the case of equation of state materials, the β_R factor can apply only to elements that use a tabulated equation of state material definition, a linear $U_s - U_p$ equation of state material definition, or a $P - \alpha$ equation of state material definition. In addition, the β_R factor applies to all Abaqus/Explicit elements that use a hyperelastic material definition (*Hyperelastic Behavior of Rubberlike Materials*), a hyperfoam material definition (*Hyperelastic Behavior in Elastomeric Foams*), or general shell sections (*Using a General Shell Section to Define the Section Behavior*).

In the case of a no tension elastic material the β_R factor is not used in tension, while for a no compression elastic material the β_R factor is not used in compression (see *No Compression or No Tension*). In other words, these modified elasticity models exhibit damping only when they have stiffness.

Elements

The α_R factor is applied to all elements that have mass including point mass elements (discrete DASHPOTA elements in each global direction, each with one node fixed, can also be used to introduce this type of damping). For point mass and rotary inertia elements mass proportional or composite modal damping are defined as part of the point mass or rotary inertia definitions (*Point Masses* and *Rotary Inertia*).

The β_R factor is not available for spring elements: discrete dashpot elements should be used in parallel with spring elements instead.

The β_R factor is also not applied to the transverse shear terms in Abaqus/Standard beams and shells.

The hybrid element stiffness matrix formulation is different from the corresponding nonhybrid formulation; therefore, the stiffness proportional damping is different for the same value of the β_R factor in nonlinear dynamic analysis. In linear analyses Abaqus/Standard imposes equivalent stiffness proportional damping for hybrid and nonhybrid elements.

In Abaqus/Standard composite modal damping cannot be used with or within substructures. Rayleigh damping can be introduced for substructures. When Rayleigh damping is used within a substructure, α_R and β_R are averaged over the substructure to define single values of α_R and β_R for the substructure. These are weighted averages, using the mass as the weighting factor for α_R and the volume as the weighting factor for β_R. These averaged damping values can be superseded by providing them directly in a second damping definition. See *Using Substructures*.

References

- Huang, Y., R. Sturt, and M. Willford, “A Damping Model for Nonlinear Dynamic Analysis Providing Uniform Damping Over a Frequency Range,” *Computer & Structures*, vol. 212, pp. 101–109, 2019.

Thermal Expansion

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *UEXPAN*
- *VUEXPAN*
- **EXPANSION*
- *Defining other mechanical models*
- *Defining a fluid-filled porous material*

Overview

Thermal expansion effects:

- can be defined by specifying thermal expansion coefficients so that Abaqus can compute thermal strains;
- can be isotropic, transversely isotropic, orthotropic, or fully anisotropic; or, for pore fluids, can be isotropic only;
- are defined as total expansion from a reference temperature;
- can be defined as a tangent expansion in Abaqus/Standard;
- can be specified as a function of temperature and/or field variables;
- can be defined with a distribution for solid continuum elements in Abaqus/Standard;
- can be specified directly in Abaqus/Standard in the user subroutine *UEXPAN* or in Abaqus/Explicit in user subroutine *VUEXPAN* if the thermal strains are complicated functions of temperature, time, field variables, and state variables; and
- can be defined independently for the solid grains and the pore fluid in a porous medium.

Defining Thermal Expansion Coefficients

Thermal expansion is a material property included in a material definition (see *Material Data Definition*) except when it refers to the expansion of a gasket whose material properties are not defined as part of a material definition. In that case expansion must be used with the gasket behavior definition (see *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*).

In an Abaqus/Standard analysis a spatially varying thermal expansion can be defined for homogeneous solid continuum elements by using a distribution (*Distribution Definition*). The distribution must include default values for the thermal expansion. If a distribution is used, no dependencies on temperature and/or field variables for the thermal expansion can be defined.

Input File Usage: Use the following options to define thermal expansion for most materials:

**MATERIAL*
**EXPANSION*

Use the following options to define thermal expansion for gaskets whose constitutive response is defined directly as gasket behavior:

**GASKET BEHAVIOR*
**EXPANSION*

Abaqus/CAE Usage: Use the following option with other material behaviors, including gasket behavior, to include thermal expansion effects:

Property module: material editor: **Mechanical->Expansion**

Computation of Thermal Strains

In Abaqus, thermal expansion coefficients can define the total (or secant) thermal expansion or the differential (or tangent) thermal expansion as shown in *Figure 1*.

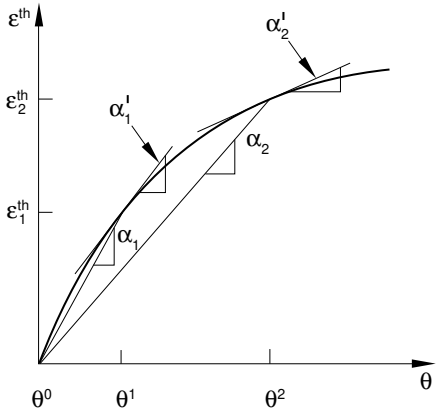


Figure 1: Definition of the thermal expansion coefficient.

Secant Thermal Expansion

The secant thermal expansion coefficients, α, define the thermal expansion from a reference temperature, θ^0, as shown in *Figure 1*.

They generate thermal strains according to the formula

$$\varepsilon^{th} = \alpha\left(\theta, f_\beta\right)\ \left(\theta - \theta^0\right) - \alpha\left(\theta^I, f_\beta^I\right)\ \left(\theta^I - \theta^0\right),$$

where

$\alpha\left(\theta, f_\beta\right)$

is the secant thermal expansion coefficient;

θ

is the current temperature;

θ^I

is the initial temperature;

f_β

are the current values of the predefined field variables;

f_β^I

are the initial values of the field variables; and

θ^0

is the reference temperature for the thermal expansion coefficient.

The second term in the above equation represents the strain arising from the difference between the initial temperature, θ^I, and the reference temperature, θ^0. This term is necessary to enforce the assumption that there is no initial thermal strain for cases in which the reference temperature does not equal the initial temperature.

Input File Usage: **EXPANSION*, DEFINITION=SECANT

Defining the Reference Temperature

If the secant coefficient of thermal expansion, α, is not a function of temperature or field variables, the value of the reference temperature, θ^0, is not needed. If α is a function of temperature or field variables, you can define θ^0.

Input File Usage: **EXPANSION*, ZERO=θ^0

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Expansion**: **Reference temperature:** θ^0

Tangent Thermal Expansion in Abaqus/Standard

Abaqus/Standard also supports the definition of thermal expansion coefficients in tangent form. This capability is particularly useful in situations when a material undergoes curing or similar irreversible transformations and the thermal expansion coefficient depends on the degree of transformation. Such processes are usually described using a differential (or tangent) form of thermal expansion that cannot be generally expressed in total (or secant) form because of the irreversible nature of the transformation.

The tangent coefficient of thermal expansion, α', is the slope of the expansion curve at a given temperature (see *Figure 1*). If the tangent coefficient is specified, the increment of the thermal strain is computed from

$$d\varepsilon^{th} = \alpha' \left(\theta, f_\beta\right) d\theta;$$

and the thermal strain, ε^{th}, is obtained by integrating the above equation from the initial to the current temperature.

Because of the incremental nature of the formulation, the resulting thermal strains generally depend on the temperature and field variable history and might be sensitive to the time increment size (small time increments are preferred). Therefore, the general recommendation is to use the total (secant) form except for the special situations of irreversible processes where the tangent formulation can be more suitable to describe the evolution of thermal strains.

You cannot specify tangent thermal expansion coefficients for short-fiber reinforced composites, the pore fluid in the porous medium, and the TNM model.

Input File Usage: **EXPANSION*, DEFINITION=TANGENT

Abaqus/CAE Usage: Defining tangent thermal expansion is not supported in Abaqus/CAE.

Converting Thermal Expansion Coefficients from Differential Form to Total Form

As discussed earlier, except for materials undergoing irreversible processes, the general recommendation is to use the total form definition of thermal expansion 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. In addition, the total form is the only option available in Abaqus/Explicit. Therefore, it is sometimes necessary to convert the thermal expansion coefficients from the differential (tangent) to the total (secant) form. This conversion is always possible when the thermal expansion coefficients are only a function of temperature (no field variable dependency).

The temperature dependent tangent thermal expansion data has the form:

$$d\varepsilon^{th} = \alpha'(\theta)\, d\theta;$$

that is, the tangent to the strain-temperature curve is provided (see *Figure 1*). To convert to the secant thermal expansion form required by Abaqus, this relationship must be integrated from a suitably chosen reference temperature, θ^0:

$$\varepsilon^{th} = \int_{\theta^0}^{\theta} \alpha' d\theta \Rightarrow \alpha(\theta) = \frac{1}{\theta - \theta^0} \int_{\theta^0}^{\theta} \alpha' d\theta.$$

For example, suppose α' is a series of constant values: α'_1 between θ^0 and θ^1; α'_2 between θ^1 and θ^2; α'_3 between θ^2 and θ^3; etc. Then,

$$\begin{aligned}
\varepsilon_1^{th} &= \alpha'_1 \left(\theta^1 - \theta^0\right) \\
\varepsilon_2^{th} &= \varepsilon_1^{th} + \alpha'_2 \left(\theta^2 - \theta^1\right) \\
\varepsilon_3^{th} &= \varepsilon_2^{th} + \alpha'_3 \left(\theta^3 - \theta^2\right).
\end{aligned}$$

The corresponding total expansion coefficients required by Abaqus are then obtained as

$$\begin{aligned}
\alpha_1 &= \varepsilon_1^{th} / \left(\theta^1 - \theta^0\right) \\
\alpha_2 &= \varepsilon_2^{th} / \left(\theta^2 - \theta^0\right) \\
\alpha_3 &= \varepsilon_3^{th} / \left(\theta^3 - \theta^0\right).
\end{aligned}$$

Computing Thermal Strains in Linear Perturbation Steps

During a linear perturbation step, temperature perturbations can produce perturbations of thermal strains in the form:

$$\Delta\varepsilon^{th} = \alpha'\left(\theta^B\right)\Delta\theta,$$

where $\Delta\theta$ is the temperature perturbation load about the base state, θ^B is the temperature in the base state, and $\alpha'\left(\theta^B\right)$ is the tangent thermal expansion coefficient evaluated in the base state. If the secant thermal expansion coefficients are specified, Abaqus computes the tangent thermal expansion coefficients from the total form as

$$\alpha'(\theta) = \frac{\partial \varepsilon^{th}}{\partial \theta} = \alpha + \frac{\partial \alpha}{\partial \theta}\left(\theta - \theta^0\right).$$

Defining Increments of Thermal Strain in User Subroutines

Increments of thermal strain can be specified in user subroutine *UEXPAN* in Abaqus/Standard and in user subroutine *VUEXPAN* in Abaqus/Explicit as functions of temperature and/or predefined field variables. User subroutine *UEXPAN* in Abaqus/Standard must be used if the thermal strain increments depend on state variables.

Input File Usage: **EXPANSION*, USER

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Expansion**: **Use user subroutine UEXPAN**

Defining the Initial Temperature and Field Variable Values

If the coefficient of thermal expansion, α, is a function of temperature or field variables, the initial temperature and initial field variable values, θ^I and f_β^I, are given as described in *Initial Conditions*.

Element Removal and Reactivation

If an element has been removed and subsequently reactivated in Abaqus/Standard (*Element and Contact Pair Removal and Reactivation*), θ^I and f_β^I in the equation for the thermal strains represent temperature and field variable values as they were at the moment of reactivation.

Defining Directionally Dependent Thermal Expansion

Isotropic, orthotropic, and fully anisotropic thermal expansion can be defined in Abaqus.

Orthotropic and anisotropic thermal expansion can be used only with materials where the material directions are defined with local orientations (see *Orientations*).

Isotropic Expansion

If the thermal expansion coefficient is defined directly, only one value of α is needed at each temperature. If the user subroutine *UEXPAN* is used, only one isotropic thermal strain increment ($\Delta\varepsilon = \Delta\varepsilon_{11} = \Delta\varepsilon_{22} = \Delta\varepsilon_{33}$) must be defined.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

**EXPANSION*, TYPE=ISO

Use the following option to define the thermal expansion with user subroutine *UEXPAN* in Abaqus/Standard and *VUEXPAN* in Abaqus/Explicit:

**EXPANSION*, TYPE=ISO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical->Expansion**: **Type: Isotropic**

Use the following input to define the thermal expansion with user subroutine *UEXPAN*:

Property module: material editor: **Mechanical->Expansion**: **Type: Isotropic**, **Use user subroutine UEXPAN**

Orthotropic Expansion

If the thermal expansion coefficients are defined directly, the three expansion coefficients in the principal material directions (α_{11}, α_{22}, and α_{33}) should be given as functions of temperature. If user subroutines *UEXPAN* and *VUEXPAN* are used, the three components of thermal strain increment in the principal material directions ($\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, and $\Delta\varepsilon_{33}$) must be defined.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

**EXPANSION*, TYPE=ORTHO

Use the following option to define the thermal expansion with user subroutine *UEXPAN* in Abaqus/Standard and *VUEXPAN* in Abaqus/Explicit:

**EXPANSION*, TYPE=ORTHO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical->Expansion**: **Type: Orthotropic**

Use the following input to define the thermal expansion with the user subroutine *UEXPAN*:

Property module: material editor: **Mechanical->Expansion**: **Type: Orthotropic**, **Use user subroutine UEXPAN**

Transversely Isotropic Expansion

A special subclass of orthotropy is *transverse isotropy*, which is characterized by a plane of isotropy at every point in the material. Abaqus assumes the 2–3 plane to be the plane of isotropy at every point; therefore, $\alpha_{22} = \alpha_{33}$. Only two expansion coefficients in the principal material directions (α_{11} and α_{22}) are needed as functions of temperature.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

**EXPANSION*, TYPE=TRANSVERSELY ISOTROPIC

Abaqus/CAE Usage: Defining thermal expansion for a transversely isotropic material is not supported in Abaqus/CAE.

Anisotropic Expansion

If the thermal expansion coefficients are defined directly, all six components of $\boldsymbol{\alpha}$ (α_{11}, α_{22}, α_{33}, α_{12}, α_{13}, α_{23}) must be given as functions of temperature. If user subroutine *UEXPAN* is used in Abaqus/Standard, all six components of the thermal strain increment ($\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, $\Delta\varepsilon_{33}$, $\Delta\varepsilon_{12}$, $\Delta\varepsilon_{13}$, $\Delta\varepsilon_{23}$) must be defined. If user subroutine *VUEXPAN* is used in Abaqus/Explicit, all six components of the thermal strain increment ($\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, $\Delta\varepsilon_{33}$, $\Delta\varepsilon_{12}$, $\Delta\varepsilon_{23}$,$\Delta\varepsilon_{13}$) must be defined.

In an Abaqus/Standard analysis if a distribution is used to define the thermal expansion, the number of expansion coefficients given for each element in the distribution, which is determined by the associated distribution table (*Distribution Definition*), must be consistent with the level of anisotropy specified for the expansion behavior. For example, if orthotropic behavior is specified, three expansion coefficients must be defined for each element in the distribution.

Input File Usage: Use the following option to define the thermal expansion coefficient directly:

**EXPANSION*, TYPE=ANISO

Use the following option to define the thermal expansion with user subroutine *UEXPAN* in Abaqus/Standard and *VUEXPAN* in Abaqus/Explicit:

**EXPANSION*, TYPE=ANISO, USER

Abaqus/CAE Usage: Use the following input to define the thermal expansion coefficient directly:

Property module: material editor: **Mechanical->Expansion**: **Type: Anisotropic**

Use the following input to define the thermal expansion with user subroutine *UEXPAN*:

Property module: material editor: **Mechanical->Expansion**: **Type: Anisotropic**, **Use user subroutine UEXPAN**

Defining Thermal Expansion for a Short-Fiber Reinforced Composite

The thermal expansion coefficient of a short-fiber reinforced composite (for example, an injection molded composite) can be computed using the orientation averaging described by Zheng *(2011)*:

$$\langle \alpha \rangle = \langle D^{UD} \rangle^{-1} \langle D^{UD} \alpha^{UD} \rangle ,$$

where $\langle D^{UD} \rangle$ is the orientation-averaged elasticity matrix computed using the elasticity of the unidirectional (UD) composite and the second-order orientation tensor (see *Defining the Elasticity of a Short-Fiber Reinforced Composite*), and $\langle D^{UD} \alpha^{UD} \rangle$ is given by:

$$\langle D^{UD}_{ijkl} \alpha^{UD}_{kl} \rangle = \left[\left(D^{UD}_{1111} - D^{UD}_{1122} \right) \alpha^{UD}_{11} + \left(2D^{UD}_{1122} - D^{UD}_{2222} - D^{UD}_{2233} \right) \alpha^{UD}_{22} \right] a_{ij} + \left[D^{UD}_{2211} \alpha^{UD}_{11} + \left(D^{UD}_{2222} + D^{UD}_{2233} \right) \alpha^{UD}_{22} \right] \delta_{ij},$$

where D^{UD} and α^{UD} are the elasticity matrix and thermal expansion coefficient of the unidirectional composite with the 1-direction as the fiber direction, a_{ij} is the second-order orientation tensor, and δ_{ij} is the Kronecker delta. The unidirectional composite is assumed to be transversely isotropic. Similar to elasticity, you must define the material directions with local orientations (see *Orientations*), and the axes of the local system must align with the principal directions of the second-order orientation tensor.

Input File Usage: Use the following option to define the transversely isotropic thermal expansion coefficient of the unidirectional composite:

**EXPANSION*,TYPE=ORTHO

Use the following option to define the second-order orientation tensor in the orientation definition in Abaqus/Explicit:

**ORIENTATION*, DISPERSION

Use the following option to define the second-order orientation tensor in the material definition:

**FIBER DISPERSION*

Thermal Stress

When a structure is not free to expand, a change in temperature will cause stress. For example, consider a single two-node truss of length L that is completely restrained at both ends. The cross-sectional area; the Young's modulus, E; and the thermal expansion coefficient, α, are all constant. The stress in this one-dimensional problem can then be calculated from Hooke's Law as $\sigma_x = E\left(\varepsilon_x - \varepsilon_x^{th}\right)$, where ε_x is the total strain and $\varepsilon_x^{th} = \alpha\Delta\theta$ is the thermal strain, where $\Delta\theta$ is the temperature change. Since the element is fully restrained, $\varepsilon_x = 0$. If the temperature at both nodes is the same, we obtain the stress $\sigma_x = -E\alpha\Delta\theta$.

Constrained thermal expansion can cause significant stress. For typical structural metals, temperature changes of about 150°C (300°F) can cause yield. Therefore, it is often important to define boundary conditions with particular care for problems involving thermal loading to avoid overconstraining the thermal expansion.

Energy Balance Considerations

Abaqus does not account for thermal expansion effects in the total energy balance equation, which can lead to an apparent imbalance of the total energy of the model. For example, in the example above of a two-node truss restrained at both ends, constrained thermal expansion introduces strain energy that will result in an equivalent increase in the total energy of the model.

Material Options

Thermal expansion can be combined with any other (mechanical) material (see *Combining Material Behaviors*) behavior in Abaqus.

Using Thermal Expansion with Other Material Models

For most materials, thermal expansion is defined by a single coefficient or set of orthotropic or anisotropic coefficients or by defining the incremental thermal strains in user subroutine *UEXPAN* in Abaqus/Standard and *VUEXPAN* in Abaqus/Explicit. For porous media such as soils or rock, you can define thermal expansion for the solid grains and for the permeating fluid (when using either the coupled pore fluid diffusion/stress procedure in Abaqus/Standard or Abaqus/Explicit for the solid grains and for the permeating fluid using either the coupled pore fluid diffusion/stress procedure—see *Coupled Pore Fluid Diffusion and Stress Analysis*—or undrained pore fluid flow and stress analysis in Abaqus/Explicit—see *Undrained Pore Fluid Flow and Stress Analysis*). In such cases, you should repeat the thermal expansion definition to define the different thermal expansion effects.

Using Thermal Expansion with Gasket Behaviors

Thermal expansion can be used with any gasket behavior definition. Thermal expansion will affect the expansion of the gasket in the membrane direction and/or the expansion in the gasket's thickness direction.

Elements

Thermal expansion can be used with any stress/displacement or fluid element in Abaqus.

Output

THE:

Thermal strain tensor (symmetric).

THEFL:

Thermal strain in the pore fluid in a porous medium (scalar).

References

- Zheng, R., I. Tanner, and X. Fan, Injection Molding: Integration of Theory and Modeling Methods, Springer , 2011.

Field Expansion

Products: Abaqus/Standard

References:

- *About the Material Library*
- *UEXPAN*
- **EXPANSION*

Overview

Field expansion effects:

- can be defined by specifying field expansion coefficients so that Abaqus/Standard can compute field expansion strains that are driven by changes in predefined field variables;
- can be isotropic, orthotropic, or fully anisotropic; or, for pore fluids, can be isotropic only;
- are defined as total expansion from a reference value of the predefined field variable;
- can be specified as a function of temperature and/or predefined field variables;
- can be specified directly in user subroutine *UEXPAN* (if the field expansion strains are complicated functions of field variables and state variables);
- can be defined for more than one predefined field variable; and
- can be defined independently for the solid grains and the pore fluid in a porous medium.

Defining Field Expansion Coefficients

Field expansion is a material property included in a material definition (see *Material Data Definition*) except when it refers to the expansion of a gasket whose material properties are not defined as part of a material definition. In that case field expansion must be used in conjunction with the gasket behavior definition (see *Defining the Gasket Behavior Directly Using a Gasket Behavior Model*).

Input File Usage: Use the following options to define field expansion associated with predefined field variable number *n* for most materials:

**MATERIAL*
**EXPANSION*, FIELD=*n*

The **EXPANSION* option can be repeated with different values of the predefined field variable number *n* to define field expansion associated with more than one field.

Use the following options to define field expansion associated with predefined field variable number *n* for gaskets whose constitutive response is defined directly as gasket behavior:

**GASKET BEHAVIOR*
**EXPANSION*, FIELD=*n*

The **EXPANSION* option can be repeated with different values of the predefined field variable number n to define field expansion associated with more than one field.

Computation of Field Expansion Strains

Abaqus/Standard requires field expansion coefficients, α_f, that define the total field expansion from a reference value of the predefined field variable n, f_n^0, as shown in *Figure 1*.

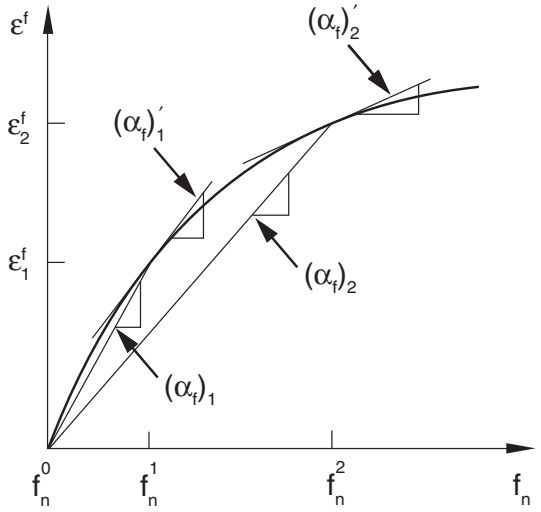


Figure 1: Definition of the field expansion coefficient.

The field expansion for each specified field generates field expansion strains according to the formula

$$\varepsilon^f = \alpha_f(\theta, f_\beta)\,(f_n - f_n^0) - \alpha_f\left(\theta^I, f_\beta^I\right)\,(f_n^I - f_n^0),$$

where

$\alpha_f(\theta, f_\beta)$

is the field expansion coefficient;

f_n

is the current value of the predefined field variable n;

f_n^I

is the initial value of the predefined field variable n;

f_β

are the current values of the predefined field variables;

f_β^I

are the initial values of the predefined field variables; and

f_n^0

is the reference value of the predefined field variable n for the field expansion coefficient.

The second term in the above equation represents the strain due to the difference between the initial value of the predefined field variable$n$, f_n^I, and the corresponding reference value, f_n^0. This term is necessary to enforce the assumption that there is no initial field expansion strain for cases in which the reference value of the predefined field variable n does not equal the corresponding initial value.

Defining the Reference Value of the Predefined Field Variable

If the coefficient of field expansion, α_f, is not a function of temperature or field variables, the reference value of the predefined field variable, f_n^0, is not needed. If α_f is a function of temperature or field variables, you can define f_n^0.

Input File Usage: **EXPANSION*, FIELD=n, ZERO=f_n^0

Converting Field Expansion Coefficients from Differential Form to Total Form

Total field expansion coefficients can be provided directly as outlined in the previous section. However, you may have field expansion data available in differential form:

$$d\varepsilon^f = \alpha'_f(f_n)\, df_n;$$

that is, the tangent to the strain-field variable curve is provided (see *Figure 1*). To convert to the total field expansion form required by Abaqus, this relationship must be integrated from a suitably chosen reference value of the field variable, f_n^0:

$$\varepsilon^f = \int_{f_n^0}^{f_n} \alpha'_f\, df_n \Rightarrow \alpha_f(f_n) = \frac{1}{f_n - f_n^0} \int_{f_n^0}^{f_n} \alpha'_f\, df_n.$$

For example, suppose α'_f is a series of constant values: $\left(\alpha'_f\right)_1$ between f_n^0 and f_n^1; $\left(\alpha'_f\right)_2$ between f_n^1 and f_n^2; $(\alpha')_3$ between f_n^2 and f_n^3; etc. Then,

$$\begin{aligned}
\varepsilon_1^f &= \left(\alpha'_f\right)_1 \left(f_n^1 - f_n^0\right), \\
\varepsilon_2^f &= \varepsilon_1^f + \left(\alpha'_f\right)_2 \left(f_n^2 - f_n^1\right), \\
\varepsilon_3^f &= \varepsilon_2^f + \left(\alpha'_f\right)_3 \left(f_n^3 - f_n^2\right).
\end{aligned}$$

The corresponding total expansion coefficients required by Abaqus are then obtained as

$$\begin{aligned}
(\alpha_f)_1 &= \varepsilon_1^f / \left(f_n^1 - f_n^0\right), \\
(\alpha_f)_2 &= \varepsilon_2^f / \left(f_n^2 - f_n^0\right), \\
(\alpha_f)_3 &= \varepsilon_3^f / \left(f_n^3 - f_n^0\right).
\end{aligned}$$

Computing Field Expansion Strains in Linear Perturbation Steps

During a linear perturbation step, field variable perturbations can produce perturbations of field expansion strains in the form:

$$\Delta\varepsilon^f = \alpha'_f\left(f_n^B\right)\Delta f_n,$$

where Δf_n is the field variable perturbation load about the base state, f_n^B is the field variable in the base state, and $\alpha'_f\left(f_n^B\right)$ is the tangent field expansion coefficient evaluated in the base state. Abaqus computes the tangent field expansion coefficients from the total form as

$$\alpha'_f\left(f_n\right) = \frac{\partial\varepsilon^f}{\partial f_n} = \alpha_f + \frac{\partial\alpha_f}{\partial f_n}\left(f_n - f_n^0\right).$$

Defining Increments of Field Expansion Strain in User Subroutine UEXPAN

Increments of field expansion strain can be specified in user subroutine *UEXPAN* as functions of temperature and/or predefined field variables. User subroutine *UEXPAN* must be used if the field expansion strain increments depend on state variables.

You can use user subroutine *UEXPAN* more than once within a single material definition. In particular, you can define both thermal and field expansions or multiple field expansions within the same material definition using user subroutine *UEXPAN*.

Input File Usage: **EXPANSION*, FIELD=*n*, USER

Defining the Initial Temperature and Field Variable Values

If the coefficient of field expansion, α_f, is a function of temperature and/or predefined field variables, the initial temperature and initial predefined field variable values, θ^I and f_β^I, are given as described in *Initial Conditions*.

Element Removal and Reactivation

If an element has been removed and subsequently reactivated (*Element and Contact Pair Removal and Reactivation*), θ^I and f_β^I in the equation for the field expansion strains represent temperature and predefined field variable values as they were at the moment of reactivation.

Defining Directionally Dependent Field Expansion

Isotropic, orthotropic, or fully anisotropic field expansion can be defined.

Orthotropic and anisotropic field expansion can be used only with materials where the material directions are defined with local orientations (see *Orientations*).

Only isotropic field expansion is allowed with the hyperelastic and hyperfoam material models.

Isotropic Expansion

If the field expansion coefficient is defined directly, only one value of α_f is needed at each temperature and/or predefined field variable. If user subroutine *UEXPAN* is used, only one isotropic field expansion strain increment ($\Delta\varepsilon^f = \Delta\varepsilon_{11}^f = \Delta\varepsilon_{22}^f = \Delta\varepsilon_{33}^f$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficient directly:

**EXPANSION*, FIELD=*n*, TYPE=ISO

Use the following option to define the field expansion with user subroutine *UEXPAN*:

**EXPANSION*, FIELD=*n*, TYPE=ISO, USER

Orthotropic Expansion

If the field expansion coefficients are defined directly, the three expansion coefficients in the principal material directions (α_{f11}, α_{f22}, and α_{f33}) should be given as functions of temperature and/or predefined field variables. If user subroutine *UEXPAN* is used, the three components of field expansion strain increment in the principal material directions ($\Delta\varepsilon^f_{11}$, $\Delta\varepsilon^f_{22}$, and $\Delta\varepsilon^f_{33}$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficients directly:

**EXPANSION*, FIELD=*n*, TYPE=ORTHO

Use the following option to define the field expansion with user subroutine *UEXPAN*:

**EXPANSION*, FIELD=*n*, TYPE=ORTHO, USER

Anisotropic Expansion

If the field expansion coefficients are defined directly, all six components of α_f (α_{f11}, α_{f22}, α_{f33}, α_{f12}, α_{f13}, α_{f23}) must be given as functions of temperature and/or predefined field variables. If user subroutine *UEXPAN* is used, all six components of the field expansion strain increment ($\Delta\varepsilon^f_{11}$, $\Delta\varepsilon^f_{22}$, $\Delta\varepsilon^f_{33}$, $\Delta\varepsilon^f_{12}$, $\Delta\varepsilon^f_{13}$, $\Delta\varepsilon^f_{23}$) must be defined.

Input File Usage: Use the following option to define the field expansion coefficients directly:

**EXPANSION*, FIELD=*n*, TYPE=ANISO

Use the following option to define the field expansion with user subroutine *UEXPAN*:

**EXPANSION*, FIELD=*n*, TYPE=ANISO, USER

Field Expansion Stress

When a structure is not free to expand, a change in a predefined field variable will cause stress if there is field expansion associated with that predefined field variable. For example, consider a single 2-node truss of length *L* that is completely restrained at both ends. The cross-sectional area; the Young's modulus, *E*; and the field expansion coefficient, α_f, are all constants. The stress in this one-dimensional problem can then be calculated from Hooke's Law as $\sigma_x = E\left(\varepsilon_x - \varepsilon^f_x\right)$, where ε_x is the total strain and $\varepsilon^f_x = \alpha_f \Delta f_n$ is the field expansion strain, where Δf_n is the change in the value of the predefined field variable number *n*. Since the element is fully restrained, $\varepsilon_x = 0$. If the values of the field variable at both nodes are the same, we obtain the stress $\sigma_x = -E\alpha_f \Delta f_n$.

Depending on the value of the field expansion coefficient and the change in the value of the corresponding predefined field variable, a constrained field expansion can cause significant stress and introduce strain energy that will result in an equivalent increase in the total energy of the model. Therefore, it is often important to define boundary conditions with particular care for problems involving this property to avoid overconstraining the field expansion.

Material Options

Field expansion can be combined with any other (mechanical) material (see *Combining Material Behaviors*) behavior in Abaqus/Standard.

Using Field Expansion with Other Material Models

For most materials field expansion is defined by a single coefficient or a set of orthotropic or anisotropic coefficients or by defining the incremental field expansion strains in user subroutine *UEXPAN*.

Using Field Expansion with Gasket Behavior

Field expansion can be used in conjunction with any gasket behavior definition. Field expansion will affect the expansion of the gasket in the membrane direction and/or the expansion in the gasket's thickness direction.

Elements

Field expansion can be used with any stress/displacement element in Abaqus/Standard, except for beam and shell elements using a general section behavior.

Output

The following variables for the thermal expansion strains related output are available:

FVE:

Field expansion strain tensor (symmetric). If you define more than one field expansion option for the material, output request FVE outputs the expansion strains corresponding to all field variables for which expansion is defined. Abaqus/Standard limits the total number of field expansion strains available for output to a maximum of 50.

FVE*n*:

Field expansion strain tensor (symmetric) corresponding to the field variable *n*.

FVEFL:

Field expansion strain (scalar) in the pore fluid in a porous medium (scalar). If you define more than one field expansion option for the material, output request FVEFL outputs expansion strains corresponding to all the field variables for which expansion is defined (subject to limitations noted below). Abaqus/Standard limits the total number of field expansion strains in the pore fluid available for output to a maximum of 50.

FVEFL*n*:

Field expansion strain (scalar) in the pore fluid corresponding to the field variable *n*.

Pore Fluid Pressure

Products: Abaqus/Standard Abaqus/Explicit

References:

- **PORE FLUID PRESSURE*
- *Predefined Fields for Sequential Coupling*

Overview

Pore fluid pressure:

- allows the inclusion of the effects of a known pore fluid pressure field in a stress analysis by associating the known pore fluid pressure field with a predefined field variable in Abaqus; and
- can be used in a static or in an explicit dynamic stress analysis.

Defining Pore Fluid Pressure

In geotechnical applications a porous medium typically consists of a solid skeleton of numerous solid grains that are "loosely packed" and a pore fluid that flows through a system of interconnected pores among these solid grains. When subjected to loading, both the solid skeleton and the pore fluid respond in a manner that leads generally to a fully coupled response where the deformation in the solid skeleton depends on the pressure of the pore fluid and vice versa. Abaqus/Standard supports analysis of such a porous medium through a fully coupled pore pressure–displacement analysis (see *Coupled Pore Fluid Diffusion and Stress Analysis*).

In some situations you can approximate the two-way coupling using a sequentially coupled approach that consists of the following two parts:

1. A purely pore fluid flow analysis that computes the pore fluid pressure distribution in the porous medium without accounting for the deformation in the solid skeleton. Typically, third-party software is used to complete this part (outside of Abaqus).
2. A pure stress analysis that computes only the deformation of the solid skeleton, taking into account the effects of the pore fluid pressure field computed in the first part. Typically, a static or explicit dynamic analysis in Abaqus is used to complete this part.

The pore fluid pressure definition allows the inclusion of a known pore fluid pressure field as a predefined field variable in Abaqus in the second part of the sequential workflow above (see *Predefined Fields for Sequential Coupling*). The material constitutive response in the stress analysis is defined in terms of:

- the behavior of the solid skeleton, which can utilize any of the mechanical constitutive models available in Abaqus, and
- the known pore fluid pressure field computed in the first part of the sequential analysis and included in the second part as a predefined field variable.

The pore fluid pressure, u_w, contributes to the total stress as

$$\overline{\boldsymbol{\sigma}}^* = \boldsymbol{\sigma} + u_w \mathbf{I}.$$

In the above expression $\boldsymbol{\sigma}$ is the total stress, and $\overline{\boldsymbol{\sigma}}^*$ is the effective stress that is a function of the effective strain and the constitutive response of the solid skeleton (see *Effective stress principle for porous media*). A static stress analysis in Abaqus/Standard can account for the compressibility of the solid grains (see *Porous Bulk*

Moduli) in computing the effective strain by subtracting the volumetric strain in the solid grain due to the pore pressure from the total strain. An explicit dynamic stress analysis in Abaqus/Explicit assumes that the solid grains are incompressible and that the effective strain is equal to the total strain.

Input File Usage: Use the following options to define pore fluid pressure:

**MATERIAL*, NAME=*name*
**PORE FLUID PRESSURE*, FIELD=*n*

Defining Initial Values of Pore Fluid Pressure

To initialize the known pore fluid pressure field, you must initialize the corresponding field variable.

Input File Usage: **INITIAL CONDITIONS*, TYPE=FIELD, VARIABLE=*n*

Elements

Pore fluid pressure can be used with any solid (continuum) stress/displacement element that supports a three-dimensional stress state.

Viscosity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Viscous Shear Behavior*
- **VISCOSITY*
- **EOS*
- **TRS*
- *Defining viscosity*

Overview

Material shear viscosity in Abaqus/Explicit:

- can be a function of temperature and shear strain rate; and
- must be used in combination with an equation of state (*Equation of State*).

Material shear viscosity in Abaqus/Standard:

- is valid only for fluid pipe and fluid pipe connector elements;
- can be a function of temperature and field variables; and
- is available for the Newtonian, power law, Herschel-Bulkley, and Bingham plastic viscous fluid models.

Viscous Shear Behavior

The resistance to flow of a viscous fluid is described by the following relationship between deviatoric stress and strain rate

$$\mathbf{S} = 2\eta\dot{\mathbf{e}} = \eta\dot{\boldsymbol{\gamma}},$$

where $\mathbf{S}$ is the deviatoric stress, $\dot{\mathbf{e}}$ is the deviatoric part of the strain rate, η is the viscosity, and $\dot{\boldsymbol{\gamma}} = 2\dot{\mathbf{e}}$ is the engineering shear strain rate.

Newtonian fluids are characterized by a viscosity that only depends on temperature, $\eta(\theta)$. In the more general case of non-Newtonian fluids the viscosity is a function of the temperature and shear strain rate:

$$\eta = \eta(\dot{\gamma}, \theta),$$

where $\dot{\gamma} = \sqrt{\frac{1}{2}\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}}$ is the equivalent shear strain rate. In terms of the equivalent shear stress, $\tau = \sqrt{\frac{1}{2}\mathbf{S} : \mathbf{S}}$, we have:

$$\tau = \eta\dot{\gamma}.$$

Non-Newtonian fluids can be classified as shear-thinning (or pseudoplastic), when the apparent viscosity decreases with increasing shear rate, and shear-thickening (or dilatant), when the viscosity increases with strain rate.

In addition to the Newtonian viscous fluid model, Abaqus/Explicit supports several models of nonlinear viscosity to describe non-Newtonian fluids: power law, Carreau-Yasuda, Cross, Herschel-Bulkley, Powell-Eyring, and

Ellis-Meter. Other functional forms of the viscosity can also be specified in tabular format. In addition, in Abaqus/Explicit user subroutine *VUVISCOSITY* can be used.

In addition to the Newtonian viscous fluid model, Abaqus/Standard supports several models of nonlinear viscosity to describe non-Newtonian fluids for use with fluid pipe elements and fluid pipe connector elements: power law, Herschel-Bulkley, and Bingham plastic.

Newtonian

The Newtonian model is useful to model viscous laminar flow governed by the Navier-Poisson law of a Newtonian fluid, $\tau = \eta\dot{\gamma}$. Newtonian fluids are characterized by a viscosity that depends only on temperature, $\eta(\theta)$. You need to specify the viscosity as a tabular function of temperature when you define the Newtonian viscous deviatoric behavior.

In Abaqus/Standard this model is used to define the fluid behavior in fluid pipe and fluid pipe connector elements.

Input File Usage: **VISCOSITY*, DEFINITION=NEWTONIAN (default)

Abaqus/CAE Usage: Property module: material editor: **Mechanical->Viscosity**

Power Law

The power law model is commonly used to describe the viscosity of non-Newtonian fluids. The viscosity is expressed as

$$\eta = k\dot{\gamma}^{n-1}\ ; \quad \eta_{\min} \le \eta \le \eta_{\max},$$

where k is the flow consistency index and n is the flow behavior index. When $n < 1$, the fluid is shear-thinning (or pseudoplastic): the apparent viscosity decreases with increasing shear rate. When $n > 1$, the fluid is shear-thickening (or dilatant); and when $n = 1$, the fluid is Newtonian. Optionally, you can place a lower limit, $\eta_{\min}$, and/or an upper limit, $\eta_{\max}$, on the value of the viscosity computed from the power law.

In Abaqus/Standard this model is used to define the fluid behavior in fluid pipe and fluid pipe connector elements.

Input File Usage: **VISCOSITY*, DEFINITION=POWER LAW

Abaqus/CAE Usage: The power law model is not supported in Abaqus/CAE.

Carreau-Yasuda

The Carreau-Yasuda model describes the shear thinning behavior of polymers and is available only in Abaqus/Explicit. This model often provides a better fit than the power law model for both high and low shear strain rates. The viscosity is expressed as

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)\left(1 + (\lambda\dot{\gamma})^a\right)^{\frac{n-1}{a}},$$

where η_0 is the low shear rate Newtonian viscosity, η_∞ is the infinite shear viscosity (at high shear strain rates), λ is the natural time constant of the fluid ($1/\lambda$ is the critical shear rate at which the fluid changes from Newtonian to power law behavior), and n represents the flow behavior index in the power law regime. The coefficient a is a material parameter. The original Carreau model is recovered when a=2.

Input File Usage: **VISCOSITY*, DEFINITION=CARREAU-YASUDA

Abaqus/CAE Usage: The Carreau-Yasuda model is not supported in Abaqus/CAE.

Cross

The Cross model is commonly used when it is necessary to describe the low-shear-rate behavior of the viscosity and is available only in Abaqus/Explicit. The viscosity is expressed as

$$\eta = \eta_\infty + \frac{(\eta_0 - \eta_\infty)}{1+(\lambda\dot{\gamma})^{1-n}},$$

where η_0 is the Newtonian viscosity, η_∞ is the infinite shear viscosity (usually assumed to be zero for the Cross model), λ is the natural time constant of the fluid ($1/\lambda$ is the critical shear rate at which the fluid changes from Newtonian to power-law behavior), and n is the flow behavior index in the power law regime.

Input File Usage: **VISCOSITY*, DEFINITION=CROSS

Abaqus/CAE Usage: The Cross model is not supported in Abaqus/CAE.

Herschel-Bulkley

The Herschel-Bulkley model can be used to describe the behavior of viscoplastic fluids that exhibit a yield response. The viscosity is expressed as

$$\eta = \begin{cases} \eta_0 & \text{if } \tau < \tau_0; \\ \frac{1}{\dot{\gamma}}\left(\tau_0 + k\left(\dot{\gamma}^n - (\tau_0/\eta_0)^n\right)\right) & \text{if } \tau \geq \tau_0. \end{cases}$$

Here τ_0 is the "yield" stress and η_0 is a penalty viscosity to model the "rigid-like" behavior in the very low strain rate regime ($\dot{\gamma} \leq \tau_0/\eta_0$), when the stress is below the yield stress, $\tau \leq \tau_0$. With increasing strain rates, the viscosity transitions into a power law model once the yield threshold is reached, $\tau \geq \tau_0$. The parameters k and n are the flow consistency and the flow behavior indexes in the power law regime, respectively.

In Abaqus/Standard this model is used to define the fluid behavior in fluid pipe and fluid pipe connector elements.

Input File Usage: **VISCOSITY*, DEFINITION=HERSCHEL-BULKLEY

Abaqus/CAE Usage: The Herschel-Bulkley model is not supported in Abaqus/CAE.

Bingham Plastic

The Bingham plastic model is a special case of the Herschel-Bulkley model. Bingham plastics correspond to the case where the flow behavior index in the power law regime, n, is equal to one.

This model is available only in Abaqus/Standard for fluid pipe and fluid pipe connector elements.

Input File Usage: **VISCOSITY*, DEFINITION=BINGHAM PLASTIC

Abaqus/CAE Usage: The Bingham plastic model is not supported in Abaqus/CAE.

Powell-Eyring

The Powell-Eyring model, which is derived from the theory of rate processes, is relevant primarily to molecular fluids but can be used in some cases to describe the viscous behavior of polymer solutions and viscoelastic suspensions over a wide range of shear rates. This model is available only in Abaqus/Explicit. The viscosity is expressed as

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)\frac{\sinh^{-1}(\lambda\dot{\gamma})}{\lambda\dot{\gamma}},$$

where η_0 is the Newtonian viscosity, η_∞ is the infinite shear viscosity, and λ represents a characteristic time of the measured system.

Input File Usage: **VISCOSITY*, DEFINITION=POWELL-EYRING

Abaqus/CAE Usage: The Powell-Eyring model is not supported in Abaqus/CAE.

Ellis-Meter

The Ellis-Meter model, available only in Abaqus/Explicit, expresses the viscosity in terms of the effective shear stress, $\tau = \sqrt{\frac{1}{2}\mathbf{S}:\mathbf{S}}$, as:

$$\eta = \eta_\infty + \frac{(\eta_0 - \eta_\infty)}{1+\left(\tau/\tau_{1/2}\right)^{(1-n)/n}},$$

where $\tau_{1/2}$ is the effective shear stress at which the viscosity is 50% between the Newtonian limit, η_0, and the infinite shear viscosity, η_∞, and n represents the flow index in the power law regime.

Input File Usage: **VISCOSITY*, DEFINITION=ELLIS-METER

Abaqus/CAE Usage: The Ellis-Meter model is not supported in Abaqus/CAE.

Tabular

In Abaqus/Explicit the viscosity can be specified directly as a tabular function of shear strain rate and temperature.

Input File Usage: **VISCOSITY*, DEFINITION=TABULAR

Abaqus/CAE Usage: Specifying the viscosity directly as a tabular function is not supported in Abaqus/CAE.

User-Defined

In Abaqus/Explicit you can specify a user-defined viscosity in user subroutine *VUVISCOSITY* (see *VUVISCOSITY*).

Input File Usage: **VISCOSITY*, DEFINITION=USER

Abaqus/CAE Usage: User-defined viscosity is not supported in Abaqus/CAE.

Temperature Dependence of Viscosity (Abaqus/Explicit Only)

The temperature-dependence of the viscosity of many polymer materials of industrial interest obeys a time-temperature shift relationship in the form:

$$\eta\left(\dot{\gamma},\theta\right)=a_T\left(\theta\right)\eta\left(a_T\left(\theta\right)\dot{\gamma},\theta_0\right),$$

where $a_T\left(\theta\right)$ is the shift function and θ_0 is the reference temperature at which the viscosity versus shear strain rate relationship is known. This concept for temperature dependence is usually referred to as thermorheologically simple (TRS) temperature dependence. In the Newtonian limit for low shear rates, when $\dot{\gamma}\rightarrow 0$, we have

$$\eta_0\left(\theta\right)=\lim_{\dot{\gamma}\rightarrow 0}\eta\left(\dot{\gamma}\theta\right)=a_T\left(\theta\right)\eta_0\left(\theta_0\right).$$

Thus, the shift function is defined as the ratio of the Newtonian viscosity at the temperature of interest to that of the chosen reference state: $a_T\left(\theta\right)=\eta_0\left(\theta\right)/\eta_0\left(\theta_0\right)$.

See *Thermorheologically Simple Temperature Effects* for a description of the different forms of the shift function available in Abaqus.

Input File Usage: Use the following options to define a thermorheologically simple (TRS) temperature-dependent viscosity:

**VISCOSITY*
**TRS*

Abaqus/CAE Usage: Defining a thermorheologically simple temperature-dependent viscosity is not supported in Abaqus/CAE.

Material Options

Material shear viscosity in Abaqus/Explicit must be used in combination with an equation of state to define the material's volumetric mechanical behavior (see *Equation of State*).

Elements

Material shear viscosity can be used with any solid (continuum) elements in Abaqus/Explicit except plane stress elements.

Material shear viscosity can be used only with fluid pipe elements and fluid pipe connector elements in Abaqus/Standard.

CZone Crush Stress

Products: Abaqus/Explicit Abaqus/CAE

References:

- **CRUSH STRESS*
- **CRUSH STRESS VELOCITY FACTOR*
- *CZone Analysis*

Overview

CZone crush stress:

- is required for the CZone crush modeling technique (see *CZone Analysis*);
- is specified as a material property but does not directly influence constitutive calculations associated with element integration points;
- limits the contact pressure magnitude at an actively crushing interface, such that the compressive stress of the underlying material approximately equals the crush stress at an active crush front;
- is relevant only in the context of shell elements; and
- can depend on the deformation rate at the crush front, material direction, temperature, or field variables.

CZone Crush Stress

The crush stress can be thought of as the stress of the crushing body at the crushing interface. The crush stress is specified as a material property but actually does not directly influence the constitutive calculations associated with element integration points; rather, the crush stress limits the contact pressure magnitude at an actively crushing interface as discussed in *CZone Analysis*.

The crush stress in the composite material can be a constant or can vary as a function of the angle, θ, between the local x-direction of the crushable element and the contact normal direction projected to the plane of the element.

Input File Usage: Use the following option to define a constant crush stress:

```
*CRUSH STRESS
crush stress
```

Use the following option to define the dependence of crush stress on the angle θ:

```
*CRUSH STRESS
crush stress, material angle in degrees
```

For example:

```
*CRUSH STRESS
90.0, 0.0
60.0, 45
90.0, 90
```

60.0, 135
90.0, 180

Abaqus/CAE Usage: Use the following option to define the crush stress:

Property module: material editor: **Mechanical->Crush Stress**

Crush Stress as a Function of a Field Variable

The crush stress in the composite material can vary as a function of a predefined field variable. This field variable can represent any physical or geometrical quantity that might affect the crush stress, such as curvature of the laminate.

Input File Usage: Use the following option to define the crush stress as a function of a field variable:

**CRUSH STRESS*, DEPENDENCIES

For example, the following definition can be used to define a crush stress that increases by 10% in curved regions of the model versus flat regions. The example assumes that curvature information is defined by the second field variable (`FV2`), which has been initialized with a value of one at the nodes in the curved regions of the model and a value of zero in the flat regions. Field variable dependency is evaluated only at the beginning of the analysis. Subsequent changes in the field variable will have no effect on the crush stress of the material.

**CRUSH STRESS*, DEPENDENCIES=5
90.0, 0.0, , , 0.0
60.0, 45.0, , , 0.0
90.0, 90.0, , , 0.0
60.0, 135.0, , , 0.0
90.0, 180.0, , , 0.0
99.0, 0.0, , , 1.0
66.0, 45.0, , , 1.0
99.0, 90.0, , , 1.0
66.0, 135.0, , , 1.0
99.0, 180.0, , , 1.0

Abaqus/CAE Usage: Use the following option to define the crush stress:

Property module: material editor: **Mechanical->Crush Stress**

Toggle on **Use temperature-dependent data** to define the crush stress as a function of temperature, and select the **Number of field variables** to define the crush stress as a function of field variables.

Crush Stress as a Function of Velocity

The crush stress in the composite material can depend on the deformation rate at the crush front. The crush stress velocity factor is used to scale the crush stress based upon velocity. The scaling factor is calculated from a user-supplied table of scaling factor versus relative surface velocity.

The final crush stress is the velocity dependence factor multiplied by the original crush stress. No extrapolation of the supplied data is performed for velocities that lie outside the maximum or minimum value supplied, as shown in *Figure 1*.

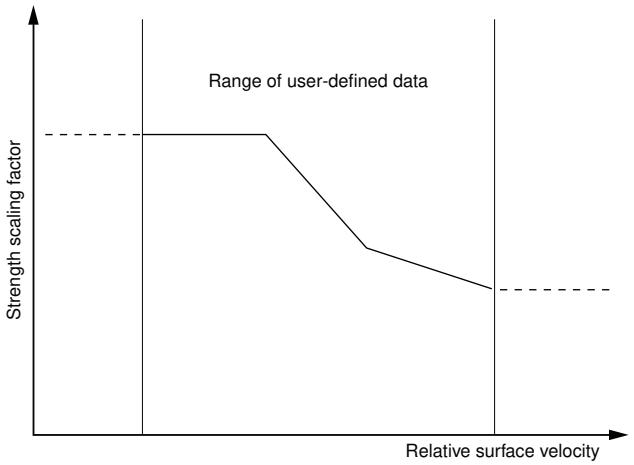


Figure 1: Crush stress is held constant outside of the supplied data range.

Input File Usage: Use the following option to specify the crush stress velocity factor:

**CRUSH STRESS VELOCITY FACTOR*

For example:

**CRUSH STRESS VELOCITY FACTOR*
1.5, 0.0
1.5, 1.0
1.25, 10.0
1.0, 100.

Abaqus/CAE Usage: Use the following option to define the crush stress velocity factor:

Property module: material editor: **Mechanical->Crush Stress**: **Suboptions->Velocity Factor**

Elements

CZone crush stress is relevant only for shell elements in Abaqus/Explicit.

Heat Transfer Properties

In this section:

- *About Thermal Properties*
- *Conductivity*
- *Specific Heat*
- *Latent Heat*

About Thermal Properties

Overview

Thermal behavior of a material can be used in heat transfer and thermal stress analyses.

The following properties describe the thermal behavior of a material and can be used in heat transfer and thermal stress analyses (see *About Heat Transfer Analysis Procedures*):

Thermal conductivity

When heat flows by conduction, the thermal conductivity must be defined (*Conductivity*).

Specific heat

In transient heat transfer analyses as well as adiabatic stress analyses the specific heat of a material must be defined (*Specific Heat*).

Latent heat

When a material changes phase, the change in internal energy can be significant. The amount of energy liberated or absorbed can be defined by specifying a latent heat for each phase change a material undergoes (*Latent Heat*).

References:

Conductivity

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *About Thermal Properties*
- **CONDUCTIVITY*
- *Specifying thermal conductivity*

Overview

A material's thermal conductivity:

- must be defined for *Uncoupled Heat Transfer Analysis*, *Fully Coupled Thermal-Stress Analysis*, and *Coupled Thermal-Electrical Analysis*;
- can be linear or nonlinear (by defining it as a function of temperature);
- can be isotropic, orthotropic, or fully anisotropic; and
- can be specified as a function of temperature and field variables.

Directional Dependence of Thermal Conductivity

Isotropic, orthotropic, or fully anisotropic thermal conductivity can be defined. Only isotropic thermal conductivity can be defined for an incompressible fluid dynamic analysis that includes an energy equation. For orthotropic or anisotropic thermal conductivity, a local orientation (*Orientations*) must be used to specify the material directions used to define the conductivity.

Isotropic Conductivity

For isotropic conductivity, only one value of conductivity is required at each temperature and field variable value. Isotropic conductivity is the default.

Input File Usage: **CONDUCTIVITY*, TYPE=ISO

Abaqus/CAE Usage: Property module: material editor: **Thermal->Conductivity**: **Type: Isotropic**

Orthotropic Conductivity

For orthotropic conductivity three values of conductivity (k_{11}, k_{22}, k_{33}) are required at each temperature and field variable value.

Input File Usage: **CONDUCTIVITY*, TYPE=ORTHO

Abaqus/CAE Usage: Property module: material editor: **Thermal->Conductivity**: **Type: Orthotropic**

Transversely Isotropic Conductivity

A special subclass of orthotropy is *transverse isotropy*, which is characterized by a plane of isotropy at every point in the material. Abaqus assumes the 2–3 plane to be the plane of isotropy at every point; therefore, $k_{22} = k_{33}$ and only two values of conductivity (k_{11}, k_{22}) are required at each temperature and field variable value.

Input File Usage: **CONDUCTIVITY*, TYPE=TRANSVERSELY ISOTROPIC

Abaqus/CAE Usage: Computing conductivity for a transversely isotropic material is not supported in Abaqus/CAE.

Anisotropic Conductivity

For fully anisotropic conductivity six values of conductivity (k_{11}, k_{12}, k_{22}, k_{13}, k_{23}, k_{33}) are required at each temperature and field variable value.

Input File Usage: **CONDUCTIVITY*, TYPE=ANISO

Abaqus/CAE Usage: Property module: material editor: **Thermal->Conductivity**: **Type: Anisotropic**

Conductivity for a Short-Fiber Reinforced Composite

Abaqus/Standard computes conductivity of a short-fiber reinforced composite using orientation averaging of a unidirectional composite with transversely isotropic conductivity (k_L, k_T):

$$k_{11} = (k_L - k_T) \cdot a_1 + k_T,$$
$$k_{22} = (k_L - k_T) \cdot a_2 + k_T,$$
$$k_{33} = (k_L - k_T) \cdot a_3 + k_T,$$

where a_1, a_2, and a_3 are the maximum, mid, and minimum principal values of a second-order orientation tensor describing fiber dispersion at the material point.

Input File Usage: **CONDUCTIVITY*, TYPE=ORTHO
**FIBER DISPERSION*

Abaqus/CAE Usage: Computing conductivity for a short-fiber reinforced composite is not supported in Abaqus/CAE.

Elements

Thermal conductivity is active in all heat transfer, coupled temperature-displacement, coupled thermal-electrical-structural, and coupled thermal-electrical elements in Abaqus. Thermal conductivity for a short-fiber reinforced composite is supported only in Abaqus/Standard.

Specific Heat

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *About Thermal Properties*
- **SPECIFIC HEAT*
- *Defining specific heat*

Overview

A material's specific heat:

- is required for transient *Uncoupled Heat Transfer Analysis*, transient *Fully Coupled Thermal-Stress Analysis*, transient *Coupled Thermal-Electrical Analysis*, and *Adiabatic Analysis*;
- must appear in conjunction with a density definition (see *Density*);
- can be linear or nonlinear (by defining it as a function of temperature); and
- can be specified as a function of temperature and/or field variables.

Defining Specific Heat

The specific heat of a substance is defined as the amount of heat required to increase the temperature of a unit mass by one degree. Mathematically, this physical statement can be expressed as:

$$c = \frac{\delta Q}{d\theta} = \theta \left(\frac{ds}{d\theta} \right),$$

where δQ is the infinitessimal heat added per unit mass and s is the entropy per unit mass. Since heat transfer depends on the conditions encountered during the whole process (a path function), it is necessary to specify the conditions used in the process to unambiguously characterize the specific heat. Thus, a process where the heat is supplied keeping the volume constant defines the specific heat as:

$$c_v = \left(\frac{\delta Q}{d\theta} \right)\Big|_v = \theta \left(\frac{\partial s}{\partial \theta} \right)\Big|_v = \left(\frac{\partial u}{\partial \theta} \right)\Big|_v,$$

where u is the internal energy per unit mass.

Whereas, a process where the heat is supplied keeping the pressure constant defines the specific heat as:

$$c_p = \left(\frac{\delta Q}{d\theta} \right)\Big|_p = \theta \left(\frac{\partial s}{\partial \theta} \right)\Big|_p = \left(\frac{\partial h}{\partial \theta} \right)\Big|_p,$$

where $h = u + pv$ is the enthalpy per unit mass. In general, the specific heats are functions of temperature. For solids and liquids, c_v and c_p are equivalent; thus, there is no need to distinguish between them. When possible, large changes in internal energy or enthalpy during a phase change should be modeled using *Latent Heat* instead of specific heat.

Defining Constant-Volume Specific Heat

The specific heat per unit mass is given as a function of temperature and field variables. By default, specific heat at constant volume is assumed.

Input File Usage: **SPECIFIC HEAT*

Abaqus/CAE Usage: Property module: material editor: **Thermal->Specific Heat**; **Type**: **Constant Volume**

Elements

Specific heat effects can be defined for all heat transfer, coupled thermal-electrical-structural, coupled temperature-displacement, and coupled thermal-electrical elements in Abaqus. Specific heat can also be defined for stress/displacement elements for use in adiabatic stress analysis.

Specific heat must be defined for all transient thermal analyses even if the only elements in the model are user-defined elements (*User-Defined Elements*), in which case a dummy specific heat must be specified.

Latent Heat

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About the Material Library*
- *About Thermal Properties*
- **LATENT HEAT*
- *Specifying latent heat data*

Overview

A material's latent heat:

- models large changes in internal energy during phase change of a material;
- is active only during transient heat transfer, coupled thermal-stress, coupled thermal-electrical-structural and coupled thermal-electrical analysis in Abaqus (see *About Heat Transfer Analysis Procedures*);
- must appear in conjunction with a density definition (see *Density*); and
- always makes an analysis nonlinear.

Defining Latent Heat

Latent heat effects can be significant and must be included in many heat transfer problems involving phase change. When latent heat is given, it is assumed to be in addition to the specific heat effect (see *Uncoupled heat transfer analysis* for details).

The latent heat is assumed to be released over a range of temperatures from a lower (solidus) temperature to an upper (liquidus) temperature. To model a pure material with a single phase change temperature, these limits can be made very close.

As many latent heats as are necessary can be defined to model several phase changes in the material. Latent heat can be combined with any other material behavior in Abaqus, but it should not be included in the material definition unless necessary; it always makes the analysis nonlinear.

Direct Data Specification

If the phase change occurs within a known temperature range, the solidus and liquidus temperatures can be given directly. The latent heat should be given per unit mass.

Input File Usage: **LATENT HEAT*

Abaqus/CAE Usage: Property module: material editor: **Thermal->Latent Heat**

Defining a Smooth Latent Heat Transition

Abaqus/Standard can calculate a smooth transition from solidus to liquidus temperature using

$$U_\theta = (LH)\xi^3\left(10 - 15\xi + 6\xi^2\right) \quad \text{for } \theta_S \le \theta \le \theta_L,$$

where U_θ is the internal energy, LH is the latent heat, and $\xi = (\theta - \theta_S) / (\theta_L - \theta_S)$. The above function is such that the first and second derivatives of U_θ are zero at θ_S and θ_L. This definition is intended to ramp up from one value to another. In addition, you can extend the solidus-liquidus temperature interval using a scale factor as follows:

$$\theta_{S_new} = \frac{1}{2}[\theta_S + \theta_L - (\theta_L - \theta_S) \cdot scale\ factor]$$

$$\theta_{L_new} = \frac{1}{2}[\theta_S + \theta_L + (\theta_L - \theta_S) \cdot scale\ factor]$$

Input File Usage: **LATENT HEAT*, SMOOTH

LH, θ_S, θ_L, *scale factor*

Abaqus/CAE Usage: Defining a smooth latent heat transition is not supported in Abaqus/CAE.

User Subroutine

In some cases it may be necessary to include a kinetic theory for the phase change to model the effect accurately in Abaqus; for example, the prediction of crystallization in a polymer casting process. In such cases you can model the process in considerable detail using solution-dependent state variables (*About User Subroutines and Utilities*) and user subroutine *HETVAL* (Abaqus/Standard) or *VHETVAL* (Abaqus/Explicit).

Input File Usage: Use the following options:

**HEAT GENERATION*
**DEPVAR*

Abaqus/CAE Usage: Property module: material editor:
Thermal->Heat Generation
General->Depvar

Elements

Latent heat effects can be used in all diffusive heat transfer, coupled temperature-displacement, coupled thermal-electrical-structural and coupled thermal-electrical elements in Abaqus but cannot be used with convective heat transfer elements. Strong latent heat effects are best modeled with first-order or modified second-order elements, which use integration methods designed to provide accurate results for such cases.

See *Freezing of a square solid: the two-dimensional Stefan problem* for an example of a heat conduction problem involving latent heat.

Curing Processes in Polymers

Products: Abaqus/Standard

References:

- **CURE GLASS TRANSITION TEMPERATURE*
- **CURE HEAT GENERATION*
- **CURE KINETICS*
- **CURE MAX CONVERSION*
- **CURE SHRINKAGE*
- *Modeling the Cure Process in Thermosetting Polymers*

Overview

The cure modeling capabilities:

- are intended to model the curing process in adhesives and other thermosetting polymer materials;
- predict the degree of cure, volumetric heat generation, and shrinkage strain due to curing reactions;
- allow you to specify a maximum value of the degree of cure;
- are active in transient procedures that use elements with displacement degrees of freedom, in the transient heat transfer procedure, and in the transient coupled thermal-electrical procedure; and
- are intended for use with existing elastic and viscoelastic behaviors that describe the mechanical response of the material as a function of the degree of cure.

The cure modeling capabilities described in this section are based on dedicated cure modeling options and output variables. These capabilities are more general and are intended to supersede those available as a special-purpose material modeling capability based on built-in user-defined material options (*Modeling the Cure Process in Thermosetting Polymers*).

Modeling Curing Processes

Curing processes are essential to the manufacturing of products that use thermosetting polymers (such as epoxy resins) to bond components. The use of epoxies and other cured structural adhesives is common in many industries. As a result of the curing reaction, chemical shrinkage strains and residual stresses develop, which can result in damage to the adherents or warpage of the bonded assembly. The Abaqus cure modeling capability enables you to analyze curing processes, including the reaction kinetics, heat generation, shrinkage strain development, and the evolution of mechanical properties. The model is based on the work of *Lindeman et al. (2021)* and *Li et al. (2004)*. The mechanical response can include both elastic and viscoelastic effects. You can define the elastic, viscoelastic, and thermal expansion properties of the material as functions of the degree of cure and temperature.

Reaction Kinetics

The degree of cure (or conversion) of a material is characterized by a normalized quantity, α, with a value changing from 0 (corresponding to the uncured state; that is, no bonds) to 1 (corresponding to the fully cured material). Since curing is an irreversible process, the value of the degree of cure that Abaqus computes never decreases. The value either increases or remains constant. Abaqus provides four methods of specifying the cure

kinetics: the Kamal equations, the Grindling equation, a conversion rate table, and a user-defined form using user subroutine *UCURE*.

Kamal model

The Kamal equation provides a well-accepted description of the cure reaction kinetics that is known to produce accurate results, particularly for epoxy resins. It is given by the following rate form:

$$\dot{\alpha} = \sum_{i=1}^{N} Z_i e^{-\frac{E_i}{R\left(\theta-\theta^Z\right)}} \left(b_i + \alpha^{m_i}\right)\left(\alpha_{\max} - \alpha\right)^{n_i},$$

where

N
: is the number of terms,

Z_i
: are rate constants,

E_i
: are activation energies,

m_i and n_i
: are reaction constants,

θ^Z
: is the absolute zero on the temperature scale used, and

R
: is the universal gas constant.

The material constants, b_i, are introduced to allow a nonzero initial conversion rate by setting at least one of them to a small positive value. Alternatively, you can specify a nonzero initial conversion value, $\alpha|_{t=0}$, using initial conditions (see *Initial Conditions*) or set at least one of the parameters, m_i, to zero. If you do not define b_i, $\alpha|_{t=0}$, or $m_i = 0$, the degree of cure (conversion) remains equal to zero throughout the analysis.

The constant, $\alpha_{\max}$, controls the maximum degree of cure. By default, $\alpha_{\max} = 1$. You can define a different value, as described in *Maximum Conversion*.

Input File Usage:

Use the following options to define the conversion rate in the Kamal equation with constant coefficients:

**PHYSICAL CONSTANTS*, ABSOLUTE ZERO= ^{Z}
**MATERIAL*
**CURE KINETICS*, DEFINITION=KAMAL

Use the following options to define the conversion rate in the Kamal equation with coefficients that might be functions of temperature and/or predefined field variables:

**PHYSICAL CONSTANTS*ABSOLUTE ZERO= ^{Z}
**MATERIAL*
**CURE KINETICS*, DEFINITION=KAMAL, NUMBER TERMS=*n*

Grindling model

The Kamal equation describes cure kinetics very well at cure temperatures higher than the glass transition temperature, where the reaction kinetics is chemically controlled. However, during isothermal curing processes the curing temperature can fall below the glass transition temperature, θ_g, even if it was higher than θ_g at the beginning of the curing process. This is because the glass transition temperature depends strongly on the degree of cure, and its value changes during the curing process. If the curing temperature falls below glass transition temperature, premature vitrification takes place and the material undergoes a transition from a rubbery to a glassy state, which slows the curing process rate considerably because the reaction kinetics becomes diffusion controlled. Although the Kamal model can account for this behavior by limiting the maximum value of conversion by specifying $\alpha_{\max} < 1.0$ for temperatures below the glass transition temperature, the Grindling model can predict gradual transition from the chemically controlled to the diffusion-controlled reaction rates and produce accurate results in this case. The conversion rate for the Grindling model is given by:

$$\dot{\alpha} = K_1(\alpha_{\max} - \alpha)^{n_1} + K_{eff}\alpha^m(\alpha_{\max} - \alpha)^{n_2}$$

where n_1, n_2, and m are material parameters, K_1 is reaction rate, which is assumed to have the Arrhenius-type form:

$$K_1 = A_1 e^{-\left(\frac{E_1}{R\left(\theta-\theta^Z\right)}\right)},$$

where A_1 is a material parameter and E_1 is the activation energy.

The effective rate coefficient, K_{eff}, is used to switch between the chemically and the diffusion-controlled reaction, and it is given by

$$\frac{1}{K_{eff}} = \frac{1}{K_D} + \frac{1}{K_2},$$

where K_2 and K_D are the chemically and diffusion controlled rates, respectively. The rate coefficient, K_2, has the Arrhenius-type form similar to the rate coefficient K_1:

$$K_2 = A_2 e^{-\left(\frac{E_2}{R\left(\theta-\theta^Z\right)}\right)},$$

where A_2 is a material parameter and E_2 is the activation energy.

The diffusion-controlled rate coefficient, K_D, is computed from the following relations depending on the value of the glass transition temperature:

$$\text{If } \theta > (\theta_g + \Delta\theta_g): \quad K_D = K_D^{\theta=\theta_g} e^{\frac{c_1\Delta\theta_g\left(c_2+\Delta\theta_g\right)\left(\theta-\theta^Z\right)-c_1c_2\left(\theta_g-\theta^Z+\Delta\theta_g\right)\left(\theta_g-\theta+\Delta\theta_g\right)}{\left(c_2+\Delta\theta_g\right)^2\left(\theta-\theta^Z\right)}}$$

$$\text{If } \theta_g \le \theta \le (\theta_g + \Delta\theta_g): \quad K_D = K_D^{\theta=\theta_g} e^{\frac{c_1(\theta-\theta_g)}{c_2+\theta-\theta_g}}$$

$$\text{If } \theta < \theta_g: \quad K_D = K_D^{\theta=\theta_g} e^{-\left(\frac{c_1}{c_2}\right)\left(\frac{\theta_g-\theta^Z}{\theta-\theta^Z}\right)\left(\theta_g-\theta\right)},$$

where c_1, c_2, and $\Delta\theta_g$ are material parameters and $K_D^{\theta=\theta_g}$ is the diffusion-controlled rate coefficient at the temperature equal to the glass transition temperature.

The diffusion-controlled rate, K_D, strongly depends on the glass transition temperature, which in turn depends on the degree of cure and possibly other quantities, such as temperature. Therefore, the glass transition temperature must be specified when using the Grindling model (see *Glass transition temperature*).

Input File Usage: Use the following options to define the conversion rate in the Grindling equation:

*PHYSICAL CONSTANTS, ABSOLUTE ZERO= Z
*MATERIAL
*CURE KINETICS, DEFINITION=GRINDLING

Glass transition temperature

The glass transition temperature plays an important role during the curing processes. The curing kinetics models might require the glass transition temperature to be specified, and the material properties might depend on it.

The glass transition temperature is required for the Grindling model. You must specify the glass transition temperature for other cure kinetics models if you want to request glass transition temperature output or the output of the difference between the glass transition temperature and the current temperature (see *Output*). In addition, you usually take this approach if you want to consider the dependency of material properties on the difference between the glass transition temperature and the current temperature (see *Cure-Dependent Material Properties*).

Abaqus provides two methods to define the glass transition temperature: the DiBenedetto model and the tabular form. The DiBenedetto model is widely used and defines θ_g as a function of the degree of cure in the following form:

$$\frac{\theta_g-\theta_{g,0}}{\theta_{g,\infty}-\theta_{g,0}}=\frac{\lambda\alpha}{1-(1-\lambda)\alpha},$$

where λ is a material parameter ($\lambda > 0$); $\theta_{g,0}$ is the glass transition temperature in the uncured state; and $\theta_{g,\infty}$ is the glass transition temperature in the fully cured state.

Alternatively, you can use a tabular definition to specify θ_g as a piecewise linear function of degree of cure, temperature, and field variables.

Input File Usage: Use the following option to specify the glass transition temperature using the DiBenedetto equation:

*CURE GLASS TRANSITION TEMPERATURE, DEFINITION=DIBENEDETTO

Use the following option to specify the glass transition temperature in tabular form:

*CURE GLASS TRANSITION TEMPERATURE, DEFINITION=TABULAR

Conversion rate in tabular form

You can also specify the rate of conversion, $\dot{\alpha}$, in tabular format. This format allows you to express the rate as a function of conversion and, optionally, temperature and field variables:

$$\dot{\alpha}=f(\alpha,\theta,FVs).$$

Input File Usage: Use the following options to define the conversion rate in tabular form:

*MATERIAL
*CURE KINETICS, DEFINITION=TABULAR

User-defined conversion rate

An alternative method to define the cure kinetics is to specify the degree of cure rate in user subroutine *UCURE*. In addition to the degree of cure rate, you must also specify the values of the derivatives of the degree of cure rate with respect to the temperature and the degree of cure. Optionally, you can also specify the number of property values needed as data in the user subroutine as well as the number of solution-dependent variables (see *About User Subroutines and Utilities*).

Input File Usage: Use the following options to specify the degree of cure rate in user subroutine *UCURE*:

**MATERIAL*
**CURE KINETICS*, DEFINITION=USER, PROPERTIES=*n*

Maximum Conversion

In general, the degree of cure can reach a maximum value of 1 (corresponding to a fully cured material). However, at lower temperatures, the reaction might slow down considerably, and a fully cured state might not be reached (corresponding to $\alpha_{\max} < 1$). You can specify the maximum conversion as a function of temperature. The default is $\alpha_{\max} = 1$.

Input File Usage: Use the following option to specify the maximum conversion:

**CURE MAX CONVERSION*

Cure Heat Generation

Curing reactions are irreversible, exothermic processes that are activated by mixing or heating. The amount of heat released per unit volume per unit time is given by the relationship:

$$q = \rho_0 Q \dot{\alpha},$$

where ρ_0 is the density, Q is the specific heat of the reaction, and $\dot{\alpha} = \frac{d\alpha}{dt}$ is the conversion rate.

Input File Usage: Use the following options to define the volumetric heat generation rate:

**MATERIAL*
**DENSITY*
**CURE HEAT GENERATION*

Cure Shrinkage Strain

During the curing process, the material undergoes permanent shrinkage. The shrinkage is due to cross-linking because the formation of bonds moves the atoms closer together than in the unbonded state. The shrinkage and thermal strains that develop during the curing process result in residual stresses that might cause warpage of the final product. Predicting residual stress distributions is often one of the main reasons for performing numerical simulations. You can model the thermal strain using Abaqus capabilities (see *Thermal Expansion*). The cure shrinkage strain is expressed in the following general rate form:

$$\dot{\boldsymbol{\varepsilon}}^s = -\boldsymbol{\gamma}\dot{\alpha},$$

where $\boldsymbol{\varepsilon}^s$ is the cure shrinkage strain, $\boldsymbol{\gamma}$ is the shrinkage coefficient matrix, and $\dot{\alpha}$ is the conversion rate. Abaqus supports four forms of cure shrinkage coefficients: volumetric, isotropic, orthotropic, and anisotropic. You can use the orthotropic and anisotropic forms only with materials where the material directions are defined with local orientations (see *Orientations*).

Volumetric Cure Shrinkage Strain

The volumetric cure shrinkage strain is computed from the following rate equation:

$$\dot{\boldsymbol{\varepsilon}}^s = -\frac{1}{3}\gamma_{vol}\mathbf{I}\dot{\alpha}(t).$$

In this case, you need to specify only a single coefficient, γ_{vol}, as a function of temperature and field variables.

Input File Usage: Use the following option to specify the volumetric shrinkage coefficient:

**CURE SHRINKAGE*, TYPE=VOL

Isotropic Cure Shrinkage Strain

The isotropic cure shrinkage strain is computed from the following rate equation:

$$\dot{\boldsymbol{\varepsilon}}^s = -\gamma \mathbf{I} \dot{\alpha}(t).$$

In this case, you need to specify only a single coefficient, γ, as a function of temperature and field variables.

Input File Usage: Use the following option to specify the isotropic shrinkage coefficient:

**CURE SHRINKAGE*, TYPE=ISO

Orthotropic Cure Shrinkage Strain

The orthotropic cure shrinkage strain is computed from the following rate equation:

$$\begin{pmatrix} \dot{\varepsilon}^s_{11} & 0 & 0 \\ 0 & \dot{\varepsilon}^s_{22} & 0 \\ 0 & 0 & \dot{\varepsilon}^s_{33} \end{pmatrix} = - \begin{pmatrix} \gamma_{11} & 0 & 0 \\ 0 & \gamma_{22} & 0 \\ 0 & 0 & \gamma_{33} \end{pmatrix} \dot{\alpha}(t).$$

In this case, you specify the coefficients in the principal material directions as functions of temperature and field variables.

Input File Usage: Use the following option to specify the orthotropic shrinkage coefficients:

**CURE SHRINKAGE*, TYPE=ORTHO

Anisotropic Cure Shrinkage Strain

The anisotropic cure shrinkage strain is computed from the following rate equation:

$$\begin{pmatrix} \dot{\varepsilon}^s_{11} & \dot{\varepsilon}^s_{12} & \dot{\varepsilon}^s_{13} \\ \dot{\varepsilon}^s_{12} & \dot{\varepsilon}^s_{22} & \dot{\varepsilon}^s_{23} \\ \dot{\varepsilon}^s_{13} & \dot{\varepsilon}^s_{23} & \dot{\varepsilon}^s_{33} \end{pmatrix} = - \begin{pmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{12} & \gamma_{22} & \gamma_{23} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} \end{pmatrix} \dot{\alpha}(t).$$

In this case, you must specify all six components of the shrinkage coefficient matrix, γ_{ij}, as functions of temperature and field variables.

Input File Usage: Use the following option to specify the anisotropic shrinkage coefficients:

**CURE SHRINKAGE*, TYPE=ANISO

Example: Defining the Cure Properties of a Material

This example illustrates defining the cure modeling capabilities in combination with a typical viscoelastic material definition. The material properties are defined as a function of the degree of cure and the difference between the

glass transition temperature and the current temperature using field variable dependency. Field variables 1 and 2 are used for the example.

```
*HEADING
*MATERIAL, NAME="CURE_MATERIAL_matName"
*DENSITY
Data lines to specify mass density
*ELASTIC, DEPENDENCIES=2
Data lines to specify linear elastic parameters
*VISCOELASTIC
Data lines to specify viscoelastic parameters
*TRS, DEFINITION=TABULAR, DEPENDENCIES=1
Data line to specify logarithm of the shift function
*EXPANSION
Data line to specify thermal expansion coefficients
*CURE KINETICS, DEFINITION=KAMAL
Data line to specify the Kamal model parameters
*CURE MAX CONVERSION
Data line to specify maximum value of the degree of cure
*CURE SHRINKAGE, TYPE=ISO
Data line to specify the isotropic shrinkage coefficients
*CURE HEAT GENERATION
Data lines to define the volumetric heat generation rate
*CONDUCTIVITY
Data line to specify the thermal conductivity
*SPECIFIC HEAT
Data lines to specify the specific heat
*USER DEFINED FIELD, TYPE=SPECIFIED
1, DOC
2, TGTDIFF
*STEP
*COUPLED TEMPERATURE-DISPLACEMENT
Data line to control incrementation and to specify the total time
*END STEP
```

Initial Conditions

You can specify a nonzero initial value of the degree of cure of the material. This approach is commonly used to define a small value to trigger the start of the curing reaction.

Input File Usage: Use the following option to define initial conditions for the degree of cure:

**INITIAL CONDITIONS*, TYPE=CURE

Material Options

The mechanical response of the material in the uncured state is typically viscoelastic and, in the fully cured state, is often viscoelastic. Therefore, the cure modeling capabilities are typically used in combination with the small-strain or finite-strain viscoelastic modeling capabilities already available in Abaqus. When these capabilities are used in conjunction with hyperelastic materials, only the isotropic and volumetric cure shrinkage types are supported.

Cure-Dependent Material Properties

The elastic and viscoelastic material properties can change considerably as the cross-linking progresses and the material transitions from the uncured to the cured state. In general, these material properties depend on the degree of cure, α, and the difference between the glass transition temperature and the current temperature, $(\theta_g - \theta)$.

You can consider the dependency of material properties on both α and $(\theta_g - \theta)$ in the model by introducing field variables associated with these quantities and specifying the material properties as a function of these field variables (see *Specifying Material Data as Functions of Solution-Dependent Variables*). *Hedegaard et al. (2021)* describe a testing procedure for measuring viscoelastic properties as a function of temperature and conversion level.

Input File Usage: Use the following option to associate both the degree of cure and the difference between the glass transition temperature and temperature with two different field variables:

**USER DEFINED FIELD*, TYPE=SPECIFIED
FV_number_1, DOC
FV_number_2, TGTDIFF

Elements

You can use the cure modeling capabilities with any stress/displacement, coupled temperature-displacement, heat transfer, or coupled thermal-electrical elements in Abaqus (see *Choosing the Appropriate Element for an Analysis Type*). However, the cure shrinkage strain is ignored when elements without displacement degrees of freedom are used (that is, heat transfer and coupled thermal-electrical elements).

Procedures

The cure modeling capabilities are active in the following procedures:

- Transient static analysis (*Quasi-Static Analysis*)
- Implicit dynamic analysis (*Implicit Dynamic Analysis Using Direct Integration*)
- Transient fully coupled temperature-displacement analysis (*Fully Coupled Thermal-Stress Analysis*)
- Transient fully coupled thermal-electrical-structural analysis (*Fully Coupled Thermal-Electrical-Structural Analysis*)
- Transient heat transfer analysis (*Uncoupled Heat Transfer Analysis*)
- Transient coupled thermal-electrical analysis (*Coupled Thermal-Electrical Analysis*)

The capability is inactive during any other procedure, and the value of the degree of cure remains constant.

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning for the cure material model:

DOC Degree of cure (conversion), α.

DOCR Rate of degree of cure, $\dot{\alpha}$.

DDOCRDTEMP Derivative of the rate of degree of cure with respect to temperature.

CURESE Cure shrinkage strain, ε^s.

TG Glass transition temperature, θ_g.

TGTDIFF The difference between glass transition temperature and the current temperature, $\theta_g - \theta$.

References

- Grindling, J., Simulation zur Verarbeitung von reaktiven Nonpost-cure-Epoxidharz-Systemen im Druckgelieren und konventionellen Vergiessen *Dissertation, Shaker, Aachen*, 2006.
- Hedegaard, A., E. Breedlove, S. Carpenter, V. Jusuf, C. Li, and D. Lindeman, Time-Temperature-Cure Superposition (TTCS) Methods for Determining Viscoelasticity of Structural Adhesives During Curing 44th Meeting of the Adhesion Society, 2021.
- Li, C., Y. Wang, and J. Mason, "The Effects of Curing History on Residual Stresses in Bone Cement During Hip Arthroplasty," *Journal of Biomedical Materials Research*, vol. 70B, pp. 30–36, 2004.
- Lindeman, D., S. Carpenter, and C. Li, Residual Stress Development During Curing of Structural Adhesives: Experimental Characterization and Modeling 44th Meeting of the Adhesion Society, 2021.

Acoustic Medium

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *Coupled Acoustic-Structural Analysis*
- *Acoustic and Shock Loads*
- *About the Material Library*
- *Initial Conditions*
- **ACOUSTIC MEDIUM*
- **DENSITY*
- **INITIAL CONDITIONS*
- *Defining an acoustic medium*

Overview

An acoustic medium:

- is used to model sound propagation problems;
- can be used in a purely acoustic analysis or in a coupled acoustic-structural analysis such as the calculation of shock waves in a fluid or noise levels in a vibration problem;
- is an elastic medium (usually a fluid) in which stress is purely hydrostatic (no shear stress) and pressure is proportional to volumetric strain;
- is specified as part of a material definition;
- must appear in conjunction with a density definition (see *Density*);
- can include fluid cavitation in Abaqus/Explicit when the absolute pressure drops to a limit value;
- can be defined as a function of temperature and/or field variables;
- can include dissipative effects;
- can model small pressure changes (small amplitude excitation);
- can model waves in the presence of steady underlying flow of the medium; and
- is active only during dynamic analysis procedures (*About Dynamic Analysis Procedures*).

Defining an Acoustic Medium

The equilibrium equation for small motions of a compressible, inviscid fluid flowing through a resisting matrix material is taken to be

$$\frac{\partial p}{\partial \mathbf{x}} + \gamma \dot{\mathbf{u}}^f + \rho_f \ddot{\mathbf{u}}^f = 0,$$

where p is the dynamic pressure in the fluid (the pressure in excess of any initial static pressure), $\mathbf{x}$ is the spatial position of the fluid particle, $\dot{\mathbf{u}}^f$ is the fluid particle velocity, $\ddot{\mathbf{u}}^f$ is the fluid particle acceleration, ρ_f is the density of the fluid, and γ is the "volumetric drag" (force per unit volume per velocity) caused by the fluid flowing through the matrix material. The d'Alembert term has been written without convection on the assumption that

there is no steady flow of the fluid, which is usually considered to be sufficiently accurate for steady fluid velocities up to Mach 0.1.

The constitutive behavior of the fluid is assumed to be inviscid and compressible, so that the bulk modulus of an acoustic medium relates the dynamic pressure in the medium to the volumetric strain by

$$p = -\mathbf{K}_f \varepsilon_V,$$

where $\varepsilon_V = \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}$ is the volumetric strain. Both the bulk modulus $\mathbf{K}_f$ and the density ρ_f of an acoustic medium must be defined.

The bulk modulus $\mathbf{K}_f$ can be defined as a function of temperature and field variables but does not vary in value during an implicit dynamic analysis using the subspace projection method (*Implicit Dynamic Analysis Using Direct Integration*) or a direct-solution steady-state dynamic analysis (*Direct-Solution Steady-State Dynamic Analysis*); for these procedures the value of the bulk modulus at the beginning of the step is used.

Input File Usage: Use both of the following options to define an acoustic medium:

**ACOUSTIC MEDIUM*, BULK MODULUS
**DENSITY*

Abaqus/CAE Usage: Property module: material editor:
Other->Acoustic Medium: **Bulk Modulus**
General->Density

Volumetric Drag

Dissipation of energy (and attenuation of acoustic waves) might occur in an acoustic medium due to a variety of factors. Such dissipation effects are phenomenologically characterized in the frequency domain by the imaginary part of the propagation constant, which gives an exponential decay in amplitude as a function of distance. In Abaqus the simplest way to model this effect is through a "volumetric drag coefficient," γ (force per unit volume per velocity).

In frequency-domain procedures, γ might be frequency dependent. γ can be entered as a function of frequency—$\gamma(f)$, where f is the frequency in cycles per time (usually Hz)—in addition to temperature and/or field variables only when the acoustic medium is used in a steady-state dynamics procedure. If the acoustic medium is used in a direct-integration dynamic procedure (including Abaqus/Explicit), the volumetric drag coefficient is assumed to be independent of frequency and the first value entered for the current temperature and/or field variable is used.

In all procedures except direct steady-state dynamics the gradient of γ/ρ_f is assumed to be small.

Input File Usage: **ACOUSTIC MEDIUM*, VOLUMETRIC DRAG

Abaqus/CAE Usage: Property module: material editor: **Other->Acoustic Medium**: **Volumetric Drag**: **Include volumetric drag**

Porous Acoustic Material Models

Porous materials are commonly used to suppress acoustic waves; this attenuating effect arises from a number of effects as the acoustic fluid interacts with the solid matrix. For many categories of materials, the solid matrix can be approximated as either fully rigid compared to the acoustic fluid or fully limp. In these cases a mechanical

model that resolves only acoustic waves will suffice. The acoustic behavior of porous materials can be modeled in a variety of ways in Abaqus/Standard.

Craggs Model

The model discussed in *Craggs (1978)* is readily accommodated in Abaqus. Applying this model results in the dynamic equilibrium equation for the fluid expressed in this form:

$$\nabla^2 \tilde{p} + (ka)^2 K_s \Omega \tilde{p} - ika \frac{R\Omega}{\rho_f c_f} = 0,$$

where R is the real-valued resistivity, Ω is the real-valued dimensionless porosity, K_s is the dimensionless "structure factor," and $ka = \frac{\omega}{c_f}$ is the dimensionless wave number. This equation can be rewritten as

$$\nabla^2 \tilde{p} + \frac{(\omega)^2}{\rho_f c_f^2} \left[\rho_f K_s \Omega + \frac{R\Omega}{i\omega} \right] = 0.$$

This model, therefore, can be applied straightforwardly in Abaqus by setting the material density equal to $\rho_f K_s \Omega$, the volumetric drag equal to $R\Omega$, and the bulk modulus equal to $\rho_f c_f^2$. The Craggs model is supported for all acoustic procedures in Abaqus.

Delany-Bazley and Delany-Bazley-Miki Models

Abaqus/Standard supports the well-known empirical model proposed in *Delany & Bazley (1970)*, which determines the material properties as a function of frequency and user-defined flow resistivity, R; density, ρ_f; and bulk modulus, K_f. A variation on this model, proposed by *Miki (1990)* is also available. These models are supported only for steady-state dynamic procedures.

Both models compute frequency-dependent material characteristic impedance, $\tilde{Z}$, and wavenumber or propagation constant, $\tilde{k}$, according to the following formula:

$$\tilde{k} \equiv \frac{\omega}{c_0} \left(1 + C_1 x^{C_2} - iC_3 x^{C_4}\right),$$
$$\tilde{Z} \equiv \rho_f c_0 \left(1 + C_5 x^{C_6} - iC_7 x^{C_8}\right),$$

where

$$c_0 \equiv \sqrt{\frac{K_f}{\rho_f}},$$

and

$$x \equiv \frac{\rho_f \omega}{2\pi R}.$$

The constants are as given in the table below:

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Delany- Bazley	0.0978	–0.7	0.189	–0.595	0.0571	–0.754	0.087	–0.732
Miki	0.1227	–0.618	0.1792	–0.618	0.0786	–0.632	0.1205	–0.632

The material characteristic impedance and the wavenumber are converted internally to complex density and complex bulk modulus for use in Abaqus. The signs of the imaginary parts in these formulas are consistent with the Abaqus sign convention for time-harmonic dynamics.

Input File Usage: Use the following options to use the Delany-Bazley model:
**DENSITY*
**ACOUSTIC MEDIUM*, BULK MODULUS
**ACOUSTIC MEDIUM*, POROUS MODEL=DELANY BAZLEY
Use the following options to use the Miki model:
**DENSITY*
**ACOUSTIC MEDIUM*, BULK MODULUS
**ACOUSTIC MEDIUM*, POROUS MODEL=MIKI

Abaqus/CAE Usage: Porous acoustic material models are not supported in Abaqus/CAE.

General Frequency-Dependent Models

For steady-state dynamic procedures, Abaqus/Standard supports general frequency-dependent complex bulk modulus and complex density. Using these parameters, data from a wide range of models can be accommodated in an analysis; for example, see *Allard, et. al (1998)*, *Attenborough (1982)*, *Song & Bolton (1999)*, and *Wilson (1993)*. These models are used in a variety of applications, such as ocean acoustics, aerospace, automotive, and architectural acoustic engineering.

The signs of these parameters must be consistent with the sign conventions used in Abaqus, and with conservation of energy. Abaqus uses a Fourier transform pair formally equivalent to assuming $e^{i\omega t}$ time dependence. Consequently, the real parts of the density and bulk modulus are positive for all values of frequency, the imaginary part of the bulk modulus must be positive, and the imaginary part of the density must be negative.

A linear isotropic acoustic material can be fully described with the two frequency-dependent parameters: bulk modulus, $\widetilde{K}_f$, and density, $\tilde{\rho}_f$. It is common, however, to encounter materials defined in terms of other parameter pairs, such as characteristic impedance, $\widetilde{Z}$, wave number or propagation constant, $\tilde{k}$, or speed of sound, $\tilde{c}$. Data defined with the pair $(\widetilde{Z}, \tilde{k})$ or $(\widetilde{Z}, \tilde{c})$ can be converted into the complex density and bulk modulus form, beginning from the following standard formulas:

$$\widetilde{Z} \equiv \sqrt{\tilde{\rho}_f \widetilde{K}_f},$$

$$\tilde{k} \equiv \omega \sqrt{\frac{\tilde{\rho}_f}{\widetilde{K}_f}},$$

$$\tilde{c} \equiv \sqrt{\frac{\tilde{\rho}_f}{\widetilde{K}_f}}.$$

Consistent with the Abaqus sign conventions, the real parts of $\tilde{k}$ and $\tilde{c}$ must be positive; the imaginary part of $\tilde{k}$ must be negative, and the imaginary part of $\tilde{c}$ must be positive. In commonly observed materials, the ratio of the magnitude of the imaginary part to the real part for each of these constants is usually much less than one.

Input File Usage: Use the following option to use the general frequency-dependent model:
**ACOUSTIC MEDIUM*, COMPLEX BULK MODULUS
**ACOUSTIC MEDIUM*, COMPLEX DENSITY

If desired, either complex material option can be used instead in conjunction with a real-valued, frequency-independent material option:

**ACOUSTIC MEDIUM*, COMPLEX BULK MODULUS
**DENSITY*

or, alternatively,

**ACOUSTIC MEDIUM*, BULK MODULUS
**ACOUSTIC MEDIUM*, COMPLEX DENSITY

Abaqus/CAE Usage: General frequency-dependent acoustic material models are not supported in Abaqus/CAE.

Conversion from Complex Material Impedance and Wavenumber

Since

$$\tilde{\rho}_f = \tfrac{1}{\omega}\tilde{k}\tilde{Z}$$

and

$$\widetilde{K}_f = \omega\tfrac{\tilde{Z}}{\tilde{k}},$$

the real and imaginary parts of $\tilde{\rho}_f$ are, respectively:

$$\Re(\tilde{\rho}_f) = \tfrac{1}{\omega}\left(\Re(\tilde{k})\,\Re(\tilde{Z}) - \Im(\tilde{k})\,\Im(\tilde{Z})\right),$$
$$\Im(\tilde{\rho}_f) = \tfrac{1}{\omega}\left(\Im(\tilde{k})\,\Re(\tilde{Z}) + \Re(\tilde{k})\,\Im(\tilde{Z})\right);$$

and the real and imaginary parts of $\widetilde{K}_f$ are, respectively:

$$\Re\left(\widetilde{K}_f\right) = \tfrac{\omega}{|\tilde{k}|^2}\left(\Re(\tilde{k})\,\Re(\tilde{Z}) + \Im(\tilde{k})\,\Im(\tilde{Z})\right),$$
$$\Im\left(\widetilde{K}_f\right) = \tfrac{\omega}{|\tilde{k}|^2}\left(\Re(\tilde{k})\,\Im(\tilde{Z}) - \Im(\tilde{k})\,\Re(\tilde{Z})\right).$$

.

Conversion from Complex Impedance and Speed of Sound

Since

$$\tilde{\rho}_f = \tfrac{\tilde{Z}}{\tilde{c}}$$

and

$$\widetilde{K}_f = \tilde{Z}\tilde{c},$$

the real and imaginary parts of $\tilde{\rho}_f$ are, respectively:

$$\Re(\tilde{\rho}_f) = \tfrac{1}{|\tilde{c}|^2}\left(\Re(\tilde{c})\,\Re(\tilde{Z}) + \Im(\tilde{c})\,\Im(\tilde{Z})\right),$$
$$\Im(\tilde{\rho}_f) = \tfrac{1}{|\tilde{c}|^2}\left(\Re(\tilde{c})\,\Im(\tilde{Z}) - \Im(\tilde{c})\,\Re(\tilde{Z})\right);$$

and the real and imaginary parts of $\widetilde{K}_f$ are, respectively:

$$\Re\left(\widetilde{K}_f\right) = \left(\Re(\tilde{c})\,\Re(\tilde{Z}) - \Im(\tilde{c})\,\Im(\tilde{Z})\right),$$

$$\Im\left(\widetilde{K}_f\right) = \left(\Im(\tilde{c})\,\Re(\widetilde{Z}) + \Re(\tilde{c})\,\Im(\widetilde{Z})\right).$$

.

Fluid Cavitation

In general, fluids cannot withstand any significant tensile stress and are likely to undergo large volume expansion when the absolute pressure is close to or less than zero. Abaqus/Explicit allows modeling of this phenomenon through a cavitation pressure limit for the acoustic medium. When the fluid absolute pressure (sum of the dynamic and initial static pressures) reduces to this limit, the fluid undergoes free volume expansion (that is, cavitation), without a further drop in the pressure. If this limit is not defined, the fluid is assumed not to undergo cavitation even under a tensile, negative absolute pressure, condition.

The constitutive behavior for an acoustic medium capable of undergoing cavitation can be stated as

$$p = \max\left\{p_v, p_c - p_0\right\},$$

where a pseudo-pressure p_v, a measure of the volumetric strain, is defined as

$$p_v = -\mathbf{K}_f \varepsilon_V,$$

where p_c is the fluid cavitation limit and p_0 is the initial acoustic static pressure. A total wave formulation is used for a nonlinear acoustic medium undergoing cavitation. This formulation is very similar to the scattered wave formulation except that the pseudo-pressure, defined as the product of the bulk modulus and the compressive volumetric strain, plays the role of the material state variable instead of the acoustic dynamic pressure and the acoustic dynamic pressure is readily available from this pseudo-pressure subject to the cavitation condition.

Input File Usage: **ACOUSTIC MEDIUM*, CAVITATION LIMIT

Abaqus/CAE Usage: Fluid cavitation is not supported in Abaqus/CAE.

Defining the Wave Formulation

In the presence of cavitation in Abaqus/Explicit the fluid mechanical behavior is nonlinear. Hence, for an acoustic problem with incident wave loading and possible cavitation in the fluid, the scattered wave formulation, which provides a solution for only a scattered wave dynamic acoustic pressure, might not be appropriate. For these cases the total wave formulation, which solves for the total dynamic acoustic pressure, should be selected. See *Acoustic and Shock Loads* for details.

Input File Usage: **ACOUSTIC WAVE FORMULATION*, TYPE=TOTAL WAVE

Abaqus/CAE Usage: Any module: **Model->Edit Attributes->*model_name***. Toggle on **Specify acoustic wave formulation**: **Total wave**

Defining the Initial Acoustic Static Pressure

Cavitation occurs when the absolute pressure reaches the cavitation limit value. Abaqus/Explicit allows for an initial linearly varying hydrostatic pressure in the fluid medium (see *Defining Initial Acoustic Static Pressure*). You can specify pressure values at two locations and a node set of the acoustic medium nodes. Abaqus/Explicit interpolates from these data to initialize the static pressure at all the nodes in the specified node set. If the pressure at only one location is specified, the hydrostatic pressure in the fluid is assumed to be uniform. The acoustic

static pressure is used only for determining the cavitation status of the acoustic element nodes and does not apply any static loads to the acoustic or structural mesh at their common wetted interface.

Input File Usage: **INITIAL CONDITIONS*, TYPE=ACOUSTIC STATIC PRESSURE

Abaqus/CAE Usage: Initial acoustic pressures are not supported in Abaqus/CAE.

Defining a Steady Flow Field

Acoustic finite elements can be used to simulate time-harmonic wave propagation and natural frequency analysis in the presence of a steady mean flow of the medium. For example, air might move at a speed large enough to affect the propagation speed of waves in the direction of flow and against it. These effects are modeled in Abaqus/Standard by specifying an acoustic flow velocity during the linear perturbation analysis step definition; you do not need to alter the acoustic material properties. See *Coupled Acoustic-Structural Analysis* for details.

Elements

An acoustic material definition can be used only with the acoustic elements in Abaqus (see *Choosing the Appropriate Element for an Analysis Type*).

In Abaqus/Standard second-order acoustic elements are more accurate than first-order elements. Use at least six nodes per wavelength λ in the acoustic medium to obtain accurate results.

Output

You can use nodal output variable PAC (which is an alias of POR) to output the pressure magnitude for an acoustic medium. When the scattered wave formulation is used with incident wave loading in Abaqus/Explicit, output variable POR represents only the scattered pressure response of the model and does not include the incident wave loading itself. When the total wave formulation is used, output variable POR represents the total dynamic acoustic pressure, which includes contributions from both incident and scattered waves as well as the dynamic effects of fluid cavitation. For either formulation output variable POR does not include the acoustic static pressure, which is used only to evaluate the cavitation status in the acoustic medium.

In addition, in Abaqus/Standard nodal output variable PPOR (the pressure phase) is available for an acoustic medium. In Abaqus/Explicit nodal output variable PABS (the absolute pressure, equal to the sum of POR and the acoustic static pressure) is available for an acoustic medium.

References

- Allard, J. F., M. Henry, J. Tizianel, L. Kelders, and W. Lauriks, "Sound Propagation in Air-Saturated Random Packings of Beads," *Journal of the Acoustical Society of America*, vol. 104, no. 4 2004, 1998.
- Attenborough, K. F., "Acoustical Characterisitics of Rigid Fibrous Absorbents and Granular Materials," *Journal of the Acoustical Society of America*, vol. 73, no. 3 785, 1982.
- Craggs, A., "A Finite Element Model for Rigid Porous Absorbing Materials," *Journal of Sound and Vibration*, vol. 61, no. 1 101, 1978.
- Craggs, A., "Coupling of Finite Element Acoustic Absorption Models," *Journal of Sound and Vibration*, vol. 66, no. 4 605, 1979.
- Delany, M. E., and E. N. Bazley , "Acoustic Properties of Fibrous Absorbent Materials," *Applied Acoustics*, vol. 3 105, 1970.
- Miki, Y., "Acoustical Properties of Porous Materials - Modifications of Delany-Bazley Models," *Journal of the Acoustical Society ofJapan (E)*, vol. 11, no. 1 19, 1990.

- Song, B. H., and J. S. Bolton, "A Transfer-Matrix Approach for Estimating the Characteristic Impedance and Wavenumbers of Limp and Rigid Porous Materials," *Journal of the Acoustical Society of America*, vol. 107, no. 3 1131, 1999.
- Wilson, D. K., "Relaxation-Matched Modeling of Propagation through Porous Media, Including Fractal Pore Structure," *Journal of the Acoustical Society of America*, vol. 94, no. 2 1136, 1993.

Poroelastic Acoustic Medium

Products: Abaqus/Standard

References:

- **ACOUSTIC MEDIUM*
- **ELASTIC*
- **INITIAL CONDITIONS*
- *Poroelastic acoustic medium in frequency domain*

Overview

A poroelastic acoustic medium:

- is used in a coupled acoustic-structural analysis, such as determining noise levels in a vibration problem;
- is a poroelastic medium fully saturated with fluid;
- is specified as part of a material definition;
- must appear in conjunction with a linear elastic isotropic material definition (see *Linear Elastic Behavior*);
- can be defined as a function of frequency;
- can include dissipative effects; and
- is active only during direct steady-state dynamic analysis procedures (see *About Dynamic Analysis Procedures*).

Frequency Response of a Poroelastic Acoustic Medium

The frequency domain theory of the poroelastic acoustic medium is presented in *Poroelastic acoustic medium in frequency domain*. The primary variables in the formulation are structural displacements u_i^s and fluid pore pressure p. The dynamic equilibrium equations are

$$\widehat{\sigma}^s_{ij,j} + \omega^2 \tilde{\rho} u_i^s + \tilde{\gamma} p_{,i} = 0,$$

$$\Delta p + \frac{\tilde{\rho}_{22}}{\tilde{R}} \omega^2 p - \frac{\tilde{\rho}_{22}}{\phi^2} \tilde{\gamma} \omega^2 \Theta^s = 0.$$

Here, ω is the steady-state dynamics excitation frequency, ϕ is the porosity defined as the ratio of the connected volume fully saturated by fluid to the whole volume, $\widehat{\sigma}^s_{ij}$ is the structural stress, Θ^s is the structural dilatation, and parameters with an overtilde ($\tilde{}$) are derived from the material properties of the solid structural skeleton and fluid. The $\tilde{\gamma}$ terms in the equations above represent the fluid-structure volumetric coupling terms. Without these terms the equations are the regular dynamic equilibrium equations for the separate solid and fluid phases. For the solid skeleton, there is no $(1 - \phi)$ factor in the stiffness or density structural terms.

Generally, complex solid and fluid properties and coupling terms are determined from classical isotropic linear elastic solid properties, including the skeleton density plus the fluid real and complex material densities, complex bulk modulus, and the medium tortuosity. In the case of no damping, all properties are real valued with zero imaginary parts. For the solid, the damping can come from structural damping; for the fluid, from nonzero imaginary parts of the complex density and bulk modulus.

There are two ways to obtain the fluid complex density and bulk modulus: either directly from input data as implemented in the Biot-Atalla acoustic medium porous model or using special theory under the Biot-Johnson model. The Biot-Atalla model is a convenient way to compare poroelastic acoustic results versus pure acoustic results.

Poroelastic Acoustic Material Behavior

The response of the solid skeleton is assumed to be isotropic linear elastic with (optionally) structural damping. In addition, the porosity of the medium must be specified as an initial condition.

For the pore fluid, two poroelastic acoustic material models are available: the Biot-Atalla model and the Biot-Johnson model. You must specify the following parameters for both models: fluid density, tortuosity at infinite frequency, skeleton material bulk modulus, and the fluid-structure coupling factor (the default is one). Specifying a coupling factor value of zero leads to an uncoupled fluid and structural response. Additional input is necessary for both models.

Input File Usage: Use the following options for both models:

**ACOUSTIC MEDIUM*
fluid density, *tortuosity*, *bulk modulus*, *fluid-structure coupling factor*
**INITIAL CONDITIONS*, TYPE=POROSITY
**MATERIAL*
**DENSITY*
**ELASTIC*
**DAMPING*, STRUCTURAL

Biot-Atalla Model

The Biot-Atalla model relies on direct specification of the complex bulk modulus and complex density of the pore fluid.

Input File Usage: Use the following options for the Biot-Atalla model:

**ACOUSTIC MEDIUM*, COMPLEX BULK MODULUS
**ACOUSTIC MEDIUM*, COMPLEX DENSITY
**ACOUSTIC MEDIUM*, POROUS MODEL=BIOT-ATALLA

Biot-Johnson Model

To compute the complex bulk modulus and complex density of the pore fluid, the model requires you to define the following material parameters:

- static flow resistivity,
- viscous characteristic length,
- thermal characteristic length,
- fluid dynamic shear viscosity,
- ambient fluid standard pressure,
- ambient fluid heat capacity ratio (the ratio of the specific heat per unit mass at constant pressure over the specific heat per unit mass at constant volume), and
- Prandtl number (the ratio of the viscous diffusion rate over the thermal diffusion rate).

Input File Usage: Use the following option for the Biot-Johnson model:

*ACOUSTIC MEDIUM, POROUS MODEL=BIOT-JOHNSON

Elements

An acoustic material definition can be used only with three-dimensional first-order poroelastic acoustic elements in Abaqus (see *Choosing the Appropriate Element for an Analysis Type*).

Output

Nodal output variable POR (real and imaginary pore pressure components) is available for poroelastic acoustic elements in Abaqus. Similar to regular acoustic finite elements, the output variable POR does not include the acoustic static pressure. In addition, the nodal output variable PPOR (pressure magnitude and phase) is available for a poroelastic acoustic medium.

Element integration point output variable GRADP (pore pressure spatial gradient) is available for the fluid phase of poroelastic acoustic elements. Other element integration point output variables are ACDISP (pore fluid displacement), ACV (pore fluid velocity), and ACADMIT (poroelastic acoustic admittance, a reciprocal of impedance).

The solid phase output is the same as for regular solid elements. It includes nodal displacements, velocities, accelerations, element integration point strains and stresses, etc., all as complex variables.

The stresses in the solid phase include only structural terms, without fluid phase pore pressure contribution. Similarly, output variable POR is the pure fluid pore pressure.

Mass Diffusion Properties

In this section:

- *Diffusivity*
- *Solubility*

Diffusivity

Products: Abaqus/Standard Abaqus/CAE

References:

- *Mass Diffusion Analysis*
- *About the Material Library*
- **DIFFUSIVITY*
- **KAPPA*
- *Defining mass diffusion*

Overview

Diffusivity:

- defines the diffusion or movement of one material through another, such as the diffusion of hydrogen through a metal;
- must always be defined for mass diffusion analysis;
- must be defined in conjunction with *Solubility*;
- can be defined as a function of concentration, temperature, and/or predefined field variables;
- can be used in conjunction with a "Soret effect" factor to introduce mass diffusion caused by temperature gradients;
- can be used in conjunction with a pressure stress factor to introduce mass diffusion caused by gradients of equivalent pressure stress (hydrostatic pressure); and
- can produce a nonlinear mass diffusion analysis when dependence on concentration is included (the same can be said for the Soret effect factor and the pressure stress factor).

Defining Diffusivity

Diffusivity is the relationship between the concentration flux, $\mathbf{J}$, of the diffusing material and the gradient of the chemical potential that is assumed to drive the mass diffusion process. Either general mass diffusion behavior or Fick's diffusion law can be used to define diffusivity, as discussed below.

General Chemical Potential

Diffusive behavior provides the following general chemical potential:

$$\mathbf{J} = -s\mathbf{D} \cdot \left[\frac{\partial \phi}{\partial \mathbf{x}} + \kappa_s \frac{\partial}{\partial \mathbf{x}} \left(\ln\left(\theta - \theta^Z\right) \right) + \kappa_p \frac{\partial p}{\partial \mathbf{x}} \right],$$

where

$\mathbf{D}\left(c, \theta, f_i\right)$

is the diffusivity;

$s(\theta, f_i)$

is the solubility (see *Solubility*);

$\kappa_s(c, \theta, f_i)$

is the Soret effect factor, providing diffusion because of temperature gradient (see below);

$\kappa_p(c, \theta, f_i)$

is the pressure stress factor, providing diffusion because of the gradient of the equivalent pressure stress (see below);

$\phi \stackrel{\text{def}}{=} c/s$

is the normalized concentration;

c

is the concentration of the diffusing material;

θ

is the temperature;

θ^Z

is the temperature at absolute zero (see below);

$p \stackrel{\text{def}}{=} -\text{trace}(\boldsymbol{\sigma})/3$

is the equivalent pressure stress; and

f_i

are any predefined field variables.

Input File Usage: **DIFFUSIVITY*, LAW=GENERAL (default)

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: General**

Fick's Law

An extended form of Fick's law can be used as an alternative to the general chemical potential:

$$\mathbf{J} = -\mathbf{D} \cdot \left(\frac{\partial c}{\partial \mathbf{x}} + s\kappa_p \frac{\partial p}{\partial \mathbf{x}} \right).$$

Input File Usage: **DIFFUSIVITY*, LAW=FICK

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: Fick**

Directional Dependence of Diffusivity

Isotropic, orthotropic, or fully anisotropic diffusivity can be defined. For non-isotropic diffusivity a local orientation of the material directions must be specified (see *Orientations*).

Isotropic Diffusivity

For isotropic diffusivity only one value of diffusivity is needed at each concentration, temperature, and field variable value.

Input File Usage: **DIFFUSIVITY*, TYPE=ISO

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Type: Isotropic**

Orthotropic Diffusivity

For orthotropic diffusivity three values of diffusivity (D_{11}, D_{22}, D_{33}) are needed at each concentration, temperature, and field variable value.

Input File Usage: **DIFFUSIVITY*, TYPE=ORTHO

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Type: Orthotropic**

Anisotropic Diffusivity

For fully anisotropic diffusivity six values of diffusivity (D_{11}, D_{12}, D_{22}, D_{13}, D_{23}, D_{33}) are needed at each concentration, temperature, and field variable value.

Input File Usage: **DIFFUSIVITY*, TYPE=ANISO

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Type: Anisotropic**

Temperature-Driven Mass Diffusion

The Soret effect factor, κ_s, governs temperature-driven mass diffusion. It can be defined as a function of concentration, temperature, and/or field variables in the context of the constitutive equation presented above. The Soret effect factor cannot be specified in conjunction with Fick's law since it is calculated automatically in this case (see *Mass Diffusion Analysis*).

Input File Usage: Use both of the following options to specify general temperature-driven mass diffusion:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=TEMP

Use the following option to specify temperature-driven diffusion governed by Fick's law:

*DIFFUSIVITY, LAW=FICK

Abaqus/CAE Usage: Use the following options to specify general temperature-driven mass diffusion:

Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: General**: **Suboptions->Soret Effect**

Use the following option to specify temperature-driven diffusion governed by Fick's law:

Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: Fick**

Pressure Stress-Driven Mass Diffusion

The pressure stress factor, κ_p, governs mass diffusion driven by the gradient of the equivalent pressure stress. It can be defined as a function of concentration, temperature, and/or field variables in the context of the constitutive equation presented above.

Input File Usage: Use both of the following options:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=PRESS

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: General**: **Suboptions->Pressure Effect**

Mass Diffusion Driven by Both Temperature and Pressure Stress

Specifying both κ_s and κ_p causes gradients of temperature and equivalent pressure stress to drive mass diffusion.

Input File Usage: Use all of the following options to specify general diffusion driven by gradients of temperature and pressure stress:

*DIFFUSIVITY, LAW=GENERAL
*KAPPA, TYPE=TEMP
*KAPPA, TYPE=PRESS

Use both of the following options to specify diffusion driven by the extended form of Fick's law:

*DIFFUSIVITY, LAW=FICK
*KAPPA, TYPE=PRESS

Abaqus/CAE Usage: Use the following options to specify general diffusion driven by gradients of temperature and pressure stress:

Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: General**: **Suboptions->Soret Effect** and **Suboptions->Pressure Effect**

Use the following options to specify diffusion driven by the extended form of Fick's law:

Property module: material editor: **Other->Mass Diffusion->Diffusivity**: **Law: Fick**: **Suboptions->Pressure Effect**

Specifying the Value of Absolute Zero

You can specify the value of absolute zero as a physical constant.

Input File Usage: **PHYSICAL CONSTANTS*, ABSOLUTE ZERO=θ^Z

Abaqus/CAE Usage: Any module: **Model->Edit Attributes->*model_name***: **Absolute zero temperature**

Elements

The mass diffusion law can be used only with the two-dimensional, three-dimensional, and axisymmetric solid elements that are included in the heat transfer/mass diffusion element library.

Solubility

Products: Abaqus/Standard Abaqus/CAE

References:

- *Mass Diffusion Analysis*
- *About the Material Library*
- **SOLUBILITY*
- *Defining solubility*

Overview

Solubility:

- is needed only for mass diffusion analysis;
- is also known as Sievert's parameter (in Sievert's law);
- must always accompany a diffusivity definition (see *Diffusivity*); and
- can be defined as a function of temperature and/or predefined field variables.

Defining Solubility

Solubility, s, is used to define the "normalized concentration," ϕ, of the diffusing phase in a mass diffusion process:

$$\phi = c/s,$$

where c is the concentration. The normalized concentration is often also referred to as the "activity" of the diffusing material, and the gradients of the normalized concentration, along with gradients of temperature and pressure stress, drive the diffusion process (see *Diffusivity*).

Input File Usage: **SOLUBILITY*

Abaqus/CAE Usage: Property module: material editor: **Other->Mass Diffusion->Solubility**

Elements

The mass diffusion law can be used only with the two-dimensional, three-dimensional, and axisymmetric solid elements that are included in the heat transfer/mass diffusion element library.

Electromagnetic Properties

In this section:

- *Electrical Conductivity*
- *Electrical Resistivity*
- *Piezoelectric Behavior*
- *Piezoresistive Effect*
- *Magnetic Permeability*

Electrical Conductivity

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- **ELECTRICAL CONDUCTIVITY*
- *Defining electrical conductivity*

Overview

A material's electrical conductivity:

- must be defined for *Coupled Thermal-Electrical Analysis* if electrical resistivity is not defined;
- must be defined for *Fully Coupled Thermal-Electrical-Structural Analysis* if electrical resistivity is not defined;
- must be used to define the electromagnetic response of a conductor for *Eddy Current Analysis*;
- can be linear or nonlinear (by defining it as a function of temperature);
- can be isotropic, orthotropic, or fully anisotropic;
- can be specified as a function of temperature and/or field variables;
- can be specified as a function of frequency for *Eddy Current Analysis*;
- cannot be used as part of a material definition that also includes electrical resistivity; and
- cannot be used as part of a material definition that also includes piezoresistivity.

Directional Dependence of Electrical Conductivity

Isotropic, orthotropic, or fully anisotropic electrical conductivity can be defined. For non-isotropic conductivity a local orientation for the material directions must be specified (*Orientations*).

Isotropic Electrical Conductivity

For isotropic electrical conductivity only one value of electrical conductivity is needed at each temperature and field variable value. Isotropic electrical conductivity is the default.

Input File Usage: **ELECTRICAL CONDUCTIVITY*, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Electrical Conductivity**: **Type: Isotropic**

Orthotropic Electrical Conductivity

For orthotropic electrical conductivity three values of electrical conductivity (σ^E_{11}, σ^E_{22}, σ^E_{33}) are needed at each temperature and field variable value.

Input File Usage: **ELECTRICAL CONDUCTIVITY*, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Electrical Conductivity**: **Type: Orthotropic**

Anisotropic Electrical Conductivity

For fully anisotropic electrical conductivity six values (σ_{11}^E, σ_{12}^E, σ_{22}^E, σ_{13}^E, σ_{23}^E, σ_{33}^E) are needed at each temperature and field variable value.

Input File Usage: **ELECTRICAL CONDUCTIVITY*, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Electrical Conductivity**: **Type: Anisotropic**

Frequency-Dependent Electrical Conductivity

Electrical conductivity can be defined as a function of frequency in an eddy current analysis.

Input File Usage: **ELECTRICAL CONDUCTIVITY*, FREQUENCY

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Electrical Conductivity**: Toggle on **Use frequency-dependent data**

Elements

Electrical conductivity is active only in coupled thermal-electrical elements, coupled thermal-electrical-structural elements, electromagnetic elements, coupled thermal-electrochemical elements, coupled thermal-electrochemical-structural elements, and coupled thermal-electrochemical-structural-pore pressure elements (see *Choosing the Appropriate Element for an Analysis Type*).

Electrical Resistivity

Products: Abaqus/Standard

References:

- *About the Material Library*
- **ELECTRICAL RESISTIVITY*
- *Defining electrical conductivity*
- *Piezoresistive Effect*

Overview

A material's electrical resistivity:

- must be defined for *Coupled Thermal-Electrical Analysis* if electrical conductivity is not defined;
- must be defined for *Fully Coupled Thermal-Electrical-Structural Analysis* if electrical conductivity is not defined;
- must be defined to model materials with piezoresistive effects;
- can be linear or nonlinear (by defining electrical resistivity as a function of temperature);
- can be isotropic, orthotropic, or fully anisotropic;
- can be specified as a function of temperature and/or field variables;
- cannot be used as part of a material definition that also includes electrical conductivity; and
- cannot be used in an eddy current analysis.

Directional Dependence of Electrical Resistivity

You can define isotropic, orthotropic, or fully anisotropic electrical resistivity. For non-isotropic resistivity, you must specify a local orientation for the material directions (*Orientations*).

Isotropic Electrical Resistivity

For isotropic electrical resistivity, only one value of electrical resistivity is needed at each temperature and field variable value. Isotropic electrical resistivity is the default.

Input File Usage: **ELECTRICAL RESISTIVITY*, TYPE=ISOTROPIC

Orthotropic Electrical Resistivity

For orthotropic electrical resistivity, three values of electrical resistivity (ρ^E_{11}, ρ^E_{22}, ρ^E_{33}) are needed at each temperature and field variable value.

Input File Usage: **ELECTRICAL RESISTIVITY*, TYPE=ORTHOTROPIC

Anisotropic Electrical Resistivity

For fully anisotropic electrical resistivity, six values (ρ^E_{11}, ρ^E_{12}, ρ^E_{22}, ρ^E_{13}, ρ^E_{23}, ρ^E_{33}) are needed at each temperature and field variable value.

Input File Usage: **ELECTRICAL RESISTIVITY*, TYPE=ANISOTROPIC

Elements

Electrical resistivity is active only in coupled thermal-electrical elements, coupled thermal-electrical-structural elements, coupled thermal-electrochemical elements, coupled thermal-electrochemical-structural elements, and coupled thermal-electrochemical-structural-pore pressure elements (see *Choosing the Appropriate Element for an Analysis Type*).

Piezoelectric Behavior

Products: Abaqus/Standard

References:

- *Piezoelectric Analysis*
- *About the Material Library*
- **DIELECTRIC*
- **PIEZOELECTRIC*
- *Defining dielectric material properties*
- *Defining piezoelectric properties*

Overview

A piezoelectric material:

- is one in which an electrical field causes the material to strain, while stress causes an electric potential gradient;
- provides linear relations between mechanical and electrical fields; and
- is used in piezoelectric elements, which have both displacement and electrical potential as nodal variables.

Defining a Piezoelectric Material

A piezoelectric material responds to an electric potential gradient by straining, while stress causes an electric potential gradient in the material. This coupling between electric potential gradient and strain is the material's piezoelectric property. The material will also have a dielectric property so that an electrical charge exists when the material has a potential gradient. Piezoelectric material behavior is discussed in *Piezoelectric analysis*.

The mechanical properties of the material must be modeled by linear elasticity (*Linear Elastic Behavior*). The mechanical behavior can be defined by

$$\sigma_{ij} = D^E_{ijkl}\varepsilon_{kl} - e^{\varphi}_{mij}E_m,$$

in terms of the piezoelectric stress coefficient matrix, e^{φ}_{mij}, or by

$$\sigma_{ij} = D^E_{ijkl}\left(\varepsilon_{kl} - d^{\varphi}_{mkl}E_m\right),$$

in terms of the piezoelectric strain coefficient matrix, d^{φ}_{mkl}. The electrical behavior is defined by

$$q_i = e^{\varphi}_{ijk}\varepsilon_{jk} + D^{\varphi(\varepsilon)}_{ij}E_j,$$

where

σ_{ij}

is the mechanical stress tensor;

ε_{ij}

is the strain tensor;

q_i

is the electric “displacement” vector;

D^E_{ijkl}

is the material's elastic stiffness matrix defined at zero electrical potential gradient (short circuit condition);

e^{φ}_{mij}

is the material's piezoelectric stress coefficient matrix, defining the stress σ_{ij} caused by the electrical potential gradient E_m in a fully constrained material (it can also be interpreted as the electrical displacement q_m caused by the applied strain ε_{ij} at a zero electrical potential gradient);

d^{φ}_{mkl}

is the material's piezoelectric strain coefficient matrix, defining the strain ε_{kl} caused by the electrical potential gradient E_m in an unconstrained material (an alternative interpretation is given later in this section);

φ

is the electrical potential;

$D^{\varphi(\varepsilon)}_{ij}$

is the material's dielectric property, defining the relation between the electric displacement q_i and the electric potential gradient E_j for a fully constrained material; and

E_i

is the electrical potential gradient vector, $-\partial\varphi/\partial x_i$.

The material's electrical and electro-mechanical coupling behaviors are, thus, defined by its dielectric property, $D^{\varphi(\varepsilon)}_{ij}$, and its piezoelectric stress coefficient matrix, e^{φ}_{mij}, or its piezoelectric strain coefficient matrix, d^{φ}_{mkl}. These properties are defined as part of the material definition (*Material Data Definition*).

Alternative Forms of the Constitutive Equations

Alternative forms of the piezoelectric constitutive equations are presented in this section. These forms of the equations involve material properties that cannot be used directly as input for Abaqus/Standard. However, they are related to the Abaqus/Standard input through simple relations that are presented in *Piezoelectric analysis*.

The intent of this section is to draw connections between the Abaqus/Standard terminology and input to that used commonly in the piezoelectricity community. The mechanical behavior can also be defined by

$$\sigma_{ij} = D^{q}_{ijkl}\left(\varepsilon_{kl} - g^{\varphi}_{mkl}q_m\right)$$

in terms of the piezoelectric coefficient matrix g^{φ}_{mij}, and the stiffness matrix D^{q}_{ijkl}, which defines the mechanical properties at zero electrical displacement (open circuit condition). Likewise, the electrical behavior can also be defined by

$$q_i = d^{\varphi}_{ijk}\sigma_{jk} + D^{\varphi(\sigma)}_{ij}E_j$$

in terms of the dielectric matrix $D^{\varphi(\sigma)}_{ij}$ for an unconstrained material or by

$$q_i = D^{\varphi(\sigma)}_{im}g^{\varphi}_{mjk}\sigma_{jk} + D^{\varphi(\sigma)}_{ij}E_j,$$

where

D^{q}_{ijkl}

is the material's elastic stiffness matrix defined at zero electrical displacement;

d^{φ}_{ijk}

is the material's piezoelectric strain coefficient matrix used earlier, and based on the equations, may alternatively be interpreted as the electrical displacement q_i caused by the stress σ_{jk} at zero electrical potential gradient;

g^{φ}_{mkl}

is the material's piezoelectric coefficient matrix, which can be interpreted as defining either the strain ε_{kl} caused by the electrical displacement q_m in an unconstrained material or the electrical potential gradient E_m caused by the stress σ_{kl} at zero electrical displacement; and

$D^{\varphi(\sigma)}_{ij}$

is the material's dielectric property, defining the relation between the electric displacement q_i and the electric potential gradient E_j for an unconstrained material.

These are useful relationships that are often seen in the piezoelectric literature. In *Piezoelectric analysis* the properties g^{φ}_{mkl}, $D^{\varphi(\sigma)}_{ij}$, and D^{q}_{ijkl} are expressed in terms of the properties d^{φ}_{mkl}, $D^{\varphi(\varepsilon)}_{ij}$, and D^{E}_{ijkl}, that are used as input for Abaqus/Standard.

Specifying Dielectric Material Properties

The dielectric matrix can be isotropic, orthotropic, or fully anisotropic. For non-isotropic dielectric materials a local orientation for the material directions must be specified (*Orientations*). The entries of the dielectric matrix (what are referred to as “dielectric constants” in Abaqus) refer to what is more commonly known in the literature as the permittivity of the material.

Isotropic Dielectric Properties

The dielectric matrix $D_{ij}^{\varphi(\varepsilon)}$ can be fully isotropic, so that

$$D_{ij}^{\varphi(\varepsilon)} = D^{\varphi(\varepsilon)}\delta_{ij}.$$

You specify the single value $D^{\varphi(\varepsilon)}$ for the dielectric constant. $D^{\varphi(\varepsilon)}$ must be determined for a constrained material. Isotropic behavior is the default.

Input File Usage: **DIELECTRIC*, TYPE=ISO

Orthotropic Dielectric Properties

For orthotropic behavior you must specify three values in the dielectric matrix ($D_{11}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, and $D_{33}^{\varphi(\varepsilon)}$).

Input File Usage: **DIELECTRIC*, TYPE=ORTHO

Anisotropic Dielectric Properties

For fully anisotropic behavior you must specify six values in the dielectric matrix ($D_{11}^{\varphi(\varepsilon)}$, $D_{12}^{\varphi(\varepsilon)}$, $D_{22}^{\varphi(\varepsilon)}$, $D_{13}^{\varphi(\varepsilon)}$, $D_{23}^{\varphi(\varepsilon)}$, and $D_{33}^{\varphi(\varepsilon)}$).

Input File Usage: **DIELECTRIC*, TYPE=ANISO

Specifying Piezoelectric Material Properties

The piezoelectric material properties can be defined by giving the stress coefficients, e_{mij}^{φ} (this is the default), or by giving the strain coefficients, d_{mkl}^{φ}. In either case, 18 components must be given in the following order (substitute *d* for *e* for strain coefficients):

$$\begin{aligned}
&e_{1\,11}^{\varphi},\ e_{1\,22}^{\varphi},\ e_{1\,33}^{\varphi},\ e_{1\,12}^{\varphi},\ e_{1\,13}^{\varphi},\ e_{1\,23}^{\varphi},\\
&e_{2\,11}^{\varphi},\ e_{2\,22}^{\varphi},\ e_{2\,33}^{\varphi},\ e_{2\,12}^{\varphi},\ e_{2\,13}^{\varphi},\ e_{2\,23}^{\varphi},\\
&e_{3\,11}^{\varphi},\ e_{3\,22}^{\varphi},\ e_{3\,33}^{\varphi},\ e_{3\,12}^{\varphi},\ e_{3\,13}^{\varphi},\ e_{3\,23}^{\varphi}
\end{aligned}$$

The first index on these coefficients refers to the component of electric displacement (sometimes called the electric flux), while the last pair of indices refers to the component of mechanical stress or strain.

Thus, the piezoelectric components causing electrical displacement in the 1-direction are all given first, then those causing electrical displacement in the 2-direction, and then those causing electrical displacement in the 3-direction. (Some references list these coupling terms in a different order.)

Input File Usage: Use the following option to give the stress coefficients:

**PIEZOELECTRIC*, TYPE=S

Use the following option to give the strain coefficients:

**PIEZOELECTRIC*, TYPE=E

Converting Double Index Notation to Triple Index Notation

Industry-supplied piezoelectric data often use a double index notation. A double index notation can be converted easily to the required triple index notation in Abaqus/Standard by noting the convention followed in Abaqus for the correspondence between (second-order) tensor and vector notations: the 11, 22, 33, 12, 13, and 23 components of the tensor correspond to the 1, 2, 3, 4, 5, and 6 components, respectively, of the corresponding vector.

Energy Balance Considerations

Abaqus does not account for piezoelectric effects in the total energy balance equation, which can lead to an apparent imbalance of the total energy of the model in some situations. For example, if a piezoelectric truss is fixed at one end point and subjected to a potential difference between its two end points, it deforms due to the piezoelectric effect. Subsequently if the truss is held fixed in this deformed configuration and the potential difference removed, strain energy will be generated due to the constraints. This results in an equivalent increase in the total energy of the model.

Elements

Piezoelectric coupling is active only in piezoelectric elements (those with displacement degrees of freedom and electrical potential degree of freedom 9). See *Choosing the Appropriate Element for an Analysis Type*.

Piezoresistive Effect

Products: Abaqus/Standard

References:

- *About the Material Library*
- **PIEZORESISTIVITY*
- *Electrical Resistivity*
- **ELECTRICAL RESISTIVITY*

Overview

A piezoresistive material:

- is one in which stress causes the electrical resistivity of the material to change;
- provides linear relations between stress and relative resistivity changes;
- must be used with electrical resistivity;
- cannot be used as part of a material definition that also includes electrical conductivity; and
- is used in coupled thermal-electrical-structural elements, which have displacement, temperatures, and electrical potential as nodal variables.

Defining a Piezoresistive Material

A piezoresistive material undergoes a change in its electrical resistivity when it is stressed. This coupling between stress and electrical resistivity is the material's piezoresistivity property. The electrical response of the material is also defined in terms of electrical resistivity properties when the material is unstressed.

The piezoresistive effect can be defined by

$$\Delta_{ij} = \pi_{ijkl}\sigma_{kl},$$

where

σ_{ij}

is the mechanical stress tensor;

π_{ijkl}

the material’s piezoresistivity coefficient tensor; and

Δ_{ij}

is a tensor that represents the relative change in the electrical resistivity with stress.

The material’s electrical resistivity when stressed is defined by

$$\rho_{ij}^{E} = \mathbf{sym}\left[\rho_{ik}^{0}\left(\delta_{kj} + \Delta_{kj}\right)\right],$$

where

ρ_{ij}^0

is the material's electrical resistivity tensor when it is in an unstressed state; and

δ_{ij}

is the Kronecker delta.

Specifying Piezoresistive Material Properties

The piezoresistive material properties can be defined by providing the piezoresistivity coefficients. The piezoresistivity coefficient tensor can be isotropic, cubic, orthotropic, or fully anisotropic. For nonisotropic piezoresistive materials, you must specify a local orientation for the material directions (*Orientations*). A piezoresistive material response also requires the electrical resistivity of the unstressed material. You must specify the electrical resistivity at zero stress directly; you cannot define the electrical properties in terms of the electrical conductivity tensor.

Isotropic Piezoresistivity

For isotropic piezoresistivity, two values (π_{1111}, π_{1122}) of the piezoresistivity coefficient are required at each temperature and field variable value. Isotropic behavior is the default.

Input File Usage: **PIEZORESISTIVITY*, TYPE=ISOTROPIC

Cubic Piezoresistivity

For cubic piezoresistivity, three values (π_{1111}, π_{1122}, π_{1212}) of the piezoresistivity coefficient are required at each temperature and field variable value.

Input File Usage: **PIEZORESISTIVITY*,TYPE=CUBIC

Orthotropic Piezoresistivity

For orthotropic piezoresistivity, nine values (π_{1111}, π_{1122}, π_{2222}, π_{1133}, π_{2233}, π_{3333}, π_{1212}, π_{1313}, π_{2323}) of the piezoresistivity coefficient are required at each temperature and field variable value.

Input File Usage: **PIEZORESISTIVITY*, TYPE=ORTHOTROPIC

Anisotropic Piezoresistivity

For fully anisotropic piezoresistivity, 21 values (π_{1111}, π_{1122}, π_{2222}, π_{1133}, π_{2233}, π_{3333}, π_{1112}, π_{2212}, π_{3312}, π_{1212}, π_{1113}, π_{2213}, π_{3313}, π_{1213}, π_{1313}, π_{1123}, π_{2223}, π_{3323}, π_{1223}, π_{1323}, π_{2323}) of the piezoresistivity coefficient are required at each temperature and field variable value.

Input File Usage: **PIEZORESISTIVITY*, TYPE=ANISOTROPIC

Elements

Piezoresistivity is active only in coupled thermal-electrical-structural elements. See *Choosing the Appropriate Element for an Analysis Type*.

Magnetic Permeability

Products: Abaqus/Standard Abaqus/CAE

References:

- *About the Material Library*
- **MAGNETIC PERMEABILITY*
- **NONLINEAR BH*
- **PERMANENT MAGNETIZATION*
- *Defining magnetic permeability*

Overview

A material's magnetic permeability:

- must be defined for *Eddy Current Analysis* and *Magnetostatic Analysis*;
- can be specified directly for linear magnetic behavior or through one or more B–H curves for nonlinear magnetic behavior;
- can be isotropic, orthotropic, or (in the case of linear behavior) fully anisotropic;
- can be specified as a function of temperature and/or field variables;
- can be specified as a function of frequency in a time-harmonic eddy current procedure; and
- can be combined with permanent magnetization.

Linear Magnetic Behavior

Linear magnetic behavior is defined by direct specification of magnetic permeability.

Directional Dependence of Magnetic Permeability

Isotropic, orthotropic, or fully anisotropic magnetic permeability can be defined. For non-isotropic magnetic permeability a local orientation for the material directions must be specified (*Orientations*).

Isotropic Magnetic Permeability

For isotropic magnetic permeability only one value of magnetic permeability is needed at each temperature and field variable value. Isotropic magnetic permeability is the default.

Input File Usage: **MAGNETIC PERMEABILITY*, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: **Type: Isotropic**

Orthotropic Magnetic Permeability

For orthotropic magnetic permeability three values of magnetic permeability (μ_{11}, μ_{22}, μ_{33}) are needed at each temperature and field variable value.

Input File Usage: **MAGNETIC PERMEABILITY*, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: **Type: Orthotropic**

Anisotropic Magnetic Permeability

For fully anisotropic magnetic permeability six values (μ_{11}, μ_{12}, μ_{22}, μ_{13}, μ_{23}, μ_{33}) are needed at each temperature and field variable value.

Input File Usage: **MAGNETIC PERMEABILITY*, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: **Type: Anisotropic**

Frequency-Dependent Magnetic Permeability

Magnetic permeability can be defined as a function of frequency in a time-harmonic eddy current analysis.

Input File Usage: **MAGNETIC PERMEABILITY*, FREQUENCY

Abaqus/CAE Usage: Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: Toggle on **Use frequency-dependent data**

Nonlinear Magnetic Behavior

Nonlinear magnetic behavior is characterized by magnetic permeability that depends on the strength of the magnetic field. The nonlinear magnetic material model in Abaqus is suitable for ideally soft magnetic materials without any hysteresis effects (see *Figure 1*) characterized by a monotonically increasing response in B–H space, where B and H refer to the strengths of the magnetic flux density vector and the magnetic field vector, respectively. Nonlinear magnetic behavior is defined through direct specification of one or more B–H curves that provide B as a function of H and, optionally, temperature and/or predefined field variables, in one or more directions. Nonlinear magnetic behavior can be isotropic, orthotropic, or transversely isotropic (which is a special case of the more general orthotropic behavior). More than one B–H curve is needed to define the nonlinear magnetic behavior if it is not isotropic. For each curve, the slope between the last pair of data points is assumed to be the permeability of the free space. Abaqus issues an error message if the slope between any two adjacent data points is less than 0.01 times the permeability of the free space, as computed based on the last two data points for that curve.

Directional Dependence of Nonlinear Magnetic Behavior

Isotropic, orthotropic, or transversely isotropic nonlinear magnetic behavior can be defined. For non-isotropic nonlinear magnetic behavior a local orientation for the material directions must be specified (*Orientations*).

Isotropic Nonlinear Magnetic Behavior

For isotropic nonlinear magnetic response only one B–H curve is needed at each temperature and field variable value. Isotropic magnetic permeability is the default. Abaqus assumes that the nonlinear magnetic behavior is governed by

$$\mathbf{B} = B\left(|\mathbf{H}|\right)\left(\frac{\mathbf{H}}{|\mathbf{H}|}\right)$$

Input File Usage:

You define $B\left(|\mathbf{H}|\right)$ through a B–H curve:

**MAGNETIC PERMEABILITY*, NONLINEAR, TYPE=ISOTROPIC
**NONLINEAR BH*, DIR=*direction*

The B–H curve in any direction (i.e., the nonlinear behavior in global direction 1, 2, or 3) will suffice as the nonlinear magnetic behavior is assumed to be the same in all directions.

Abaqus/CAE Usage:

Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: Toggle on **Specify using nonlinear B-H curve**: **Type: Isotropic**

Orthotropic Nonlinear Magnetic Behavior

For orthotropic nonlinear magnetic response three B–H curves (one curve to define the behavior in each of the local directions 1, 2, and 3) are needed at each temperature and field variable value. Abaqus assumes that the nonlinear magnetic behavior in the local material directions is governed by

$$\mathbf{B} = diag\left(B_1\left(|\mathbf{H}|\right), B_2\left(|\mathbf{H}|, B_3\left(|\mathbf{H}|\right)\right)\right)\left(\frac{\mathbf{H}}{|\mathbf{H}|}\right),$$

where $diag\left(\right)$ refers to a diagonal matrix.

Transversely isotropic nonlinear magnetic behavior is a special case of orthotropic behavior, in which the behavior in any two directions is the same and is different from that in the third direction.

Input File Usage:

You define $B_1\left(|\mathbf{H}|\right)$, $B_2\left(|\mathbf{H}|\right)$, and $B_3\left(|\mathbf{H}|\right)$, respectively, through three independent B–H curves, one in each of the directions 1, 2, and 3:

**MAGNETIC PERMEABILITY*, NONLINEAR, TYPE=ORTHOTROPIC
**NONLINEAR BH*, DIR=1
…
**NONLINEAR BH*, DIR=2
…
**NONLINEAR BH*, DIR=3
…

Abaqus/CAE Usage:

Property module: material editor: **Electrical/Magnetic->Magnetic Permeability**: Toggle on **Specify using nonlinear B-H curve**: **Type: Orthotropic**

Permanent Magnetization

Ferromagnetic materials can be magnetized by placing them in a magnetic field, which is typically created by applying currents in a system of coil windings surrounding the material being magnetized. These materials can be classified into soft and hard magnetic materials (see *Figure 1*). Soft magnetic materials lose their magnetization after removal of the applied currents (see *Nonlinear Magnetic Behavior*). Hard magnetic materials retain their magnetization permanently after removal of the applied currents. The leftover magnetization in a permanent magnet is called remanence, denoted by B_r in *Figure 2*. This magnetization can be removed by applying currents

in the opposite direction; the strength of the opposing magnetic fields that remove magnetization entirely is called coercivity, denoted by $\boldsymbol{H_c}$ in *Figure 2*.

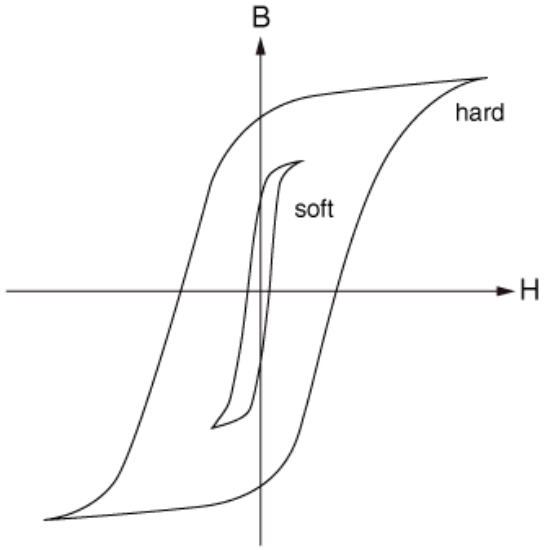


Figure 1: Response of hard and soft magnetic materials.

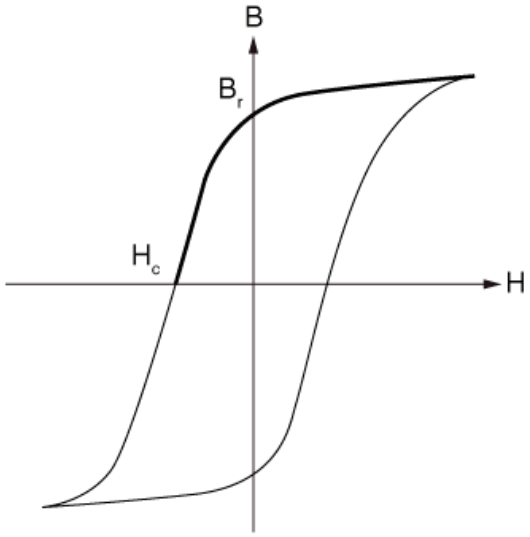


Figure 2: Remanence and coercivity in permanent magnets.

Permanent magnetization in Abaqus is suitable for hard magnetic materials when the magnets are operating around the point of remanence. This behavior captures the response of magnetization or demagnetization around the point of remanence, as shown by the darker descending line of the hysteresis loop in *Figure 2*. The underlying magnetic permeability can be linear or nonlinear. In either case, permanent magnetization is defined by its coercivity such that

$$\mathbf{H} = \boldsymbol{\mu}^{-1}\mathbf{B} - \mathbf{H}_c$$

for linear isotropic, orthotropic, or anisotropic magnetic behavior and

$$\mathbf{H} = \widehat{H}\ (|\mathbf{B}|)\ \left(\frac{\mathbf{B}}{|\mathbf{B}|}\right) - \mathbf{H}_c$$

for nonlinear isotropic $\mathbf{B}$-$\widehat{\mathbf{H}}$ response.

Input File Usage: To specify permanent magnetization with underlying linear magnetic permeability:

**MAGNETIC PERMEABILITY*
**PERMANENT MAGNETIZATION*
direction of magnetization in the global system
magnitude of coercivity

To specify permanent magnetization with underlying nonlinear magnetic permeability (nonlinear response of the left top portion of the hysteresis curve):

**MAGNETIC PERMEABILITY*, NONLINEAR
**NONLINEAR BH*
input $\mathbf{B}$-$\widehat{\mathbf{H}}$ *by shifting the response to the right by* $\boldsymbol{H_c}$
**PERMANENT MAGNETIZATION*
direction of magnetization in the global system
magnitude of coercivity

Abaqus/CAE Usage: Permanent magnetization is not supported in Abaqus/CAE.

Elements

Magnetic material behavior is active only in electromagnetic elements (see *Choosing the Appropriate Element for an Analysis Type*).

Pore Fluid Flow Properties

In this section:

- *About Pore Fluid Flow Properties*
- *Permeability*
- *Porous Bulk Moduli*
- *Sorption*
- *Swelling Gel*
- *Moisture Swelling*

About Pore Fluid Flow Properties

Overview

Abaqus allows specific properties to be defined for a fluid-filled porous material.
In Abaqus/Standard, this type of porous medium is considered in a coupled pore fluid diffusion/stress analysis (*Coupled Pore Fluid Diffusion and Stress Analysis*). In Abaqus/Explicit, this type of porous medium is considered in an undrained pore fluid flow and stress analysis (*Undrained Pore Fluid Flow and Stress Analysis*).

The following properties are available:

Permeability

Permeability defines the relationship between the flow rate of a liquid through a porous medium and the gradient of the piezometric head of that fluid (see *Permeability*) in Abaqus/Standard analyses.

You cannot specify permeability in an undrained pore fluid flow and stress analysis (where the permeability is assumed to be zero) in Abaqus/Explicit.

Porous bulk moduli

The bulk moduli of the solid grains and of the fluid in a porous medium are defined such that their compressibility is considered in an analysis (see *Porous Bulk Moduli*).

Sorption

Sorption defines the absorption/exsorption behavior of a porous material under partially saturated flow conditions (see *Sorption*) in Abaqus/Standard analyses.

You cannot specify sorption in an undrained pore fluid flow and stress analysis (where the flow is assumed to be fully saturated) in Abaqus/Explicit.

Swelling gel

The swelling gel model is used to simulate the growth of gel particles that swell and trap wetting liquid in a partially saturated porous medium (see *Swelling Gel*).

Swelling gel applies only to Abaqus/Standard analyses.

Moisture swelling

Moisture swelling defines the saturation-driven volumetric swelling of a porous medium's solid skeleton under partially saturated flow conditions (see *Moisture Swelling*).

Moisture swelling applies only to Abaqus/Standard analyses.

References:

Thermal Expansion

For porous media such as soils or rock, you can define the thermal expansion of both the solid grains and the permeating fluid. For more information, see *Thermal Expansion*.

Permeability

Products: Abaqus/Standard Abaqus/CAE

References:

- *About Pore Fluid Flow Properties*
- *About the Material Library*
- **PERMEABILITY*
- *Defining permeability*

Overview

Permeability is the relationship between the volumetric flow rate per unit area of a particular wetting liquid through a porous medium and the gradient of the effective fluid pressure. It can be specified in Abaqus/Standard.

Permeability in Abaqus/Standard:

- must be specified for a wetting liquid for an effective stress/wetting liquid diffusion analysis (see *Coupled Pore Fluid Diffusion and Stress Analysis*);
- is defined, in general, by Forchheimer's law, which accounts for changes in permeability as a function of fluid flow velocity; and
- can be isotropic, orthotropic, or fully anisotropic and can be given as a function of void ratio, saturation, temperature, and field variables.

Permeability in Abaqus/Standard

Permeability is defined for pore fluid flow.

Forchheimer's Law

According to Forchheimer's law, high flow velocities have the effect of reducing the effective permeability and, therefore, “choking” pore fluid flow. As the fluid flow velocity reduces, Forchheimer's law approximates the well-known Darcy's law. Darcy's law can, therefore, be used directly in Abaqus/Standard by omitting the velocity-dependent term in Forchheimer's law.

Forchheimer's law is written as

$$\mathbf{f}\left(1+\beta\sqrt{\mathbf{v}_w\cdot\mathbf{v}_w}\right)=-\frac{k_s}{\gamma_w}\mathbf{k}\cdot\left(\frac{\partial u_w}{\partial \mathbf{x}}-\rho_w\mathbf{g}\right),$$

where

$\mathbf{f}=sn\mathbf{v}_w$

is the volumetric flow rate of wetting liquid per unit area of the porous medium (the effective velocity of the wetting liquid);

$s = \frac{dV_w}{dV_v}$

is the fluid saturation ($s = 1$ for a fully saturated medium, $s = 0$ for a completely dry medium);

$n = \frac{dV_v}{dV}$

is the porosity of the porous medium;

$e = \frac{dV_v}{(dV_g + dV_t)}$

is the void ratio;

dV_w

is the wetting fluid volume in the medium;

dV_v

is the void volume in the medium;

dV_g

is the volume of grains of solid material in the medium;

dV_t

is the volume of trapped wetting liquid in the medium;

dV

is the total volume of the medium;

$\mathbf{v}_w$

is the fluid velocity;

$\beta(e)$

is a "velocity coefficient," which may be dependent on the void ratio of the material;

$k_s(s)$

is the dependence of permeability on saturation of the wetting liquid such that $k_s = 1.0$ at $s = 1.0$;

$\rho_w = \gamma_w / g$

is the density of the fluid;

γ_w

is the specific weight of the wetting liquid;

g

is the magnitude of the gravitational acceleration;

$\mathbf{k}(e, \theta, f_\beta)$

is the permeability of the fully saturated medium, which can be a function of void ratio (e, common in soil consolidation problems), temperature (θ), and/or field variables (f_β);

u_w

is the wetting liquid pore pressure;

$\mathbf{x}$

is position; and

$\mathbf{g}$

is the gravitational acceleration.

Permeability Definitions

Permeability can be defined in different ways by different authors; caution should, therefore, be used to ensure that the specified input data are consistent with the definitions used in Abaqus/Standard.

Permeability in Abaqus/Standard is defined as

$$\overline{\mathbf{k}} = \frac{k_s}{\left(1+\beta\sqrt{\mathbf{v}_w \cdot \mathbf{v}_w}\right)}\mathbf{k},$$

so that Forchheimer's law can also be written as

$$\mathbf{f} = -\frac{\overline{\mathbf{k}}}{\gamma_w} \cdot \left(\frac{\partial u_w}{\partial \mathbf{x}} - \rho_w \mathbf{g}\right).$$

The fully saturated permeability, $\mathbf{k}$, is typically obtained from experiments under low fluid velocity conditions. $\mathbf{k}$ can be defined as a function of void ratio, e, (common in soil consolidation problems) and/or temperature, θ.

The void ratio can be derived from the porosity, n, using the relationship $e = n/(1-n)$. Up to six variables may be needed to define the fully saturated permeability, depending on whether isotropic, orthotropic, or fully anisotropic permeability is to be modeled (discussed below).

Alternative Definition of Permeability

Some authors refer to the definition of permeability used in Abaqus/Standard, $\overline{\mathbf{k}}$ (units of $\mathbf{LT}^{-1}$), as the "hydraulic conductivity" of the porous medium and define the permeability as

$$\widehat{\mathbf{K}} = \frac{\nu}{g}\frac{k_s}{\left(1+\beta\sqrt{\mathbf{v}_w \cdot \mathbf{v}_w}\right)}\mathbf{k} = \frac{\nu}{g}\overline{\mathbf{k}},$$

where ν is the kinematic viscosity of the wetting liquid (the ratio of the liquid's dynamic viscosity to its mass density), g is the magnitude of the gravitational acceleration, and $\widehat{\mathbf{K}}$ has dimensions L^2 (or Darcy). If the permeability is available in this form, it must be converted such that the appropriate values of $\mathbf{k}$ are used in Abaqus/Standard.

Specifying the Permeability

Permeability in Abaqus/Standard can be isotropic, orthotropic, or fully anisotropic. For nonisotropic permeability a local orientation (see *Orientations*) must be used to specify the material directions.

Isotropic Permeability

For isotropic permeability in Abaqus/Standard define one value of the fully saturated permeability at each value of the void ratio.

Input File Usage: **PERMEABILITY*, TYPE=ISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Type: Isotropic**

Orthotropic Permeability

For orthotropic permeability in Abaqus/Standard define three values of the fully saturated permeability (k_{11}, k_{22}, and k_{33}) at each value of the void ratio.

Input File Usage: **PERMEABILITY*, TYPE=ORTHOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Type: Orthotropic**

Anisotropic Permeability

For fully anisotropic permeability in Abaqus/Standard define six values of the fully saturated permeability (k_{11}, k_{12}, k_{22}, k_{13}, k_{23}, and k_{33}) at each value of the void ratio.

Input File Usage: **PERMEABILITY*, TYPE=ANISOTROPIC

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Type: Anisotropic**

Velocity Coefficient

Abaqus/Standard assumes that $\beta = 0.0$ by default, meaning that Darcy's law is used. If Forchheimer's law is required ($\beta > 0.0$), $\beta(e)$ must be defined in tabular form.

Input File Usage: **PERMEABILITY*, TYPE=VELOCITY

This must be a repeated use of the **PERMEABILITY* option for the same material, since $\mathbf{k}(e, \theta)$ must also be defined.

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Suboptions**->**Velocity Dependence**

Saturation Dependence

In Abaqus/Standard you can define the dependence of permeability, $\overline{\mathbf{k}}$, on saturation, *s*, by specifying k_s. Abaqus/Standard assumes by default that $k_s = s^3$ for $s < 1.0$; $k_s = 1.0$ for $s \geq 1.0$. The tabular definition of $k_s(s)$ must specify $k_s = 1.0$ for $s \geq 1.0$.

Input File Usage: **PERMEABILITY*, TYPE=SATURATION

This must be a repeated use of the **PERMEABILITY* option for the same material, since $\mathbf{k}(e, \theta)$ must also be defined.

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Suboptions**->**Saturation Dependence**

Specific Weight of the Wetting Liquid

In Abaqus/Standard the specific weight of the fluid, γ_w, must be specified correctly even if the analysis does not consider the weight of the wetting liquid (that is, if excess pore fluid pressure is calculated).

Input File Usage: **PERMEABILITY*, TYPE=*type*, SPECIFIC=γ_w

The SPECIFIC parameter must be defined in conjunction with the fully saturated **PERMEABILITY* option for a given medium.

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Permeability**: **Specific weight of wetting liquid: γ_w**

Specifying a Stabilization Coefficient

In Abaqus/Standard spurious pressure oscillations can occur in the solution for materials with very low permeability. You can add a stabilization term to the element operator and the right-hand-side term by projecting the pressure into the strain space to eliminate the pressure oscillations. For more information, see *Spurious Oscillations due to Small Time Increments*.

Input File Usage: **PERMEABILITY*, STABILIZE=*stabilization_coeff_value*

Abaqus/CAE Usage: Specifying a stabilization coefficient is not supported in Abaqus/CAE.

Elements

In Abaqus/Standard permeability can be used only in elements that allow for pore pressure (see *Choosing the Appropriate Element for an Analysis Type*).

Porous Bulk Moduli

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *About Pore Fluid Flow Properties*
- *About the Material Library*
- **POROUS BULK MODULI*
- *Defining porous bulk moduli*

Overview

The porous bulk moduli:

- must be defined whenever the compressibility of the solid grains or the compressibility of the permeating fluid is to be considered in the analysis of a porous medium in Abaqus/Standard;
- must be defined in an undrained pore fluid flow and stress analysis in Abaqus/Explicit; and
- must be defined when a swelling gel is modeled (*Moisture Swelling*).

Defining Porous Bulk Moduli

You can specify the bulk modulus of the solid grains and the bulk modulus of the fluid as functions of temperature. If either modulus is omitted or set to zero, that phase of the material is assumed to be fully incompressible.

Input File Usage: **POROUS BULK MODULI*

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Porous Bulk Moduli**

Specifying Field Variable Dependence for the Porous Bulk Moduli

The porous bulk moduli can be defined as functions of field variables. You must specify the number of user-defined field variables required.

Input File Usage: **POROUS BULK MODULI*, DEPENDENCIES=*n*

Abaqus/CAE Usage: Specifying field variable dependence for the porous bulk moduli is not supported in Abaqus/CAE.

Elements

The porous bulk moduli can be defined only for elements that allow for pore pressure in Abaqus/Standard (see *Choosing the Appropriate Element for an Analysis Type*) or for elements that support undrained pore fluid flow and stress analysis in Abaqus/Explicit (see *Undrained Pore Fluid Flow and Stress Analysis*).

Sorption

Products: Abaqus/Standard Abaqus/CAE

References:

- *About Pore Fluid Flow Properties*
- *About the Material Library*
- **SORPTION*
- *Defining sorption*

Overview

Sorption:

- defines a porous material's absorption/exsorption behavior under partially saturated flow conditions; and
- is used in the analysis of coupled wetting liquid flow and porous medium stress (*Coupled Pore Fluid Diffusion and Stress Analysis*).

Sorption

A porous medium becomes partially saturated when the total pore liquid pressure, u_w, becomes negative (see *Effective stress principle for porous media*). Negative values of u_w represent capillary effects in the medium. For $u_w < 0$ it is known that the saturation lies within certain limits that depend on the value of the capillary pressure, $-u_w$ (see *Continuity statement for the wetting liquid phase in a porous medium*). Typical forms of these limits are shown in *Figure 1*. We write these limits as $s^a \le s \le s^e$, where $s^a(u_w)$ is the limit at which absorption will occur (so that $\dot{s} > 0$), and $s^e(u_w)$ is the limit at which exsorption will occur (so that $\dot{s} < 0$). The transition between absorption and exsorption and vice versa takes place along "scanning" curves (discussed below). These curves are approximated by the single straight line shown in *Figure 1*.

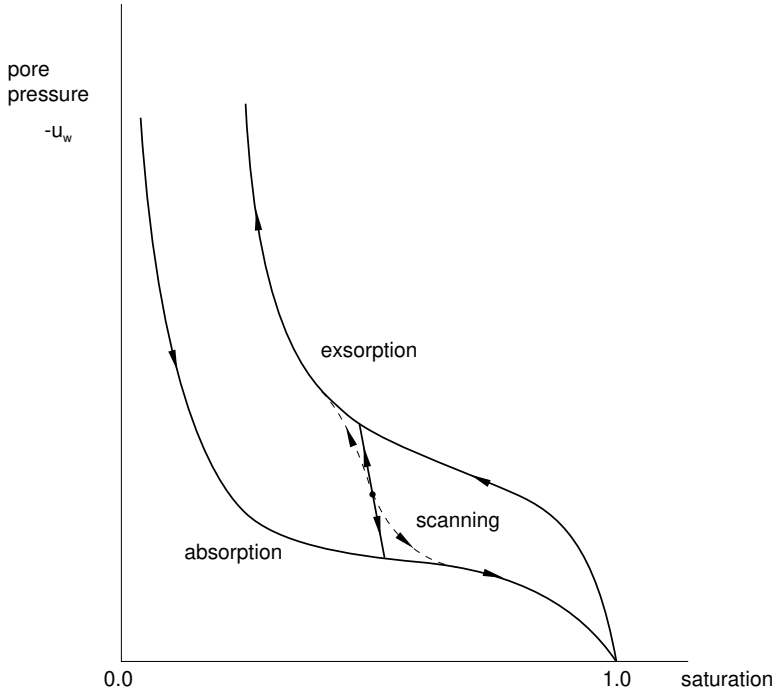


Figure 1: Typical absorption and exsorption behaviors.

When partial saturation is included in the analysis of flow through a porous medium, the absorption behavior, the exsorption behavior, and the scanning behavior (between absorption and exsorption) should each be defined. Each of these behaviors is discussed below. If sorption is not defined at all, Abaqus/Standard assumes fully saturated flow ($s = 1.0$) for all values of u_w.

Strongly unsymmetric partially saturated flow coupled equations result from the definition of sorption. Therefore, Abaqus/Standard automatically uses its unsymmetric matrix storage and solution scheme (see *Defining an Analysis*) if you request partially saturated analysis (i.e., if sorption is defined).

Defining Absorption and Exsorption

Absorption and exsorption behaviors are defined by specifying the pore liquid pressure, u_w (negative "capillary tension"), as a function of saturation. In most physical cases the wetting liquid cannot be driven to zero saturation; to achieve zero saturation, the data would have to define $u_w \to -\infty$ as $s \to 0.0$. Absorption and exsorption data can be defined in either a tabular form or an analytical form.

Tabular Form

By default, you define the absorption and exsorption behaviors by specifying u_w as a tabular function of s, where $0^+ \leq s \leq 1.0$.

Input File Usage: Use the following options:

**SORPTION*, TYPE=ABSORPTION, LAW=TABULAR
**SORPTION*, TYPE=EXSORPTION, LAW=TABULAR

If the **SORPTION* option is used only once, the behavior defined is taken as the behavior for absorption and exsorption.

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Sorption**
Absorption: **Law: Tabular**
Exsorption: toggle on **Include exsorption**: **Law: Tabular**

Analytical Form

The absorption and exsorption behaviors can be defined by the following analytical form:

$$u_w = \frac{1}{B} \ln \left[\frac{(s - s_0)}{(1 - s_0) + A(1 - s)} \right] \quad \text{for} \quad s_1 \le s < 1,$$

$$u_w = u_w|_{s_1} - \left. \frac{du_w}{ds} \right|_{s_1} (s_1 - s) \quad \text{for} \quad s_0 \le s < s_1,$$

where A, B are positive material constants and s_0, s_1 are parameters used to define the lower bound of the saturation values of interest (see *Figure 2*).

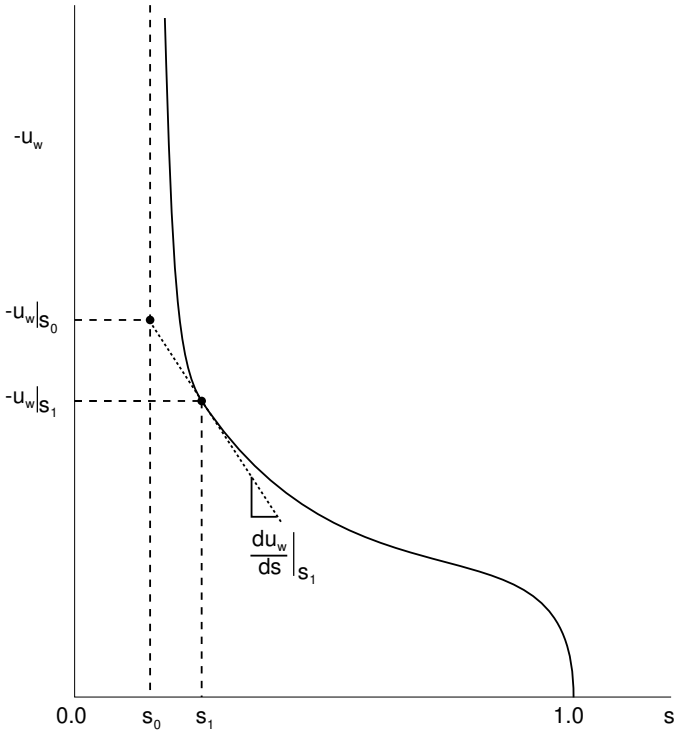


Figure 2: Logarithmic form of absorption and exsorption behaviors.

Input File Usage: Use the following options:

**SORPTION*, TYPE=ABSORPTION, LAW=LOG
**SORPTION*, TYPE=EXSORPTION, LAW=LOG

If the **SORPTION* option is used only once, the behavior defined is taken as the behavior for absorption and exsorption.

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Sorption**
Absorption: **Law: Log**
Exsorption: toggle on **Include exsorption**: **Law: Log**

Defining the Behavior between Absorption and Exsorption

The behavior between absorption and exsorption is defined by a scanning line of user-specified constant slope, $(du_w/ds)|_s$. This slope should be larger than the slope of any segment of the absorption or exsorption behaviors. If absorption and exsorption behaviors are defined with no scanning line, the slope of the scanning line is taken as 1.05 times the largest value of du_w/ds given in the absorption and exsorption behavior definitions.

Input File Usage: **SORPTION*, TYPE=SCANNING

This must be a repeated use of the **SORPTION* option for the same material.

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Sorption**: **Exsorption**: toggle on **Include exsorption** and **Include scanning**: **Slope** $(du_w/ds)|_s$

Elements

Sorption can be used only in elements that allow for pore pressure (see *Choosing the Appropriate Element for an Analysis Type*).

Swelling Gel

Products: Abaqus/Standard Abaqus/CAE

References:

- *About Pore Fluid Flow Properties*
- *About the Material Library*
- **GEL*
- *Defining a swelling gel*

Overview

The swelling gel model:

- allows for modeling of the growth of gel particles that swell and trap wetting liquid in a partially saturated porous medium;
- is intended for use in moisture absorption problems, which typically involve polymeric materials, such as in the analysis of diapers; and
- can be used in the analysis of coupled pore liquid flow and porous medium stress (see *Coupled Pore Fluid Diffusion and Stress Analysis*).

Swelling Gel Model

The simple swelling gel model is based on the idealization of a gel as a volume of individual spherical particles of equal radius, r_a. The swelling evolution (discussed in detail in *Constitutive behavior in a porous medium*) is assumed to be given by

$$\dot{r}_a = \frac{r_a^f - r_a}{\tau_1} \left\langle s - 1 + \left(\frac{\left(r_a^f\right)^3 - \left(r_a\right)^3}{\left(r_a^f\right)^3 - \left(r_a^{\text{dry}}\right)^3} \right) \right\rangle \left(1 - \left\langle \frac{r_a - r_a^t}{r_a^s - r_a^t} \right\rangle^2 \right),$$

where the value of any grouping of terms in angled brackets $\langle\rangle$ is set equal to zero if its mathematical result is not positive, and

r_a^f

is the fully swollen radius;

τ_1

is the relaxation time of the gel particles;

s

is the saturation of the surrounding medium;

r_a^{dry}

is the radius of the gel particles when they are completely dry;

$r_a^t = \left(\frac{n^0 J}{4\sqrt{2}k_a}\right)^{\frac{1}{3}}$

is the maximum radius that the gel particles can achieve before they must touch;

$r_a^s = \left(\frac{3}{4\pi}\frac{n^0 J}{k_a}\right)^{\frac{1}{3}}$

is the effective gel radius when the volume is entirely occupied with gel;

n^0

is the initial porosity of the material;

J

is the volume change in the material; and

k_a

is the number of gel particles per unit volume.

The second term in the definition of gel growth incorporates the assumption that the gel will swell only when the saturation of the surrounding medium, *s*, exceeds the effective saturation of the gel. The third term in the growth equation reduces the swelling rate when the surface of gel particles exposed to free fluid is limited by the combination of packing density and gel particle radius.

The swelling gel model is defined by specifying the variables r_a^{dry}, r_a^f, k_a, and τ_1.

Input File Usage: **GEL*

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Gel**

Elements

The swelling gel model can be used only in elements that allow for pore pressure (see *Choosing the Appropriate Element for an Analysis Type*).

Moisture Swelling

Products: Abaqus/Standard Abaqus/CAE

References:

- *About Pore Fluid Flow Properties*
- *About the Material Library*
- **MOISTURE SWELLING*
- *Defining moisture swelling*

Overview

Moisture swelling:

- defines the saturation-driven volumetric swelling of the solid skeleton of a porous medium in partially saturated flow conditions;
- can be used in the analysis of coupled wetting liquid flow and porous medium stress (see *Coupled Pore Fluid Diffusion and Stress Analysis*); and
- can be either isotropic or anisotropic.

Moisture Swelling Model

The moisture swelling model assumes that the volumetric swelling of the porous medium's solid skeleton is a function of the saturation of the wetting liquid in partially saturated flow conditions. The porous medium is partially saturated when the pore liquid pressure, u_w, is negative (see *Effective stress principle for porous media*).

The swelling behavior is assumed to be reversible. The logarithmic measure of swelling strain is calculated with reference to the initial saturation so that

$$\varepsilon_{ii}^{ms} = r_{ii}\frac{1}{3}\left(\varepsilon^{ms}\left(s\right) - \varepsilon^{ms}\left(s^I\right)\right), \quad \text{(no sum on } i\text{)}$$

where $\varepsilon^{ms}\left(s\right)$ and $\varepsilon^{ms}\left(s^I\right)$ are the volumetric swelling strains at the current and initial saturations. A typical curve is shown in *Figure 1*. The ratios r_{11}, r_{22}, and r_{33} allow for anisotropic swelling as discussed below.

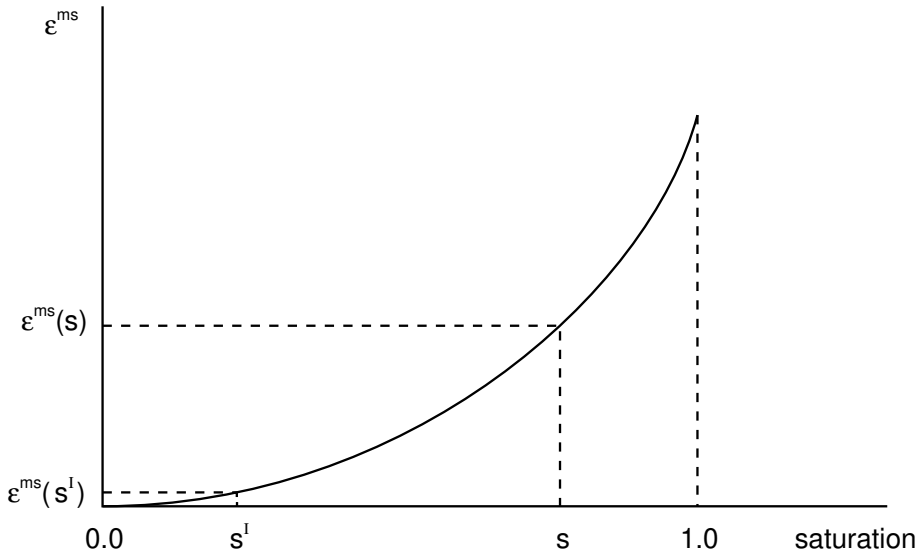


Figure 1: Typical volumetric moisture swelling versus saturation curve.

Defining Volumetric Swelling Strain

Define the volumetric swelling strain, ε^{ms}, as a tabular function of the wetting liquid saturation, s. The swelling strain must be defined for the range $0.0 \leq s \leq 1.0$.

Input File Usage: **MOISTURE SWELLING*

Abaqus/CAE Usage: Property module: material editor: **Other->Pore Fluid->Moisture Swelling**

Defining Initial Saturation Values

You can define the initial saturation values as initial conditions. If no initial saturation values are given, the default is fully saturated conditions (saturation of 1.0). For partial saturation the initial saturation and pore fluid pressure must be consistent, in the sense that the pore fluid pressure must lie within the absorption and exsorption values for the initial saturation value (see *Permeability*). If this is not the case, Abaqus/Standard will adjust the saturation value as needed to satisfy this requirement.

Input File Usage: **INITIAL CONDITIONS*, TYPE=SATURATION

Abaqus/CAE Usage: Load module: **Create Predefined Field**: **Step: Initial**: choose **Other** for the **Category** and **Saturation** for the **Types for Selected Step**

Defining Anisotropic Swelling

Anisotropy can be included in moisture swelling behavior by defining the ratios r_{11}, r_{22}, and r_{33}, such that two or more of the three ratios differ. If the ratios r_{ii} are not specified, Abaqus/Standard assumes that the swelling is isotropic and that $r_{11} = r_{22} = r_{33} = 1.0$. The orientation of the moisture swelling strain directions depends on the user-specified local orientation (see *Orientations*).

Input File Usage: Use both of the following options:

**MOISTURE SWELLING*
**RATIOS*

The **RATIOS* option should immediately follow the **MOISTURE SWELLING* option.

Abaqus/CAE Usage: Property module: material editor: **Other**->**Pore Fluid**->**Moisture Swelling**: **Suboptions**->**Ratios**

Elements

The moisture swelling model can be used only in elements that allow for pore pressure (see *Choosing the Appropriate Element for an Analysis Type*).

User Materials

In this section:

- *User-Defined Mechanical Material Behavior*
- *User-Defined Thermal Material Behavior*
- *User-Defined Material Properties*

User-Defined Mechanical Material Behavior

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *UMAT*
- *VUMAT*
- **USER MATERIAL*
- **DEPVAR*
- *Specifying solution-dependent state variables*
- *Defining constants for a user material*

Overview

User-defined mechanical material behavior in Abaqus:

- is provided by means of an interface whereby any mechanical constitutive model can be added to the library;
- requires that a constitutive model (or a library of models) is programmed in user subroutine *UMAT* (Abaqus/Standard) or *VUMAT* (Abaqus/Explicit); and
- requires considerable effort and expertise: the feature is very general and powerful, but its use is not a routine exercise.

Stress Components and Strain Increments

The subroutine interface has been implemented using Cauchy stress components ("true" stress). For soils problems "stress" should be interpreted as effective stress. The strain increments are defined by the symmetric part of the displacement increment gradient (equivalent to the time integral of the symmetric part of the velocity gradient).

The orientation of the stress and strain components in user subroutine *UMAT* depends on the use of local orientations (*Orientations*).

In user subroutine *VUMAT* all strain measures are calculated with respect to the midincrement configuration. All tensor quantities are defined in the corotational coordinate system that rotates with the material point. To illustrate what this means in terms of stresses, consider the bar shown in *Figure 1*, which is stretched and rotated from its original configuration, AB, to its new position, $A'B'$. This deformation can be obtained in two stages; the bar is first stretched, as shown in *Figure 2*, and is then rotated by applying a rigid body rotation to it, as shown in *Figure 3*.

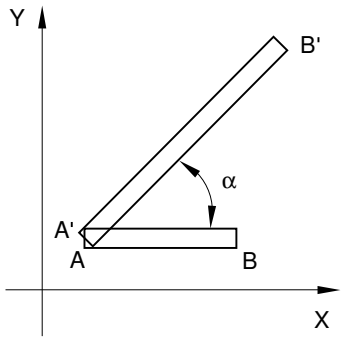


Figure 1: Stretched and rotated bar.

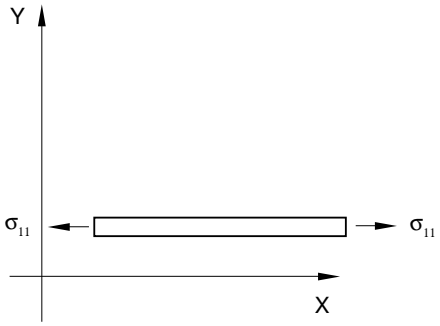


Figure 2: Stretching of bar.

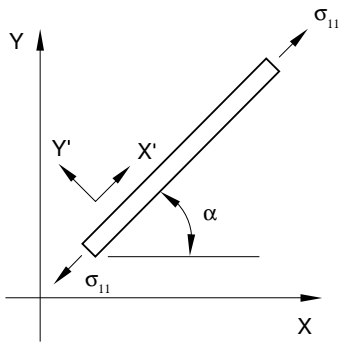


Figure 3: Rigid body rotation of bar.

The stress in the bar after it has been stretched is σ_{11}, and this stress does not change during the rigid body rotation. The $X'Y'$ coordinate system that rotates as a result of the rigid body rotation is the corotational coordinate system. The stress tensor and state variables are, therefore, computed directly and updated in user subroutine *VUMAT* using the strain tensor since all of these quantities are in the corotational system; these quantities do not have to be rotated by the user subroutine as is sometimes required in user subroutine *UMAT*.

The elastic response predicted by a rate-form constitutive law depends on the objective stress rate employed. For example, the Green-Naghdi stress rate is used in *VUMAT*. However, the stress rate used for built-in material models may differ. For example, most material models used with solid (continuum) elements in Abaqus/Explicit employ the Jaumann stress rate. This difference in the formulation will cause significant differences in the results only if finite rotation of a material point is accompanied by finite shear. For a discussion of the objective stress rates used in Abaqus, see *Stress rates*.

Material Constants

Any material constants that are needed in user subroutine *UMAT* or *VUMAT* must be specified as part of a user-defined material behavior definition. Any other mechanical material behaviors included in the same material definition (except thermal expansion and, in Abaqus/Explicit, density) will be ignored; the user-defined material behavior requires that all mechanical material behavior calculations be programmed in subroutine *UMAT* or *VUMAT*. In Abaqus/Explicit the density (*Density*) is required when using a user-defined material behavior.

Input File Usage: In Abaqus/Standard use the following option to specify a user-defined material behavior:

**USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*

In Abaqus/Explicit use both of the following options to specify a user-defined material behavior:

**USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*
**DENSITY*

In either case you must specify the number of material constants being entered.

Abaqus/CAE Usage: In Abaqus/Standard use the following option to specify a user-defined material behavior:

Property module: material editor: **General->User Material**: **User material type: Mechanical**

In Abaqus/Explicit use both of the following options to specify a user-defined material behavior:

Property module: material editor:
General->User Material: **User material type: Mechanical**
General->Density

Unsymmetric Equation Solver in Abaqus/Standard

If the user material's Jacobian matrix, $\partial \triangle \boldsymbol{\sigma} / \partial \triangle \boldsymbol{\varepsilon}$, is not symmetric, the unsymmetric equation solution capability in Abaqus/Standard should be invoked (see *Defining an Analysis*).

Input File Usage: **USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*, UNSYMM

Abaqus/CAE Usage: Property module: material editor: **General->User Material**: **User material type: Mechanical**, toggle on **Use unsymmetric material stiffness matrix**

Hybrid Formulation in Abaqus/Standard

If you use a hybrid element with user subroutine *UMAT*, by default Abaqus/Standard replaces the pressure stress calculated from the stress tensor returned by the user subroutine with that derived from the Lagrange multiplier and modifies the Jacobian appropriately (*Hybrid incompressible solid element formulation*). This approach is suitable for material models that use an incremental formulation (for example, metal plasticity) but is not consistent

with the total formulation that is commonly used for hyperelastic materials. In the latter situation the default formulation may lead to convergence problems. Such convergence problems may be observed, for example, when an almost incompressible nonlinear elastic user material is subjected to large deformations. Abaqus/Standard provides an alternate total formulation that is more appropriate in such situations. The total formulation is consistent with the native almost incompressible formulation used by Abaqus for hyperelastic materials (*Hyperelastic material behavior*), and works better than the default (incremental) formulation for such cases.

Abaqus/Standard also provides a fully incompressible formulation for use with hybrid elements to define a fully incompressible user material response. The fully incompressible formulation is consistent with the native formulation used by Abaqus for incompressible hyperelastic materials. For the total hybrid formulation it is assumed that the deviatoric and the volumetric responses of the material are decoupled and that the volumetric response can be derived from a strain energy potential function. All the native hyperelastic materials in Abaqus use this assumption. For the incompressible hybrid formulation, it is assumed that the deviatoric stress can be derived from a strain energy potential function.

The total hybrid formulation is useful for an almost incompressible hyperelastic response. The volumetric response of the material is assumed to be defined in terms of an alternate variable, $\widehat{J}$, in place of the volume change, J. The alternate variable is made available inside user subroutine *UMAT*. Further details are discussed in *UMAT*.

The fully incompressible formulation requires you to define only the deviatoric parts of the stress tensor and the material's Jacobian matrix inside the *UMAT*. Abaqus/Standard automatically accounts for the pressure stress based on the Lagrange multiplier.

Input File Usage: Use the following option to invoke the total hybrid formulation:

**USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*,
HYBRID FORMULATION=TOTAL

Use following option to invoke the incremental hybrid formulation (default):

**USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*,

HYBRID FORMULATION=INCREMENTAL

Use the following option to invoke the incompressible hybrid formulation:

**USER MATERIAL*, TYPE=MECHANICAL, CONSTANTS=*number_of_constants*,

HYBRID FORMULATION=INCOMPRESSIBLE

Abaqus/CAE Usage: Property module: material editor: **General->User Material**: **User material type: Mechanical**, **Hybrid formulation**: **Specify**: **Total**, **Incremental**, **Incompressible**, or **None** (default)

Material State

Many mechanical constitutive models require the storage of solution-dependent state variables (plastic strains, “back stress,” saturation values, etc. in rate constitutive forms or historical data for theories written in integral form). You should allocate storage for these variables in the associated material definition (see *Allocating Space for Solution-Dependent State Variables*). There is no restriction on the number of state variables associated with a user-defined material.

The user material subroutines are provided with the material state at the start of each increment, as described below. They must return values for the new stresses and the new internal state variables. State variables associated with *UMAT* and *VUMAT* can be output to the output database file (`.odb`) and results file (`.fil`) using the output identifiers SDV and SDV*n* (see *Abaqus/Standard Output Variable Identifiers* and *Abaqus/Explicit Output Variable Identifiers*).

Material State in Abaqus/Standard

User subroutine *UMAT* is called for each material point at each iteration of every increment. It is provided with the material state at the start of the increment (stress, solution-dependent state variables, temperature, and any predefined field variables) and with the increments in temperature, predefined state variables, strain, and time.

In addition to updating the stresses and the solution-dependent state variables to their values at the end of the

increment, subroutine *UMAT* must also provide the material Jacobian matrix, $\partial \Delta \boldsymbol{\sigma} / \partial \Delta \boldsymbol{\varepsilon}$, for the mechanical constitutive model. This matrix will also depend on the integration scheme used if the constitutive model is in rate form and is integrated numerically in the subroutine. For any nontrivial constitutive model these will be challenging tasks. For example, the accuracy with which the Jacobian matrix is defined will usually be a major determinant of the convergence rate of the solution and, therefore, will have a strong influence on computational efficiency.

If you specify the viscoelastic behavior of materials in the frequency domain, user subroutine *UMAT* must also provide the damping (loss modulus) contribution to the material Jacobian matrix, in addition to the stiffness (storage modulus) contribution.

Material State in Abaqus/Explicit

User subroutine *VUMAT* is called for blocks of material points at each increment. When the subroutine is called, it is provided with the state at the start of the increment (stress, solution-dependent state variables). It is also provided with the stretches and rotations at the beginning and the end of the increment. The *VUMAT* user material interface passes a block of material points to the subroutine on each call, which allows vectorization of the material subroutine.

The temperature is provided to user subroutine *VUMAT* at the start and the end of the increment. The temperature is passed in as information only and cannot be modified, even in a fully coupled thermal-stress analysis. However, if the inelastic heat fraction is defined in conjunction with the specific heat and conductivity in a fully coupled thermal-stress analysis in Abaqus/Explicit, the heat flux due to inelastic energy dissipation will be calculated automatically. If the *VUMAT* user subroutine is used to define an adiabatic material behavior (conversion of plastic work to heat) in an explicit dynamics procedure, you must specify both the inelastic heat fraction and the specific heat for the material, and you must store the temperatures and integrate them as user-defined state variables. Most often the temperatures are provided by specifying initial conditions (*Initial Conditions*) and are constant throughout the analysis.

Deleting Elements from a Mesh Using State Variables

Element deletion in a mesh can be controlled during the course of an Abaqus analysis through user subroutine *VUMAT* or *UMAT*. Deleted elements have no ability to carry stresses and, therefore, have no contribution to the stiffness of the model. You specify the state variable number controlling the element deletion flag. For example, specifying a state variable number of 4 indicates that the fourth state variable is the deletion flag in the user subroutine. The deletion state variable should be set to a value of one or zero. A value of one indicates that the material point is active, while a value of zero indicates that Abaqus should delete the material point from the model by setting the stresses to zero. In Abaqus/Explicit the structure of the block of material points passed to user subroutine *VUMAT* remains unchanged during the analysis; deleted material points are not removed from the block. Abaqus/Explicit passes zero stresses and strain increments for all deleted material points. Once a material point is flagged as deleted, it cannot be reactivated. An element is deleted from a mesh based on the material point status (active or deleted). Details for element deletion driven by material failure are described in

Material Failure and Element Deletion. The status of a material point and an element can be determined by requesting output variables STATUSMP and STATUS, respectively.

Input File Usage: **DEPVAR*, DELETE=*variable number*

Abaqus/CAE Usage: Property module: material editor: **General->Depvar**: **Variable number controlling element deletion:** *variable number*

Scaling the Transverse Shear Stiffness for Shell Elements Using State Variables

In an Abaqus/Explicit analysis, you can control the transverse shear stiffness for shell elements with user subroutine *VUMAT*. You specify the state variable number controlling the shell element transverse shear damage variable. For example, specifying a state variable number of 5 indicates that the fifth state variable is the shell element transverse shear damage variable in the user subroutine. You can set the state variable to a value between zero and one, with a default value of one indicating the initial undamaged state. The transverse shear stiffness scaling factor is computed using a weighted average through the section of the state values that you define. The transverse shear stiffness scaling factor is used to scale the transverse shear stiffness of the shell elements.

Input File Usage: **DEPVAR*, TVS DAMAGE=*variable number*

Abaqus/CAE Usage: Scaling the transverse shear stiffness for shell elements using state variables is not supported in Abaqus/CAE.

Hourglass Control and Transverse Shear Stiffness

Normally the default hourglass control stiffness for reduced-integration elements in Abaqus/Standard and the transverse shear stiffness for shell, pipe, and beam elements are defined based on the elasticity associated with the material (*Section Controls*, *Shell Section Behavior*, and *Choosing a Beam Element*). These stiffnesses are based on a typical value of the initial shear modulus of the material, which may, for example, be given as part of an elastic material behavior (*Linear Elastic Behavior*) included in the material definition. However, the shear modulus is not available during the preprocessing stage of input for materials defined with user subroutine *UMAT* or *VUMAT*. Therefore, you must provide the hourglass stiffness parameters (see *Methods for Suppressing Hourglass Modes*) when using *UMAT* to define the material behavior of elements with hourglassing modes.

You must specify the transverse shear modulus for the material (see *Defining the Elastic Transverse Shear Modulus*) or the transverse shear stiffness for the section (see *Choosing a Beam Element* or *Shell Section Behavior*) when using user subroutine *UMAT* or *VUMAT* to define the material behavior of beams and shells with transverse shear flexibility. If the transverse shear modulus is specified for a user-defined mechanical material that is associated with shells, Abaqus calls user subroutine *UMAT* or *VUMAT* in the data check phase of the analysis to obtain the initial plane stress elastic stiffness. It then computes the transverse shear stiffness by matching the shear response for the case of the shell bending about one axis (see *Transverse shear stiffness in composite shells and offsets from the midsurface*).

Defining the Effective Modulus to Control Time Incrementation in Abaqus/Explicit

The stable time increment in Abaqus/Explicit is a function of the dilatational wave speed of the material and the characteristic length of the element. By default, the dilatational wave speed is determined automatically by Abaqus/Explicit using a numerical calculation of the effective hypoelastic bulk and shear moduli from the user material's constitutive response. In situations when the user material is highly nonlinear, this calculation may not yield a conservative value of the time increment, leading to unstable behavior. To avoid these situations, you can define the values of the effective bulk and shear moduli directly inside user subroutine *VUMAT*. These

values are then used by Abaqus/Explicit to compute the dilatational wave speed and the stable time increment. For more information, see *Stability*.

Input File Usage: **USER MATERIAL*, EFFMOD

Abaqus/CAE Usage: Property module: material editor: **General->User Material**: **User material type: Mechanical**, toggle on **VUMAT defines effective modulus**

Use with Other Subroutines

In Abaqus/Standard user subroutine *UMATHT* can be used in conjunction with *UMAT* to define the constitutive thermal behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both *UMAT* and *UMATHT*. In addition, user subroutines *FRIC*, *GAPCON*, and *GAPELECTR* are available for defining mechanical, thermal, and electrical interactions between surfaces.

In Abaqus/Explicit user subroutine *VUMATHT* can be used in conjunction with user subroutine *VUMAT* to define the constitutive thermal behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both *VUMAT* and *VUMATHT*. In addition, user subroutines *VFRIC*, *VFRICTION*, *VUINTER*, and *VUINTERACTION* are availabe for defining interfacial constitutive behavior.

Material Options

A number of material behaviors can be used in the definition of a material when its mechanical behavior is defined by user subroutine *UMAT* or *VUMAT*. These behaviors include density, thermal expansion, permeability, and heat transfer properties. Thermal expansion can alternatively be an integral part of the constitutive model implemented in *UMAT* or *VUMAT*.

The temperature available in *UMAT* is always the interpolated temperature field at the element integration points. Naturally, if the thermal expansion behavior is implemented in *UMAT*, it is defined in terms of the integration point temperature. When the temperature field is interpolated differently within an element compared to the displacement field in Abaqus/Standard, implementing the thermal expansion behavior in *UMAT* may lead to differences compared to the built-in thermal expansion behavior. This situation commonly arises for coupled temperature-displacement elements. For example, for first-order coupled temperature-displacement elements, the built-in thermal expansion behavior uses a constant temperature field over the whole element (see *Fully Coupled Thermal-Stress Analysis*), while the behavior in *UMAT* will be defined in terms of a linear temperature field.

For a material defined by user subroutine *UMAT* or *VUMAT*, mass proportional damping can be included separately (see *Material Damping*), but stiffness proportional damping must be defined in the user subroutine by the Jacobian (Abaqus/Standard only) and stress definitions. Stiffness proportional damping cannot be specified if the user material is used in the direct steady-state dynamics procedure.

Elements

User subroutines *UMAT* and *VUMAT* can be used with all elements in Abaqus that include mechanical behavior (elements that have displacement degrees of freedom).

User-Defined Thermal Material Behavior

Products: Abaqus/Standard Abaqus/Explicit Abaqus/CAE

References:

- *UMATHT*
- *VUMATHT*
- **USER MATERIAL*
- **DEPVAR*
- *Defining constants for a user material*

Overview

User-defined thermal material behavior in Abaqus:

- is provided by means of an interface whereby any thermal constitutive model can be added to the library;
- requires that a constitutive model (or a library of models) is programmed in user subroutine *UMATHT* (Abaqus/Standard) or *VUMATHT* (Abaqus/Explicit);
- must appear in conjunction with a density definition (see *Density*); and
- requires considerable effort and expertise: the feature is very general and powerful, but its use is not a routine exercise.

Material Constants

Any material constants that are needed in user subroutine *UMATHT* or *VUMATHT* must be specified as part of a user-defined thermal material behavior definition. Any other thermal material behaviors included in the same material definition will be ignored: the user-defined thermal material behavior requires that all thermal behavior calculations are programmed in user subroutine *UMATHT* or *VUMATHT*.

Input File Usage: **USER MATERIAL*, TYPE=THERMAL, CONSTANTS=*number_of_constants*

You must specify the number of constants being entered.

Abaqus/CAE Usage: Property module: material editor: **General->User Material**: **User material type: Thermal**

Unsymmetric Equation Solver in Abaqus/Standard

When the conductivity is defined in user subroutine *UMATHT* as a strong function of temperature, the heat transfer equilibrium equations become nonsymmetric and you may choose to invoke the unsymmetric equation solution capability; otherwise, convergence may be poor.

Input File Usage: **USER MATERIAL*, TYPE=THERMAL, CONSTANTS=*number_of_constants*, UNSYMM

Abaqus/CAE Usage: Property module: material editor: **General->User Material**: **User material type: Thermal**, toggle on **Use unsymmetric material stiffness matrix**

Material State

Many thermal constitutive models require the storage of solution-dependent state variables. These state variables might include microstructure or phase content information when the material undergoes phase changes. You should allocate storage for these variables in the associated material definition (see *Allocating Space for Solution-Dependent State Variables*). There is no restriction on the number of state variables associated with a user-defined material.

User subroutine *UMATHT* is called for each material point at each iteration of every increment. It is provided with the thermal state of the material at the start of the increment (solution-dependent state variables, temperature, and any predefined field variables) and with the increments in temperature, predefined state variables, and time.

User subroutine *VUMATHT* is called for blocks of material points at each increment. When the subroutine is called, it is provided with the state at the start of the increment (temperature, internal energy per unit mass, solution-dependent variables, and any predefined field variables) and the state at the end of the increment (temperature, spatial gradients of temperature). The *VUMATHT* user material interface passes a block of material points to the subroutine on each call, which allows vectorization of the material subroutine.

Required Calculations

User subroutine *UMATHT* must perform the following functions: it must define the internal energy per unit mass and its variation with respect to temperature and spatial gradients of temperature; it must define the heat flux vector and its variation with respect to temperature and spatial gradients of temperature; and it must update the solution-dependent state variables to their values at the end of the increment. The components of the heat flux and spatial gradients in user subroutine *UMATHT* are in directions that depend on the use of local orientations (see *Orientations*).

User subroutine *VUMATHT* must return the following values at the end of the increment: the internal energy per unit mass and its variation with respect to temperature, the heat flux vector, solution-dependent state variables and an effective conductivity. The effective conductivity is required to estimate the element stable time increment size in Abaqus/Explicit. Since the stable time increment size is inversely proportional to conductivity, you can provide the largest value in conductivity tensor to obtain a conservative estimation of the stable time increment size.

Use with Other User Subroutines

Any other user subroutine that does not deal with constitutive thermal behavior for a material can be used in conjunction with user subroutines *UMATHT* and *VUMATHT*.

In Abaqus/Standard user subroutine *UMAT* can be used in conjunction with *UMATHT* to define the constitutive mechanical behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both *UMATHT* and *UMAT*. In addition, user subroutines *FRIC*, *GAPCON*, and *GAPELECTR* are available for separately defining mechanical, thermal, and electrical interactions between surfaces; and user subroutine *UINTER* is available for defining general interfacial constitutive behavior. User subroutine *HETVAL* for user-defined internal heat generation can be used in conjunction with user subroutine *UMATHT*.

In Abaqus/Explicit user subroutine *VUMAT* can be used in conjunction with user subroutine *VUMATHT* to define the constitutive mechanical behavior of the material. The solution-dependent variables allocated in the material definition are accessible in both *VUMATHT* and *VUMAT*. In addition, user subroutines *VFRIC*, *VFRICTION*, *VUINTER*, and *VUINTERACTION* are available for defining interfacial constitutive behavior. User subroutine *VHETVAL* for user-defined internal heat generation can be used in conjunction with user subroutine *VUMATHT*.

Material Options

Density, mechanical properties, and electrical properties can be included in the definition of a material whose constitutive thermal behavior is defined by user subroutine *UMATHT* or *VUMATHT*.

Elements

User subroutine *UMATHT* can be used with all elements in Abaqus/Standard that include thermal behavior (elements with temperature degrees of freedom such as pure heat transfer, coupled thermal-stress, and coupled thermal-electrical elements).

User subroutine *VUMATHT* can be used with thermally coupled continuum elements and Eulerian elements in Abaqus/Explicit.

User-Defined Material Properties

Products: Abaqus/Standard

References:

- **PARAMETER TABLE*
- **PROPERTY TABLE*
- *Accessing Abaqus Table Collections*

Overview

User-defined material properties:

- can be included in material definitions as property tables and parameter tables; and
- can be accessed from any material-related user subroutine that is invoked for the material for which the properties are defined.

Defining Material Properties

You can define property tables and parameter tables within a material definition. Abaqus creates an internal table collection.

Input File Usage: Use the following options to define property tables and parameter tables:

**MATERIAL*, NAME=*material_name*
**PROPERTY TABLE*, TYPE=*property_table_type*
**PARAMETER TABLE*, TYPE=*parameter_table_type*

Accessing Material Properties

You can access property tables and parameter tables from any material-related user subroutine that is invoked for the material for which the tables are defined. Abaqus activates an appropriate internal table collection. You can access these tables using the utility routines described in *Accessing Abaqus Table Collections*.

Elements

User-defined material properties can be used with all elements in Abaqus/Standard.

Special-Purpose Material Modeling Techniques

In this section:

- *About Special-Purpose Material Modeling Techniques*
- *Modeling the Cure Process in Thermosetting Polymers*
- *Modeling Metallurgical Phase Transformations*
- *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*
- *Modeling The Electrical and Mechanical Response Of Cardiac Tissue*
- *Parameter Table Type Reference*
- *Property Table Type Reference*

About Special-Purpose Material Modeling Techniques

Overview

Abaqus provides several material modeling capabilities intended for special-purpose applications.

The available special-purpose material modeling techniques include the following:

- Modeling the cure process in thermosetting polymers
- Modeling metallurgical phase transformations
- Modeling cardiac tissue
- Modeling soft matter using constitutive artificial neural networks (CANNs)
- Modeling fiber-reinforced composites using deep material networks (DMNs)
- Other special-purpose material applications (for example, the Holmquist-Johnson-Cook model for concrete)

These techniques are based on specialized implementations of Abaqus user subroutines and are referred to as "built-in material models" because they are built directly into the Abaqus software.

You can specify a built-in material model for your analysis either by defining the appropriate material user type or by choosing a material name that starts with the reserved string `ABQ_`.

Material User Type Definition

Material user types are supported only for the special-purpose material modeling techniques described in *Table 1* below. Each model has a distinct user type label.

Table 1: User type labels for special-purpose material modeling techniques.

Special-purpose material modeling technique description	User type label
Modeling cardiac tissue in the prestress mechanical analysis For more information, see the *Living Heart Model Guide*.	CARDIAC_INITSTRESS_HYBRID
Modeling cardiac tissue in the electrical analysis For more information, see *Modeling the Electrical Response of Cardiac Tissue*.	CARDIAC_TISSUE
Modeling cardiac tissue in the heartbeat mechanical analysis For more information, see *Modeling the Mechanical Response of Cardiac Tissue*.	CARDIAC_TISSUE_HYBRID
Modeling metallurgical phase transformations For more information, see *Modeling Metallurgical Phase Transformations*.	PHASE_TRANSFORMATION
Modeling soft matter using constitutive artificial neural networks (CANNs) For more information, see *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*.	CANN

Special-purpose material modeling technique description	User type label
Modeling fiber-reinforced composites (aligned fiber directions) using deep material networks (DMNs) For more information, see *Modeling Multiscale Materials with Deep Material Networks (DMNs)*.	DMN_UD
Modeling fiber-reinforced composites (dispersed fiber directions) using deep material networks (DMNs) For more information, see *Modeling Multiscale Materials with Deep Material Networks (DMNs)*.	DMN_CHOPPED

Input File Usage: Use the following option to define the material user type label:

**MATERIAL*, NAME=*material_name*, USER TYPE=*user_type_label*

Reserved Material Name Definition

You define all other special-purpose material applications (for example, the Holmquist-Johnson-Cook model for concrete) by calling the appropriate material name. These material names start with the reserved prefix `ABQ_`. These models are documented in the Dassault Systèmes Knowledge Base, and you can download the relevant files directly from the Knowledge Base articles.

For the complete list of special-purpose material applications available through the Knowledge Base, see *Built-in user material models for Abaqus/Standard and Abaqus/Explicit*)

Input File Usage: Use the following option to define the special-purpose material application:

**MATERIAL*, NAME=ABQ_*special_purpose_material_name*

Modeling the Cure Process in Thermosetting Polymers

Products: Abaqus/Standard

References:

- *Elastic Behavior*
- *Time Domain Viscoelasticity*
- **EIGENSTRAIN*
- **PARAMETER TABLE*
- **PARAMETER TABLE TYPE*
- **PROPERTY TABLE*
- **PROPERTY TABLE TYPE*

Overview

The cure modeling capabilities:

- are intended to model the curing process in adhesives and other polymer materials;
- predict the degree of cure, volumetric heat generation, and shrinkage strain due to curing reactions;
- allow you to specify a maximum value of the degree of cure;
- are intended for use with existing elastic and viscoelastic behaviors that describe the mechanical response of the material as a function of the degree of cure; and
- must be used with the fully coupled thermal-stress analysis (see *Fully Coupled Thermal-Stress Analysis*).

Alternatively, Abaqus offers another method of defining the cure model that uses dedicated cure modeling options and output variables (see *Curing Processes in Polymers*). This method is more general and is intended to supersede the capabilities described in this section.

Curing processes are essential to the manufacturing of products that use thermosetting polymers (such as epoxy resins) to bond components. The use of epoxies and other cured structural adhesives is common in many industries. As a result of the curing reaction, chemical shrinkage strains and residual stresses develop, which can result in damage to the adherends or warpage of the bonded assembly. The Abaqus cure modeling capability enables you to analyze curing processes, including the reaction kinetics, heat generation, shrinkage strain development, and the evolution of mechanical properties. The model is based on the work of *Lindeman et al. (2021)* and *Li et al. (2004)*. You can describe the reaction kinetics using either the Kamal equation or a conversion rate table. The mechanical response can include both elastic and viscoelastic effects. You can define the elastic, viscoelastic, and thermal expansion properties of the material as functions of conversion and temperature.

The cure model is available as a special-purpose material modeling capability based on built-in user-defined material options. To activate the capability, the material name must start with `"ABQ_CURE_MATERIAL"` and the material definition must include a user-defined eigenstrain definition. In addition, you must define the cure modeling coefficients using parameter and property tables with a declared type that starts with `"ABQ_Cure"` (as described in the following sections). You must allocate at least three solution-dependent state variables.

A dedicated collection of parameter and property tables is available to include all of the definitions required to use the model. You can use the **abaqus fetch** utility to obtain the file containing the type definitions for the parameter and property tables used by the cure model:

```
abaqus fetch job=ABQ_Cure_types.inp
```

Input File Usage: Use the following options to start the definition of the material based on built-in user-defined material options:

**INCLUDE*, INPUT=ABQ_Cure_types.inp
**MATERIAL*, NAME="ABQ_CURE_MATERIAL"
**EIGENSTRAIN*, USER

Reaction Kinetics

The degree of cure (or conversion) of a material is characterized by a normalized quantity, α, with a value changing from 0 (corresponding to the uncured state; that is, no bonds) to 1 (corresponding to the fully cured material).

The cure reaction kinetics control the rate of conversion, $\dot{\alpha}$, as a function of α and temperature. The Kamal equation provides a well accepted description of the cure reaction kinetics that is known to produce accurate results, particularly for epoxy resins. It is given by the following rate form:

$$\dot{\alpha} = \sum_{i=1}^{N} Z_i e^{-\frac{E_i}{R\left(\theta-\theta^Z\right)}} \left(b_i + \alpha^{m_i}\right)\left(\alpha_{\max} - \alpha\right)^{n_i},$$

Equation 1

where N is the number of terms, Z_i are rate constants, E_i are activation energies, m_i and n_i are reaction constants, θ^Z is the absolute zero on the temperature scale used, and R is the universal gas constant. The material constants, b_i, are introduced to allow a nonzero initial conversion rate by setting at least one of them to a small positive value. Alternatively, you can specify a nonzero initial conversion value, $\alpha|_{t=0}$, using initial conditions (see *Solution-Dependent State Variables*). If you do not define b_i or $\alpha|_{t=0}$, the degree of cure (conversion) remains equal to zero throughout the analysis.

The constant, $\alpha_{\max}$, controls the maximum degree of cure. By default, $\alpha_{\max} = 1$. You can define a different value, as described in *Maximum Conversion*.

In addition to the Kamal equation, you can specify the rate of conversion, $\dot{\alpha}$, in a tabular format. This format allows you to express the rate as a function of conversion and, optionally, temperature and field variables:

$$\dot{\alpha} = f(\alpha, \theta, FVs).$$

Curing is an irreversible process; therefore, the value of the degree of cure that Abaqus computes never decreases. The value either increases or remains constant.

Input File Usage: Use the following options to define the conversion rate in the Kamal equation:

**PHYSICAL CONSTANTS*, UNIVERSAL GAS CONSTANT=*R*, ABSOLUTE ZERO= ^{Z}
**MATERIAL*
**PARAMETER TABLE*, TYPE=*"ABQ_Cure_ReactionKinetics_Kamal"*

Z_1, E_1, m_1, n_1, b_1

Z_2, E_2, m_2, n_2, b_2

Z_N, E_N, m_N, n_N, b_N

Use the following options to define the conversion rate in tabular form:

*MATERIAL
*PROPERTY TABLE, TYPE="ABQ_Cure_ReactionKinetics_Tabular", TEMPERATURE
$\dot{\alpha}$, α, θ

Maximum Conversion

In general, the degree of cure can reach a maximum value of 1 (corresponding to a fully cured material). However, at lower temperatures, the reaction might slow down considerably, and a fully cured state might not be reached (corresponding to $\alpha_{max} < 1$). You can specify the maximum conversion as a function of temperature. The default is $\alpha_{max} = 1$.

Input File Usage: Use the following option to specify the maximum conversion:

*MATERIAL
*PROPERTY TABLE, TYPE="ABQ_Cure_ReactionKinetics_MaxConversion"
α_{max}

Cure Heat Generation

Curing reactions are irreversible, exothermic processes that are activated by mixing or heating. The amount of heat released per unit volume per unit time is given by the relationship:

$$q = \rho_0 Q \dot{\alpha},$$

where ρ_0 is the density, Q is the specific heat of the reaction, and $\dot{\alpha} = \frac{d\alpha}{dt}$ is the conversion rate.

Input File Usage: Use the following options to define the volumetric heat generation rate:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_Cure_HeatGeneration"
Q, ρ_0

Cure Shrinkage Strain

During the curing process, the material undergoes permanent shrinkage. The shrinkage is due to cross-linking because the formation of bonds moves the atoms closer together than in the unbonded state. The shrinkage and thermal strains that develop during the curing process result in residual stresses that might cause warpage of the final product. Predicting residual stress distributions is often one of the main reasons for performing numerical simulations. You can model the thermal strain using Abaqus capabilities (see *Thermal Expansion*). The cure shrinkage strain is expressed in the following general rate form:

$$\dot{\boldsymbol{\varepsilon}}^{s} = -\boldsymbol{\gamma}\dot{\alpha},$$

where $\boldsymbol{\varepsilon}^{s}$ is the cure shrinkage strain, $\boldsymbol{\gamma}$ is the shrinkage coefficient matrix, and $\dot{\alpha}$ is the conversion rate. Abaqus supports four forms to specify cure shrinkage coefficients: volumetric, isotropic, orthotropic, and anisotropic. You can use the orthotropic and anisotropic forms only with materials where the material directions are defined with local orientations (see *Orientations*).

The cure shrinkage strains enter the constitutive model in the form of an eigenstrain. Similar to thermal strains, the cure shrinkage strains are subtracted from the total deformation to compute the mechanical stress response.

Volumetric Cure Shrinkage Strain

The volumetric cure shrinkage strain is computed from the following rate equation:

$$\dot{\boldsymbol{\varepsilon}}^{s} = -\tfrac{1}{3}\gamma_{vol}\mathbf{I}\dot{\alpha}\left(t\right).$$

In this case you only need to specify one value, γ_{vol}, as a function of temperature and field variables.

Input File Usage: Use the following options to specify the volumetric shrinkage coefficient:

**MATERIAL*
**PROPERTY TABLE*, TYPE=*"ABQ_Cure_ShrinkageCoeff_Vol"*

γ_{vol}

Isotropic Cure Shrinkage Strain

The isotropic cure shrinkage strain is computed from the following rate equation:

$$\dot{\boldsymbol{\varepsilon}}^{s} = -\gamma\mathbf{I}\dot{\alpha}(t).$$

In this case you only need to specify one value, γ, as a function of temperature and field variables.

Input File Usage: Use the following options to specify the isotropic shrinkage coefficient:

**MATERIAL*
**PROPERTY TABLE*, TYPE=*"ABQ_Cure_ShrinkageCoeff_Iso"*

γ

Orthotropic Cure Shrinkage Strain

The orthotropic cure shrinkage strain is computed from the following rate equation:

$$\begin{pmatrix} \dot{\varepsilon}^{s}_{11} & 0 & 0 \\ 0 & \dot{\varepsilon}^{s}_{22} & 0 \\ 0 & 0 & \dot{\varepsilon}^{s}_{33} \end{pmatrix} = -\begin{pmatrix} \gamma_{11} & 0 & 0 \\ 0 & \gamma_{22} & 0 \\ 0 & 0 & \gamma_{33} \end{pmatrix}\dot{\alpha}\left(t\right).$$

In this case you specify the coefficients in the principal material directions as functions of temperature and field variables.

Input File Usage: Use the following options to specify the orthotropic shrinkage coefficients:

**MATERIAL*
**PROPERTY TABLE*, TYPE=*"ABQ_Cure_ShrinkageCoeff_Ortho"*

γ_{11}, γ_{22}, γ_{33}

Anisotropic Cure Shrinkage Strain

The anisotropic cure shrinkage strain is computed from the following rate equation:

$$\begin{pmatrix} \dot{\varepsilon}^{s}_{11} & \dot{\varepsilon}^{s}_{12} & \dot{\varepsilon}^{s}_{13} \\ \dot{\varepsilon}^{s}_{12} & \dot{\varepsilon}^{s}_{22} & \dot{\varepsilon}^{s}_{23} \\ \dot{\varepsilon}^{s}_{13} & \dot{\varepsilon}^{s}_{23} & \dot{\varepsilon}^{s}_{33} \end{pmatrix} = - \begin{pmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{12} & \gamma_{22} & \gamma_{23} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} \end{pmatrix} \dot{\alpha}(t) .$$

In this case you must specify all six components of the coefficient, $\boldsymbol{\gamma}$, as functions of temperature and field variables.

Input File Usage: Use the following options to specify the anisotropic shrinkage coefficients:

**MATERIAL*
**PROPERTY TABLE*, TYPE=*"ABQ_Cure_ShrinkageCoeff_Aniso"*

$\gamma_{11}, \gamma_{22}, \gamma_{33}, \gamma_{12}, \gamma_{13}, \gamma_{23}$

Mechanical Response of the Material

The mechanical response of the material in the uncured state is typically viscoelastic and, in the fully cured state, is often viscoelastic. Therefore, the cure modeling capabilities are typically used in combination with the small-strain viscoelastic modeling capabilities already available in Abaqus.

Cure-Dependent Material Properties

The elastic and viscoelastic material properties can change considerably as the cross-linking progresses and the material transitions from the uncured to the cured state. In general, these material properties depend on the degree of cure. To obtain accurate results, you must account for this dependency. You can consider the dependency of material properties on the degree of cure in the model by associating the value of a field variable with the degree of cure and specifying the material properties as a function of this field variable (see *Specifying Material Data as Functions of Solution-Dependent Variables*). *Hedegaard et al. (2021)* describe a testing procedure for measuring viscoelastic properties as a function of temperature and conversion level.

Solution-Dependent State Variables

The cure modeling capability is available as a special-purpose material model that makes use of solution-dependent state variables (SDVs) for multiple purposes, such as defining state variables, initial values of state variables, and output (see *Solution-Dependent State Variables*). You must allocate at least three solution-dependent state variables. The meaning of the different SDVs is described in the table below.

SDV	Label	Description
1	ALPHA	Degree of cure or conversion. The value is between 0 and 1. The value is equal to 0 if the material is uncured and is equal to 1 if the material is fully cured.
2	ALPHAR	Conversion rate, $\dot{\boldsymbol{\alpha}}$.
3	DALPHARDT	Derivative of the conversion rate with respect to temperature, $\frac{\partial \dot{\alpha}}{\partial \theta}$.

Optionally, you can specify a nonzero initial value of the degree of cure to trigger the start of the curing reaction (see *Equation 1*).

Input File Usage: Use the following option to allocate space for solution-dependent state variables:

*DEPVAR
3
1, ALPHA, “Degree of cure”
2, ALPHAR, “Degree of cure rate”
3, DALPHARDT, “Derivative of the rate of degree of cure with respect to temperature”

Use the following option to define initial conditions for the degree of cure:

INITIAL CONDITIONS, TYPE=SOLUTION

Example: Defining the Cure Modeling Capabilities

This example illustrates defining the cure modeling capabilities in combination with a typical viscoelastic material definition. The material properties are defined as a function of the degree of cure using field variable dependency. Field variable 1 is used for the purpose of the example.

```
*HEADING
*INCLUDE, INPUT=ABQ_Cure_types.inp
*MATERIAL, NAME="ABQ_CURE_MATERIAL_matName"
*DENSITY
Data lines to specify mass density
*ELASTIC, DEPENDENCIES=1
Data lines to specify linear elastic parameters
*VISCOELASTIC
Data lines to specify viscoelastic parameters
*TRS, DEFINITION=TABULAR, DEPENDENCIES=1
Data line to specify logarithm of the shift function
*EXPANSION
Data line to specify thermal expansion coefficients
*EIGENSTRAIN, USER
*PARAMETER TABLE, TYPE="ABQ_Cure_ReactionKinetics_Kamal"
Data line to specify the Kamal model parameters
*PROPERTY TABLE, TYPE="ABQ_Cure_ShrinkageCoeff_Iso"
Data line to specify the isotropic shrinkage coefficients
*CONDUCTIVITY
Data line to specify the thermal conductivity
*SPECIFIC HEAT
Data lines to specify the specific heat
*HEAT GENERATION
*PARAMETER TABLE, TYPE="ABQ_Cure_HeatGeneration"
Data lines to define the volumetric heat generation rate
*USER DEFINED FIELD, TYPE=SPECIFIED
1, SDV1
*DEPVAR
3
1, ALPHA, "Degree of cure"
2, ALPHAR, "Degree of cure rate"
3, DALPHARDT, "Derivative of the rate of degree of cure with respect to
temperature"
*STEP
*COUPLED TEMPERATURE-DISPLACEMENT
Data line to control incrementation and to specify the total time
*END STEP
```

Elements

The cure model supports three-dimensional, plane strain, and axisymmetric continuum elements that have both displacements and temperatures as nodal variables (see *Choosing the Appropriate Element for an Analysis Type*).

Procedures

The cure model is supported only in fully coupled thermal-stress analyses (see *Fully Coupled Thermal-Stress Analysis*).

Output

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers*), the following variables have special meaning for the cure material model:

SDV1
Degree of cure (conversion), α.

SDV2
Rate of degree of cure, $\dot{\alpha}$.

SDV3
Derivative of the rate of degree of cure with respect to temperature, $\frac{\delta\dot{\alpha}}{\delta\theta}$.

EEIG
Cure shrinkage strain, ε^{s}.

References

- Hedegaard, A., E. Breedlove, S. Carpenter, V. Jusuf, C. Li, and D. Lindeman, Time-Temperature-Cure Superposition (TTCS) Methods for Determining Viscoelasticity of Structural Adhesives During Curing 44th Meeting of the Adhesion Society, 2021.
- Li, C., Y. Wang, and J. Mason, "The Effects of Curing History on Residual Stresses in Bone Cement During Hip Arthroplasty," *Journal of Biomedical Materials Research*, vol. 70B, pp. 30–36, 2004.
- Lindeman, D., S. Carpenter, and C. Li, Residual Stress Development During Curing of Structural Adhesives: Experimental Characterization and Modeling 44th Meeting of the Adhesion Society, 2021.

Modeling Metallurgical Phase Transformations

Products: Abaqus/Standard

References:

- *About Table Collections, Parameter Tables, and Property Tables*
- *Metallurgical phase transformations*

Overview

A special-purpose built-in user-defined field option is available for modeling metallurgical phase transformations during additive manufacturing processes. You can define material properties as functions of field variables associated with the metallurgical state. This functionality is also applicable to more traditional heat treatment processes.

Abaqus provides a general metallurgical phase transformation framework applicable to arbitrary metal alloys. The framework accounts for material physical state changes from a stock feed (that is, a raw material, such as metal powder) via melting and solidification followed by metallurgical solid-state phase transformations induced by rapid heating or cooling events. These events are associated with typical three-dimensional printing sequences but also with slow-rate temperature evolutions associated with heat treatment applications. Grain morphology and grain size assessment models can also be included.

The metallurgical phase transformation framework is available as a special-purpose user-defined field option that relies on built-in user fields to track phase changes and the corresponding variation of material properties. To activate the framework, the material definition must include the phase transformation material user type. It must also include parameter and property tables with a declared type that starts with `"ABQ_PHASE_TRANS"`. In addition, you must specify user-defined field variables and allocate at least $3 + n_s + 3n_t$ solution-dependent state variables, where n_s and n_t are the number of all possible solid phases and solid-phase transformations associated with the material, respectively. Abaqus uses an "internal" implementation of user subroutine *USDFLD*.

A dedicated collection of parameter and property tables is available to include all the definitions required to model physical transitions and metallurgical phase transformations or to assess grain morphology and grain size. You can use the **abaqus fetch** utility to obtain the file containing the type definitions of the parameter and property tables used by the metallurgical phase transformation framework as follows:

```
abaqus fetch job=ABQ_phase_trans_types.inp
```

Input File Usage: Use the following options to activate the metallurgical phase transformation framework:

**INCLUDE*, INPUT=ABQ_phase_trans_types.inp
**MATERIAL*, NAME=*materialName*, USER TYPE=PHASE_TRANSFORMATION
…
**PARAMETER TABLE*, TYPE="ABQ_PHASE_TRANS_MeltingTemperature"
…
**USER DEFINED FIELD*
**DEPVAR*
$3 + n_s + 3n_t$

Material Physical Raw-Liquid-Solid (RLS) State

You can specify a temperature range to model physical state changes. During an additive manufacturing process, the raw (unfused) material is melted to a liquid state by an intense moving heat source, and then undergoes cooling and solidification when the heat source is removed. The solidified (fused) material can be remelted when the heat source passes by a nearby location. As shown in *Figure 1*, the transitions between raw, liquid, and solid states of a material alloy are considered to be governed by temperature and occur within a temperature range bounded by the solidus and liquidus temperature of the alloy.

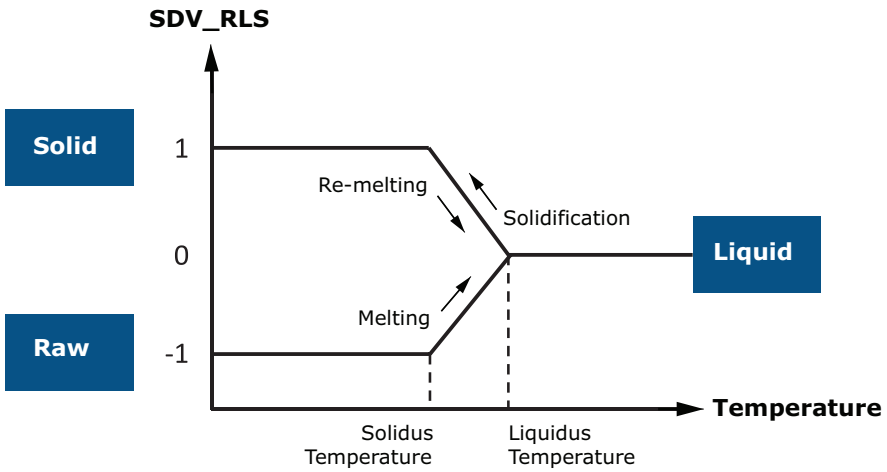


Figure 1: Diagram of material physical state changes between raw-liquid-solid states.

Input File Usage: Use the following options to specify the solidus and liquidus temperature for modeling material physical raw-liquid-solid state transition:

**MATERIAL*
**PARAMETER TABLE*, TYPE=*"ABQ_PHASE_TRANS_MeltingTemperature"*

$\theta_{solidus}$, $\theta_{liquidus}$

Solid-Phase Transformation

The rapid heating and cooling events associated with additive manufacturing processes and slower heat treatment applications can induce metallurgical solid-phase transformations. You can define all possible solid phases and the set of rules that govern their transformations.

A solid-phase transformation can be written in the form

$$p_1 + p_2 + \cdots + p_{n_p} \rightarrow c_1 + c_2 + \cdots + c_{n_c},$$

where n_p and n_c are the number of parent and child phases, respectively; and p_i and c_j are the i^{th} parent and the j^{th} child phases, respectively.

The change of volume fraction of a parent phase during a transformation is assumed to be proportional to its current volume fraction. A similar assumption is made for child phases. With these assumptions, a transformation can be simplified as

$$p \rightarrow c,$$

consisting of one main parent phase and one main child phase with volume fractions of $f_p = \sum_{i=1}^{n_p} f_{p_i}$ and $f_c = \sum_{j=1}^{n_c} f_{c_j}$, respectively.

A transformation usually occurs in a specific thermal condition. You can specify the conditions in terms of temperature rate and/or temperature range for a transformation. If both rate and range are specified, the transformation occurs when both conditions are satisfied.

Two types of transformation mechanisms are available: diffusional or martensitic (nondiffusional).

The following steps are required to define solid-phase transformations completely:

1. Define the information for all solid phases using a parameter table of type *"ABQ_PHASE_TRANS_SolidPhases"*.
2. Define the information for all solid-phase transformations using a parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.
3. Define the kinetics of each transformation by either providing time-temperature-transformation (TTT) diagrams or by specifying model coefficients directly using dedicated parameter and property tables.

Input File Usage: Use the following options to define all possible solid phases:

**MATERIAL*
**PARAMETER TABLE*, TYPE=*"ABQ_PHASE_TRANS_SolidPhases"*
solid phase name, initial fraction upon solidification
…

Repeat the data line as often as necessary to define a list of solid phases.

Use the following options to define solid-phase transformations:

**MATERIAL*
**PARAMETER TABLE*, TYPE=*"ABQ_PHASE_TRANS_Transformations"*
transformation name and definition of parent-child relations, conditions, etc.
…

Repeat the data line as often as necessary to define a list of solid-phase transformations.

Diffusional Transformation

A diffusional transformation is a phase change that involves the long-range diffusion of atoms as a function of both time and temperature. The kinetics of diffusional transformations under isothermal conditions (constant temperature θ) can be described by the Johnson-Mehl-Avrami (JMA) model

$$f_p = f_{tot}\left(1 - \left(1 - \exp[-kt^n]\right)\left(1 - f_{p,0}^{eq}\right)\right),$$

where $f_{tot} = f_p + f_c$ is the total volume fraction of all parent and child phases participating in the transformation; k and n are the reaction rate constant and Avrami exponent, respectively; and $f_{p,0}^{eq}$ is the equilibrium relative total volume fraction of the parent phases. The coefficients k and n can be temperature dependent, and they can be calibrated from a time-temperature-transformation (TTT) diagram. *Figure 2* shows a representative TTT diagram illustrating a diffusional transformation (D) and a martensitic transformation (M). A transformation is usually represented by two curves in the TTT diagram: a start curve where the transformation begins (for example, D_s, M_s in *Figure 2*) and a finish curve where the transformation completes (for example, D_f, M_f in *Figure 2*). A curve in a TTT diagram represents the time (as a function of temperature) that a transformation has completed

a certain proportion. For example, the start curve of the diffusional transformation, D_s, can be represented by the time, $t_s^D(\theta)$, when the transformation begins and the corresponding relative total volume fraction of child phases, f_s^D. Therefore, the JMA coefficients of a diffusional transformation can be calibrated from the TTT curves as

$$n(\theta) = \frac{\ln\left(\ln\left(1-f_s^D\right)/\ln\left(1-f_f^D\right)\right)}{\ln\left(t_s^D(\theta)/t_f^D(\theta)\right)},$$

$$k(\theta) = -\frac{\ln\left(1-f_s^D\right)}{\left(t_s^D(\theta)\right)^{n(\theta)}}.$$

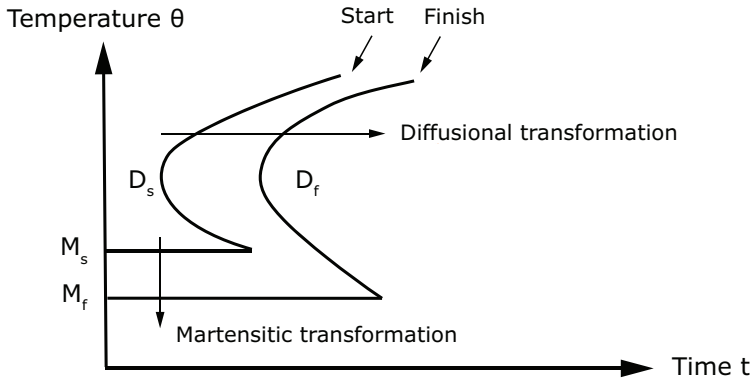


Figure 2: A representative time-temperature-transformation diagram.

Additive manufacturing processes involve continuous temperature changes in rapid heating and cooling events. The kinetics of a diffusional transformation under general non-isothermal conditions is modeled by an extended Johnson-Mehl-Avrami (JMA) model that considers each time increment is a small isothermal step,

$$f_p(\theta_t + \Delta\theta) = f_{tot}\left(1 - \left(1 - \exp[-k(\xi + \Delta t)^n]\right)\left(1 - f_{p,0}^{eq}\right)\right), \text{ if } f_p > f_{tot} f_{p,0}^{eq}$$

where Δt is the time increment size; θ_t is the temperature at the end of the last increment; $\Delta\theta$ is the temperature change for the current time increment; and ξ is the fictitious time that would provide an equivalent amount of transformed material, $f_p(\theta_t)$, at a constant temperature $\theta + \Delta\theta$:

$$\xi = \left[-\frac{1}{k}\ln\left(\frac{f_p(\theta_t) - f_{tot} f_{p,0}^{eq}}{f_{tot}\left(1-f_{p,0}^{eq}\right)}\right)\right]^{\frac{1}{n}}.$$

If the diffusional transformation is reversible ($p \leftrightarrow c$), the reverse transformation $c \rightarrow p$ can be described as

$$f_p(\theta_t + \Delta\theta) = f_{tot}\left(1 - \exp[-k(\xi + \Delta t)^n]\right) f_{p,0}^{eq}, \text{ if } f_p \leq f_{tot} f_{p,0}^{eq}.$$

You can define the kinetics of the transformation using either a TTT diagram or calibrated JMA model coefficients.

Input File Usage: Use the following options to define the kinetics of a diffusional transformation using a TTT diagram:

**MATERIAL*
**PARAMETER TABLE*, TYPE=*"ABQ_PHASE_TRANS_Diffusional_TTT_Constants"*,

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_TTT_Constants"

f_s, f_f
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Diffusional_TTT_Diagram",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_TTT_Diagram",
TEMPERATURE

$t_s(\theta)$, $t_f(\theta)$, θ
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_ParentEquiFrac",
TEMPERATURE

$f_{p,0}^{eq}(\theta)$, θ

Use the following options to define the kinetics of the diffusional transformation using JMA model coefficients:

*MATERIAL
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_JMA_Coefficients",
TEMPERATURE

$k(\theta)$, $n(\theta)$, θ
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_ParentEquiFrac",
TEMPERATURE

$f_{p,0}^{eq}(\theta)$, θ

NameOfTransformation is the name of the transformation defined in the parameter table of type *ABQ_PHASE_TRANS_Transformations*.

Martensitic Transformation

A martensitic transformation solely depends on temperature, as described by the Koistinen-Marburger (KM) model,

$$f_c(\theta) = f_c(\theta_0) + \left(f_p(\theta_0) - f_{pr}\right)\left(1 - \exp[-\gamma(M_s - \theta)]\right),$$

where θ_0 is the temperature at the beginning of the current cooling cycle, and the martensitic start temperature M_s and the transformation coefficient γ can be calibrated from TTT diagrams. Since a martensitic transformation is independent of time, this type of transformation is usually represented by at least two horizontal lines in the TTT diagram, as shown in *Figure 2*. For example, the start curve of the martensitic transformation can be represented by the temperature M_s where the transformation begins and the corresponding volume fraction of the children phases f_s^M. Thus, the transformation coefficient γ can be computed as

$$\gamma = -\frac{\ln\left(\left(1-f_s^M\right)/\left(1-f_f^M\right)\right)}{M_f - M_s}.$$

Abaqus uses a backward Euler integration of the differential form of the original KM model,

$$f_c\left(\theta+\Delta\theta\right)=\frac{f_c(\theta)-\gamma\Delta\theta\left(f_p(\theta_0)-f_{pr}+f_c(\theta_0)\right)}{1-\gamma\Delta\theta}.$$

The transformation occurs if the initial volume fraction of parent phases is higher than the retained amount of parent phases; namely, $f_p(\theta_0) > f_{pr}$.

You can define the kinetics of the transformation using either a TTT diagram or calibrated KM model coefficients.

Input File Usage: Use the following options to define the kinetics of the martensitic transformation using a TTT diagram:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_PHASE_TRANS_Martensitic_TTT_Constants",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_TTT_Constants"

f_s, f_f
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Martensitic_TTT_Diagram",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_TTT_Diagram"

M_s, M_f
*PROPERTY TABLE,
TYPE="ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac",
LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_ParentRetainedFrac"

f_{pr}, $f_p(\theta_0)$

Use the following options to define the kinetics of the martensitic transformation using KM model coefficients:

*MATERIAL
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_Martensitic_KM_Coefficients",

LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_KM_Coefficients"

γ
*PROPERTY TABLE,
TYPE="ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac",
LABEL="ABQ_PHASE_TRANS_Trans_*NameOfTransformation*_ParentRetainedFrac"

f_{pr}, $f_p(\theta_0)$

NameOfTransformation is the name of the transformation defined in the parameter table of type "ABQ_PHASE_TRANS_Transformations".

Grain Morphology

You can specify a solidification map for modeling grain morphology. As shown in Figure 3, given the magnitude of thermal gradient $G=\left|\frac{\partial\theta}{\partial x}\right|$ and solidification rate $R=\left|\dot{\theta}\right|/G$, the grain morphology can be determined to be equiaxed, columnar, or a mixture of the two morphologies.

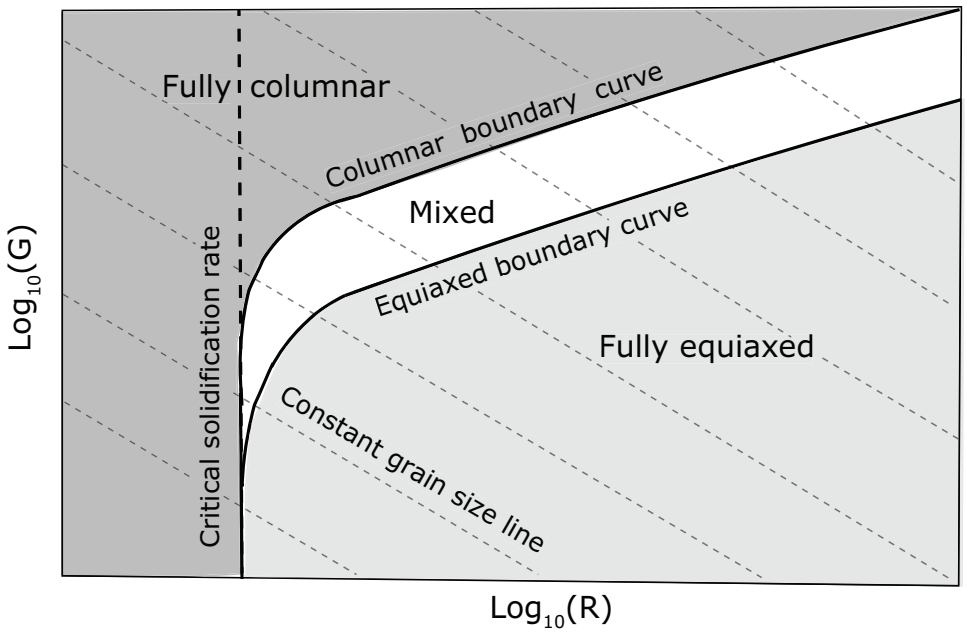


Figure 3: Solidification map.

Input File Usage: Use the following options to define a solidification map to model the evolution of the grain morphology:

**MATERIAL*
**PARAMETER TABLE*,
TYPE=*"ABQ_PHASE_TRANS_SolidificationMap_Constants"*

critical solidification rate R_{cr}, slope of the constant grain size line k_{d0}
**PROPERTY TABLE*, TYPE=*"ABQ_PHASE_TRANS_SolidificationMap_Diagram"*,

LABEL="ABQ_PHASE_TRANS_SolidificationMap_Diagram_Equiaxed"

$\log_{10}(G_{equiaxed})$, $\log_{10}(R)$
**PROPERTY TABLE*, TYPE=*"ABQ_PHASE_TRANS_SolidificationMap_Diagram"*,

LABEL="ABQ_PHASE_TRANS_SolidificationMap_Diagram_Columnar"

$\log_{10}(G_{columnar})$, $\log_{10}(R)$

Grain Size

The evolution of the grain size can be estimated using a parabolic growth law in the form:

$$d^n - d_0^n = kt \exp\left(-\frac{Q}{R\left(\theta-\theta^Z\right)}\right),$$

where d is the grain size at time t; d_0 is the initial grain size on solidification; k, n, and Q are material grain-growth properties; R is the universal gas constant; and θ^Z is the absolute zero.

Input File Usage: Use the following options to define the constants and coefficients of the grain-growth law:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_PHASE_TRANS_GrainSize_Definitions"
d_0, grain growth temperature condition, θ_1, θ_2
*PROPERTY TABLE, TYPE="ABQ_PHASE_TRANS_GrainSize_Coefficients"
k, n, Q

Use the following option to define physical constants used in the grain-growth law:

*PHYSICAL CONSTANTS, ABSOLUTE ZERO=θ^Z, UNIVERSAL GAS CONSTANT=R

Specifying Material Properties as Functions of Metallurgical State

You can specify material properties as functions of metallurgical state. For example, thermal conductivity and specific heat capacity can be different for raw and fused materials. This is achieved by introducing a series of field variables that characterize the metallurgical state and by defining the material properties as a function of those field variables (see *Specifying Material Data as Functions of Temperature and Independent Field Variables*). The metallurgical state includes material physical state, volume fraction ratio of solid phases, volume fraction of columnar grain, and grain size. The volume fraction ratio of solid phase i ($i > 1$) is defined as

$$r_i = \frac{f_i}{\sum_{k=1}^{i} f_k}.$$

This definition ensures that the $n_s - 1$ volume fraction ratios are independent and that their values range from 0 to 1. For example, if a material has four solid phases and it is found that Young's modulus, E, and Poisson's ratio, ν, are different for different solid phases, the material data can be defined as

E	ν	r_2	r_3	r_4
E_1	ν_1	0	0	0
E_2	ν_2	1	0	0
E_3	ν_3	0	1	0
E_4	ν_4	0	0	1

where E_i and ν_i are Young's modulus and Poisson's ratio of solid phase i, respectively.

A dedicated parameter table of type *"ABQ_PHASE_TRANS_FieldVariables"* is available to assign field variables to metallurgical state quantities.

Input File Usage: Use the following options to assign metallurgical state quantities to field variables used in the definition of material properties:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_PHASE_TRANS_FieldVariables"
field variable index, metallurgical state quantitiy

Repeat the data line as often as necessary to assign the metallurgical state to field variables.

Solution-Dependent State Variables

The metallurgical phase transformation framework is available as a special-purpose modeling technique that makes use of solution-dependent state variables (SDVs) (see *About User Subroutines and Utilities*) for multiple purposes, such as defining initial values, output, and workspaces used for computation. You must allocate at least $3 + n_s + 3n_t$ solution-dependent state variables, where n_s and n_t are the number of sets of data defined in the parameter tables of type *"ABQ_PHASE_TRANS_SolidPhases"* and *"ABQ_PHASE_TRANS_Transformations"*, respectively. You must ensure that those spaces are properly allocated and not overwritten by other user subroutines. The meaning of the different SDVs is described below.

SDV	Label	Description
1	RLS	Raw/liquid/solid state. This variable is varying between –1.0 to 1.0. The state is –1.0 if the material is raw, 0.0 if the material is liquid, and 1.0 if the material is solid. The state is between –1.0 and 0.0 if the material is a mixture of raw and liquid. The state is between 0.0 to 1.0 if the material is a mixture of liquid and solid.
2	FGRAINCOLUMNAR	Volume fraction of columnar grain. The variable is 0.0 if the grain is fully equiaxed, 1.0 if the grain is fully columnar, and between 0.0 and 1.0 if the grain is a mixture of equiaxed grain and columnar grain. The variable is –1.0 when the grain morphology is unavailable when the material is liquid.
3	GRAINSIZE	Grain size.
4	FPHASE_1	Volume fraction of solid phase 1.
5	FPHASE_2	Volume fraction of solid phase 2.
...	...	...
$3 + n_s$	FPHASE_N	Volume fraction of solid phase n_s.

You can output SDV1 to SDV($3 + n_s$). The ordering of volume fraction of solid phases starting from SDV4 follows the ordering defined in the parameter table of type *"ABQ_PHASE_TRANS_SolidPhases"*. You can customize a more meaningful output label of SDVs based on the name of the solid phases. For example, if phase 1 is named "alpha", you can use FPHASE_ALPHA as the output label instead of FPHASE_1.

SDV($3 + n_s + 1$) to SDV($3 + n_s + 3n_t$) are workspaces used for solid-phase transformations.

Input File Usage: Use the following option to allocate space for solution-dependent state variables:

**DEPVAR*

$3 + n_s + 3n_t$
1, RLS, “raw/liquid/solid state”
2, FGRAINCOLUMNAR, “volume fraction of columnar grain”
3, GRAINSIZE, “grain size”
4, FPHASE_1, “volume fraction of solid phase 1”
5, FPHASE_2, “volume fraction of solid phase 2”
…
$3 + n_s$, FPHASE_N, “volume fraction of solid phase n_s”

Initial Conditions

You can specify initial values of material physical state, volume fraction of solid phases, volume fraction of columnar grain, and grain size by defining initial values of the corresponding solution-dependent state variables. For example, you can specify the initial value –1 for SDV1 to represent the initial raw material. See *Defining the Initial Values of Solution-Dependent State Variables* for more details.

References

- Crespo, A., "Modelling of Heat Transfer and Phase Transformations in the Rapid Manufacturing of Titanium Components," *Convection and Conduction Heat Transfer, IntechOpen*, 2011.
- Kobryn, P. A., , and S. L. Semiatin, "Microstructure and Texture Evolution During Solidification Processing of Ti–6Al–4V," *Journal of Materials Processing Technology*, vol. 135, no. 2-3, pp. 330–339, 2003.
- Zhang, Q., J. Xie, Z. Gao, T. London, D. Griffiths, and V. Oancea, "A Metallurgical Phase Transformation Framework Applied to SLM Additive Manufacturing Processes," *Materials & Design*, vol. 166, 2019.

Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)

Products: Abaqus/Standard Abaqus/Explicit

References:

- *About Table Collections, Parameter Tables, and Property Tables*
- *Constitutive Artificial Neural Networks (CANNs) models*

Overview

A special-purpose built-in material is available for modeling soft matter using constitutive artificial neural networks (CANNs). You can use this material model to simulate the response of biological tissue and other soft material systems that can be described with an anisotropic hyperelastic strain energy potential.

Abaqus provides a general framework to integrate trained constitutive artificial neural networks (CANNs) models into an Abaqus simulation with minimal user effort. CANNs are a general class of anisotropic hyperelastic constitutive models that are described by a neural network architecture. The architecture used in the AbaqusCANNs framework is based on the work of *Linka et al. (2023)* and *Peirlinck et al. (2025)*.

To activate the framework, the material definition must include the CANNs material user type. It must also include parameter tables with a declared type that starts with "`ABQ_CANN`." In addition, you must specify a user-defined anisotropic hyperelastic material definition, with the number of fiber families, N_f, based on the specific model at hand.

A dedicated collection of parameter tables is available to include all the definitions required to model CANNs models. You can use the **abaqus fetch** utility to obtain the file containing the type definitions of the parameter tables used by the CANNs framework as follows:

```
abaqus fetch job=ABQ_CANN_types.inp
```

Input File Usage:

Use the following options to activate the compressible CANNs framework in Abaqus/Standard:

**INCLUDE*, INPUT=ABQ_CANN_types.inp
**MATERIAL*, NAME=`materialName`, USER TYPE=CANN
**ANISOTROPIC HYPERELASTIC*, DEFINITION=USER,
FORMULATION=INVARIANT,
LOCAL DIRECTIONS=`Nf`, TYPE=COMPRESSIBLE
**PARAMETER TABLE*, TYPE=*"ABQ_CANN_Coeff"*
…
**PARAMETER TABLE*, TYPE=*"ABQ_CANN_DerivedInvariant"*
…

Use the following options to activate the incompressible CANNs framework in Abaqus/Standard:

**INCLUDE*, INPUT=ABQ_CANN_types.inp
**MATERIAL*, NAME=`materialName`, USER TYPE=CANN
**ANISOTROPIC HYPERELASTIC*, DEFINITION=USER,
FORMULATION=INVARIANT,
LOCAL DIRECTIONS=`Nf`, TYPE=INCOMPRESSIBLE
**PARAMETER TABLE*, TYPE=*"ABQ_CANN_Coeff"*

```
…
*PARAMETER TABLE, TYPE="ABQ_CANN_DerivedInvariant"
…
```

Use the following options to activate the CANNs framework in Abaqus/Explicit:

```
*INCLUDE, INPUT=ABQ_CANN_types.inp
*MATERIAL, NAME=materialName,
 USER TYPE=CANN
*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=INVARIANT,
LOCAL DIRECTIONS=Nf
*PARAMETER TABLE, TYPE="ABQ_CANN_Coeff"
…
*PARAMETER TABLE, TYPE="ABQ_CANN_DerivedInvariant"
…
```

CANNs Model Architecture

The models available in the CANNs framework are described by the neural network architecture shown in *Figure 1*. This feed-forward architecture hardwires physical constraints into the network design using deformation invariants as inputs and specific activation functions in the hidden layers. The output of the architecture is the desired strain energy potential. Compared to classical neural network architectures, CANNs are typically not fully connected by design to satisfy the condition of polyconvexity before you run the analysis (see *Linka et al. (2023)*).

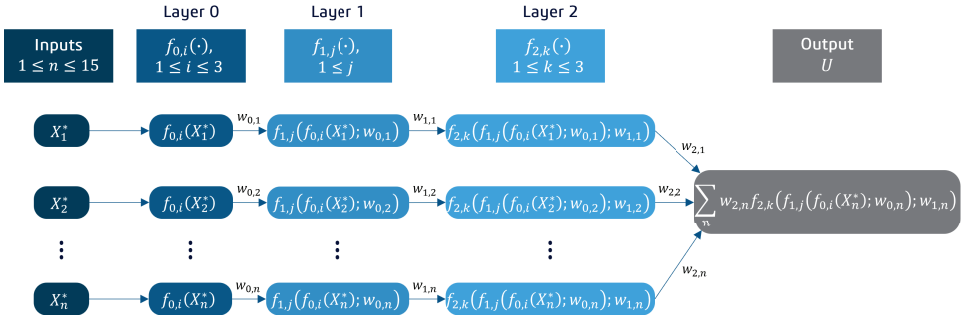


Figure 1: Architecture of a constitutive artificial neural networks model.

CANNs Model Definition

The subsections below detail the invariant inputs and activation functions used to define the CANNs material models.

Strain Energy Potential Output

The model defines a strain energy function, U, as the sum of individual polyconvex functions of the inputs

$$U = \sum_n U_n \left(X_n^*\right),$$

where each contribution U_n takes the form

$$U_n \left(X_n^*\right) = f_2 \circ f_1 \circ f_0 \left(X_n^*\right) = w_{2,n} f_{2,k} \left(f_{1,j} \left(f_{0,i}(X_n^*); w_{0,n}\right); w_{1,n}\right).$$

In the above equation, X_n^* is an invariant input; the integers i, j, k are indices specifying the activation functions $f_{0,i}$, $f_{1,j}$, $f_{2,k}$ in layers 0, 1, and 2, respectively; and $w_{0,n}$, $w_{1,n}$, $w_{2,n}$ are the weights associated with these functions for input X_n^*.

Invariant Inputs

The invariant inputs X_n^* are "shifted" invariants defined by

$$X_n^* = I_n^* - I_{n0}^*$$

where I_n^* are isotropic ($n = 1, \ldots, 3$) and anisotropic ($n = 4, \ldots, 15$) invariants associated with the state of deformation of the material.

The number of anisotropic invariants depends on the total number of material fiber directions; up to $N_f = 3$ fiber directions can be considered. The invariants follow the enumeration convention in Abaqus. The quantity I_{n0}^* represents the value of the corresponding invariant I_n^* in the reference configuration. It follows that $X_n^* = 0$ in the undeformed state. *Table 1* lists the invariant inputs.

Table 1: Inputs and corresponding invariants.

Index, n	I_n^*	I_{n0}^*	X_n^*
1	$\bar{I}_1$	3	$\bar{I}_1 - 3$
2	$\bar{I}_2$	3	$\bar{I}_2 - 3$
3	J^{el}	1	$J^{el} - 1$
4	$\bar{I}_{4(11)}$	1	$\bar{I}_{4(11)} - 1$
5	$\bar{I}_{5(11)}$	1	$\bar{I}_{5(11)} - 1$
6	$\bar{I}_{4(12)}$	ζ_{12}	$\bar{I}_{4(12)} - \zeta_{12}$
7	$\bar{I}_{5(12)}$	ζ_{12}	$\bar{I}_{5(12)} - \zeta_{12}$
8	$\bar{I}_{4(22)}$	1	$\bar{I}_{4(22)} - 1$
9	$\bar{I}_{5(22)}$	1	$\bar{I}_{5(22)} - 1$
10	$\bar{I}_{4(13)}$	ζ_{13}	$\bar{I}_{4(13)} - \zeta_{13}$
11	$\bar{I}_{5(13)}$	ζ_{13}	$\bar{I}_{5(13)} - \zeta_{13}$
12	$\bar{I}_{4(23)}$	ζ_{23}	$\bar{I}_{4(23)} - \zeta_{23}$
13	$\bar{I}_{5(23)}$	ζ_{23}	$\bar{I}_{5(23)} - \zeta_{23}$

Index, n	I_n^*	I_{n0}^*	X_n^*
14	$\overline{I}_{4(33)}$	1	$\overline{I}_{4(33)} - 1$
15	$\overline{I}_{5(33)}$	1	$\overline{I}_{5(33)} - 1$

In *Table 1*,

$\overline{I}_1$ and $\overline{I}_2$	are first and second deviatoric strain invariants,
J^{el}	is the elastic volume ratio (or third strain invariant),
$\overline{I}_{4(\alpha\beta)}$ and $\overline{I}_{5(\alpha\beta)}$	are *pseudo-invariants* of the distortional part of the right Cauchy-Green strain $\overline{\mathbf{C}}$, and
$\zeta_{\alpha\beta}$	are geometric constants (independent of deformation) equal to the cosine of the angle between the directions of any two families of fibers in the reference configuration.

For additional details, see *Anisotropic Hyperelastic Behavior* and *Thermal Expansion*.

Activation Functions

The activation functions in the three hidden layers of this CANNs architecture are the functions $f_{0,i}$, $f_{1,j}$, and $f_{2,k}$, in layers 0, 1, and 2, respectively (see *Figure 1*). For each input, X_n^* (and, therefore, index n), the integers i, j, k are indices specifying these activation functions, and $w_{0,n}$, $w_{1,n}$, $w_{2,n}$ are the weights associated with these functions. That is, $w_{0,n}$ is the weight of layer 0 in layer 1, $w_{1,n}$ is the weight of layer 1 in layer 2, and $w_{2,n}$ is the weight of layer 2 in the definition of U_n.

Table 2 defines the activation functions as follows:

- Layer 0 contains the identity $(\cdot)$, Macaulay bracket $\langle\cdot\rangle$, and absolute value $|\cdot|$ operators. The Macaulay bracket is defined as $\langle x\rangle = \frac{1}{2}(x + |x|)$.
- Layer 1 contains integer exponentiation functions of powers $1, 2, \ldots, p$.
- Layer 2 contains linear scaling, exponential and logarithmic functions. The definitions of activation functions $f_{2,2}$ and $f_{2,3}$ include a scaling factor, $\frac{1}{w_1}$, that makes them different from those in the work of *Linka et al. (2023)*.

Table 2: Activation functions.

Index i, j, k	Layer 0, $f_{0,i}$	Layer 1, $f_{1,j}$	Layer 2, $f_{2,k}$
1	$f_{0,1}(x) = x$	$f_{1,1}(x; w_0) = w_0 x$	$f_{2,1}(x; w_1) = w_1 x$
2	$f_{0,2}(x) = \langle x\rangle$	$f_{1,2}(x; w_0) = (w_0 x)^2$	$f_{2,2}(x; w_1) = \frac{1}{w_1}[\exp(w_1 x) - 1]$
3	$f_{0,3}(x) = \lvert x\rvert$	$f_{1,3}(x; w_0) = (w_0 x)^3$	$f_{2,3}(x; w_1) = \frac{-1}{w_1}\ln\left(1 - w_1 x\right)$

Index i, j, k	Layer 0, $f_{0,i}$	Layer 1, $f_{1,j}$	Layer 2, $f_{2,k}$
$\vdots$		$\vdots$	
p		$f_{1,p}(x; w_0) = (w_0 x)^p$	

Strain Energy Potential Definition

It follows from the above definitions that each contribution U_n to the CANNs strain energy potential U is uniquely defined by the tuple

$$(n, i, j, k, w_{0,n}, w_{1,n}, w_{2,n}),$$

where

n is the index specifying the input X^*_n,

i is the index specifying the activation function $f_{0,i}$ in layer 0,

j is the index specifying the activation function $f_{1,j}$ in layer 1,

k is the index specifying the activation function $f_{2,k}$ in layer 2,

$w_{0,n}$ is the weight of layer 0 in layer 1,

$w_{1,n}$ is the weight of layer 1 in layer 2, and

$w_{2,n}$ is the weight of layer 2 in the definition of U_n.

Abaqus/Explicit requires the definition of some compressibility. Therefore, the parameter table of type *ABQ_CANN_Coeff* must include a contribution U_3 from the third invariant.

Input File Usage: Use the following options to define the strain energy potential, U:

```
*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_CANN_Coeff"
n, i, j, k, w0,n, w1,n, w2,n
...
```

Repeat the data line as often as required to define the list of all the contributions U_n.

Derived Invariant Inputs

Some forms of strain energy potentials, such as the Holzapfel-Gasser-Ogden form, are expressed in terms of linear combinations of the 15 invariants in *Table 1*. Those combinations are referred to as “derived invariants” and provide a level of interconnectivity between the 15 invariants in layer 0 of the CANNs model. Derived invariant Y^*_m is defined by

$$Y^*_m = \sum_{l=1}^{15} a_{ml} X^*_l$$

and can be used as an input in the CANNs architecture using the invariant input $X^*_{100+m} = Y^*_m$ (see *Table 3*).

Table 3: Input enumeration for derived invariants.

Index $100+m$	X^*_{100+m}	Y^*_m
101	X^*_{101}	Y^*_1
102	X^*_{102}	Y^*_2
103	X^*_{103}	Y^*_3
104	X^*_{104}	Y^*_4
⋮	⋮	⋮

Input File Usage:

Use the following options to define the derived invariants Y^*_m:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_CANN_DerivedInvariant"
m, a_{m1}, a_{m2}, a_{m3}, a_{m4}, a_{m5}, a_{m6}, a_{m7}, a_{m8}, a_{m9}, a_{m10}, a_{m11}, a_{m12}, a_{m13}, a_{m14}, a_{m15}
...

Repeat the data line as often as required to define the list of all the derived invariants Y^*_m.

Examples

The examples below illustrate how to use CANNs models for the case of well-known forms of hyperelastic strain energy potentials.

Neo-Hooke Form

The Neo-Hooke strain energy potential is given by

$$U = C_{10}\left(\bar{I}_1 - 3\right) + \frac{1}{D_1}\left(J^{el} - 1\right)^2,$$

which can be rewritten as

$$U = C_{10}X^*_1 + \frac{1}{D_1}(X^*_3)^2$$

in terms of the CANNs input invariants using *Table 1*.

Each of these contributions can be expressed in its tuple definition $(n, i, j, k, w_{0,n}, w_{1,n}, w_{2,n})$:

- The contribution of the first invariant X^*_1 ($n = 1$) can be expressed as $(1, 1, 1, 1, 1.0, 1.0, C_{10})$. This corresponds to using the linear function in each layer ($i = j = k = 1$) with weights $w_{0,1}$ and $w_{1,1}$ set to 1.0 and weight $w_{2,1}$ set to C_{10}.

- The contribution of the third invariant X_3^* ($n = 3$) can be expressed as $\left(3, 1, 2, 1, 1.0, 1.0, \frac{1}{D_1}\right)$. This corresponds to using the linear function in layers 0 and 2 ($i = k = 1$) and the quadratic function in layer 1 ($j = 2$). Weights $w_{0,3}$ and $w_{1,3}$ set to 1.0 and weight $w_{2,3}$ set to $\frac{1}{D_1}$.

Input File Usage: Use the following options to define the Neo-Hooke strain energy potential as a CANNs model:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_CANN_Coeff"
1, 1, 1, 1, 1.0, 1.0, C_{10}
3, 1, 2, 1, 1.0, 1.0, $\frac{1}{D_1}$

Holzapfel-Gasser-Ogden Form (with Fiber Dispersion κ)

The Holzapfel-Gasser-Ogden strain energy potential with fiber dispersion, κ, is given by

$$U = C_{10}\left(\overline{I}_1 - 3\right) + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right) + \frac{k_1}{2k_2}\left\{\exp\left[k_2\left\langle \overline{E}_1\right\rangle^2\right] - 1\right\} + \frac{k_1}{2k_2}\left\{\exp\left[k_2\left\langle \overline{E}_2\right\rangle^2\right] - 1\right\},$$

with

$$\overline{E}_\alpha = \kappa\left(\overline{I}_1 - 3\right) + (1 - 3\kappa)\left(\overline{I}_{4(\alpha\alpha)} - 1\right) \; ; \qquad \alpha = 1, 2.$$

This can be rewritten as

$$U = C_{10}X_1^* + \frac{1}{D}\left(X_3^* + \frac{1}{2}(X_3^*)^2 - \frac{-1}{(-1)}\ln(1 - (-1)X_3^*)\right) + \frac{k_1}{2k_2}\left\{\exp\left[k_2\langle X_{101}^*\rangle^2\right] - 1\right\} + \frac{k_1}{2k_2}\left\{\exp\left[k_2\langle X_{102}^*\rangle^2\right] - 1\right\},$$

in terms of the CANNs input invariants using *Table 1* and of X_{101}^* and X_{102}^*, two derived invariant inputs defined by

$$X_{101}^* = Y_1^* = \kappa X_1^* + (1 - 3\kappa)\, X_4^*$$
$$X_{102}^* = Y_2^* = \kappa X_1^* + (1 - 3\kappa)\, X_8^*$$

representing the effect of fiber dispersion.

Input File Usage: Use the following options to define the Holzapfel-Gasser-Ogden strain energy potential with fiber dispersion, κ, as a CANNs model with derived invariants:

*MATERIAL
*PARAMETER TABLE, TYPE="ABQ_CANN_Coeff"
1, 1, 1, 1, 1.0, 1.0, C_{10}
3, 1, 1, 1, 1.0, 1.0, $\frac{1}{D}$
3, 1, 2, 1, 1.0, 0.5, $\frac{1}{D}$
3, 1, 1, 3, 1.0, −1.0, $-\frac{1}{D}$
101, 2, 2, 2, 1.0, k_2, $\frac{k_1}{2}$
102, 2, 2, 2, 1.0, k_2, $\frac{k_1}{2}$
*PARAMETER TABLE, TYPE="ABQ_CANN_DerivedInvariant"

1, κ, **0.0**, **0.0**, $1-3\kappa$, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**
2, κ, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, $1-3\kappa$, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**, **0.0**

Elements

You can use the CANNs framework with all plane stress, plane strain, and 3D stress-displacement elements in both Abaqus/Standard and Abaqus/Explicit. This set of eligible elements includes 3D solids, 2D axisymmetric, and generalized plane strain elements.

In Abaqus/Standard, you can use hybrid elements to model a fully incompressible CANNs material model. You cannot use compressible anisotropic hyperplastic material behavior (including compressible CANNs models) with hybrid elements. For reduced-integration elements, you must specify the hourglass stiffness because Abaqus does not calculate the default values for CANNs models.

In Abaqus/Standard and Abaqus/Explicit, the transverse shear stiffness definition is required for shell elements since Abaqus does not calculate the default values for CANNs models.

References

- Linka, K., and E. Kuhl, "A New Family of Constitutive Artificial Neural Networks Toward Automated Model Discovery," *Computer Methods in Applied Mechanics and Engineering*, vol. 403 115731, 2023.
- Peirlinck, M., J. A. Hurtado, M. K. Rausch, A. Buganza Tepole, and E. Kuhl, "A Universal Material Model Subroutine for Soft Matter Systems," *Engineering with Computers*, vol. 41, pp. 905–927, 2025.

Modeling Multiscale Materials with Deep Material Networks (DMNs)

Products: Abaqus/Standard

References:

- *About Table Collections, Parameter Tables, and Property Tables*
- *Mean-Field Homogenization*

Overview

The DMN material model:

- uses deep material networks (DMNs);
- can be used to model composites with multiscale modeling;
- can be used to model both linear and nonlinear behavior of fiber-reinforced composites; and
- is available as a special-purpose built-in user-defined material.

Deep material network (DMN) is a data-driven surrogate model for heterogeneous materials (*Liu and Wu, 2019*). Based on linear elastic training data generated by high-fidelity computational homogenization (such as FE-RVE simulation), DMN is capable of predicting their elastic and inelastic behaviors with high accuracy. DMN learns the morphological characterizations of the microstructure by the homogenized effective stiffness tensor $\overline{\boldsymbol{C}}$ and the stiffness of the constituents $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$. The effective stiffness tensor is usually computed using a high-fidelity homogenization method such as the FE-RVE method. The coefficients of DMN are computed by minimizing the difference between the DMN and the FE-RVE predictions, as shown in *Figure 1*.



Figure 1: Training of DMN.

The DMN model is available as built-in user-defined material options. To activate the capability, you must specify a specific material user type label in the material definition:

- Use `"DMN_UD"` if the composite is reinforced with unidirectional fibers.
- Use `"DMN_CHOPPED"` if the composite is reinforced with dispersed short fibers.

  In this case, the dispersion of the fiber is defined with a second-order orientation tensor.

Abaqus internally creates state variables used for this material model. You should not define any solution-dependent state variables because they are reserved for specific output variables.

A dedicated collection of parameter and property tables is available to include all definitions required to use the model. You can use the **abaqus fetch** utility to obtain the file containing the type definitions for the parameter and property tables used by the DMN model:

```
abaqus fetch job=ABQ_DMN_types.inp
```

Input File Usage:

Use the following options to start the definition of a unidirectional fiber-reinforced composite based on built-in user-defined material options:

**INCLUDE*, INPUT="ABQ_DMN_types.inp"
**MATERIAL*, USER TYPE="DMN_UD"
**USER MATERIAL*, CONSTANTS=0

Use the following options to start the definition of a short fiber-reinforced composite based on built-in user-defined material options:

**INCLUDE*, INPUT="ABQ_DMN_types.inp"
**MATERIAL*, USER TYPE="DMN_CHOPPED"
**USER MATERIAL*, CONSTANTS=0

DMN Model Architecture

The DMN model in Abaqus is a rotation-free DMN based on the model proposed in *Gajek and Böhlke (2020)*. Compared to the original rotation-based DMN proposed in *Liu and Wu (2019)*, the rotation-free DMN introduces arbitrary lamination directions as shown in *Figure 2*. The rotation-free DMN also has significantly improved performance compared to the original rotation-based DMN because it avoids rotation of the homogenized stiffness matrices.

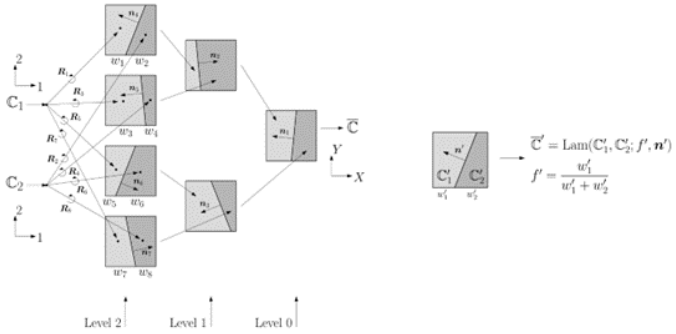


Figure 2: Architecture of a rotation-free DMN.

The rotation-free DMN uses building blocks governed by analytical micromechanics solutions. The network is a binary tree structure, with the bottom layer N containing 2^N nodes. Each node is a two-phase laminate separated by an interface with a normal vector $\mathbf{n}$. The strain and stress relationship between a parent node and its two child nodes follows the rule of mixtures:

$$\overline{\boldsymbol{\varepsilon}} = f_1\boldsymbol{\varepsilon}_1 + f_2\boldsymbol{\varepsilon}_2$$

and

$$\overline{\boldsymbol{\sigma}} = f_1\boldsymbol{\sigma}_1 + f_2\boldsymbol{\sigma}_2,$$

where $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_2$ are the local strains of the two materials, and $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are the local stresses of the two materials. f_1 and f_2 are the volume fractions of materials 1 and 2 ($f_1 + f_2 = 1$). The interface condition is defined as

$$\mathbf{H}^T(\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2) = 0,$$

where

$$\mathbf{H}^T = \begin{bmatrix} n_1 & 0 & 0 & n_2/\sqrt{2} & n_3/\sqrt{2} & 0 \\ 0 & n_2 & 0 & n_1/\sqrt{2} & 0 & n_3/\sqrt{2} \\ 0 & 0 & n_3 & 0 & n_1/\sqrt{2} & n_2/\sqrt{2} \end{bmatrix}$$

corresponding to the Mandel notation of the stress tensor $\boldsymbol{\sigma} = \left(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sqrt{2}\sigma_{12}, \sqrt{2}\sigma_{13}, \sqrt{2}\sigma_{23}\right)$. The relationship between the local strain in each material and the laminate strain is

$$\boldsymbol{\varepsilon}_1 = \overline{\boldsymbol{\varepsilon}} + f_2 \mathbf{H}\mathbf{c}, \boldsymbol{\varepsilon}_2 = \overline{\boldsymbol{\varepsilon}} - f_1 \mathbf{H}\mathbf{c},$$

where

$$\mathbf{c} = \mathbf{M}_1 \overline{\boldsymbol{\varepsilon}}$$

$$\mathbf{M}_1 = \mathbf{M}_2^{-1} \mathbf{H}^T \left(\mathbf{C}_2 - \mathbf{C}_1\right)$$

$$\mathbf{M}_2 = \mathbf{H}^T \left(f_2 \mathbf{C}_1 - f_1 \mathbf{C}_2\right) \mathbf{H}.$$

For elastic materials, the strain-stress relationship of the laminate can be written as:

$$\overline{\boldsymbol{\sigma}} = \overline{\mathbf{C}} \overline{\boldsymbol{\varepsilon}} = f_1 \mathbf{C}_1 \boldsymbol{\varepsilon}_1 + f_2 \mathbf{C}_2 \boldsymbol{\varepsilon}_2,$$

where the effective elastic stiffness matrix is:

$$\overline{\mathbf{C}} = f_1 \mathbf{C}_1 + f_2 \mathbf{C}_2 + f_1 f_2 \left(\mathbf{C}_1 - \mathbf{C}_2\right) \mathbf{H}\mathbf{M}_1.$$

Because DMN is built based on building blocks with analytical homogenization solutions as described above, it inherently satisfies conservation laws and thermodynamic consistency. For the same reason, although the network is trained solely based on linear elastic responses, its results can also be extrapolated to produce accurate nonlinear predictions.

Composites used in industry applications usually have varying volume fraction or aspect ratios from the manufacturing process. The DMN model implemented in Abaqus is fully parametric based on the work by *Li (2024)*. The coefficients of the DMN are trained using a series of FE-RVEs of varying geometric parameters such as the fiber volume fraction as shown in *Figure 3*.

You do not need to specify the coefficients for the parametric DMN. Abaqus internally stores the pretrained coefficients for all the supported microstructures. When you specify the type of the microstructure of the composite with the user type label, Abaqus automatically selects the coefficients and uses them for homogenization. For some microstructures, Abaqus stores coefficients trained with different levels, and you can specify a nondefault value for the number of levels to achieve better accuracy with additional performance cost.

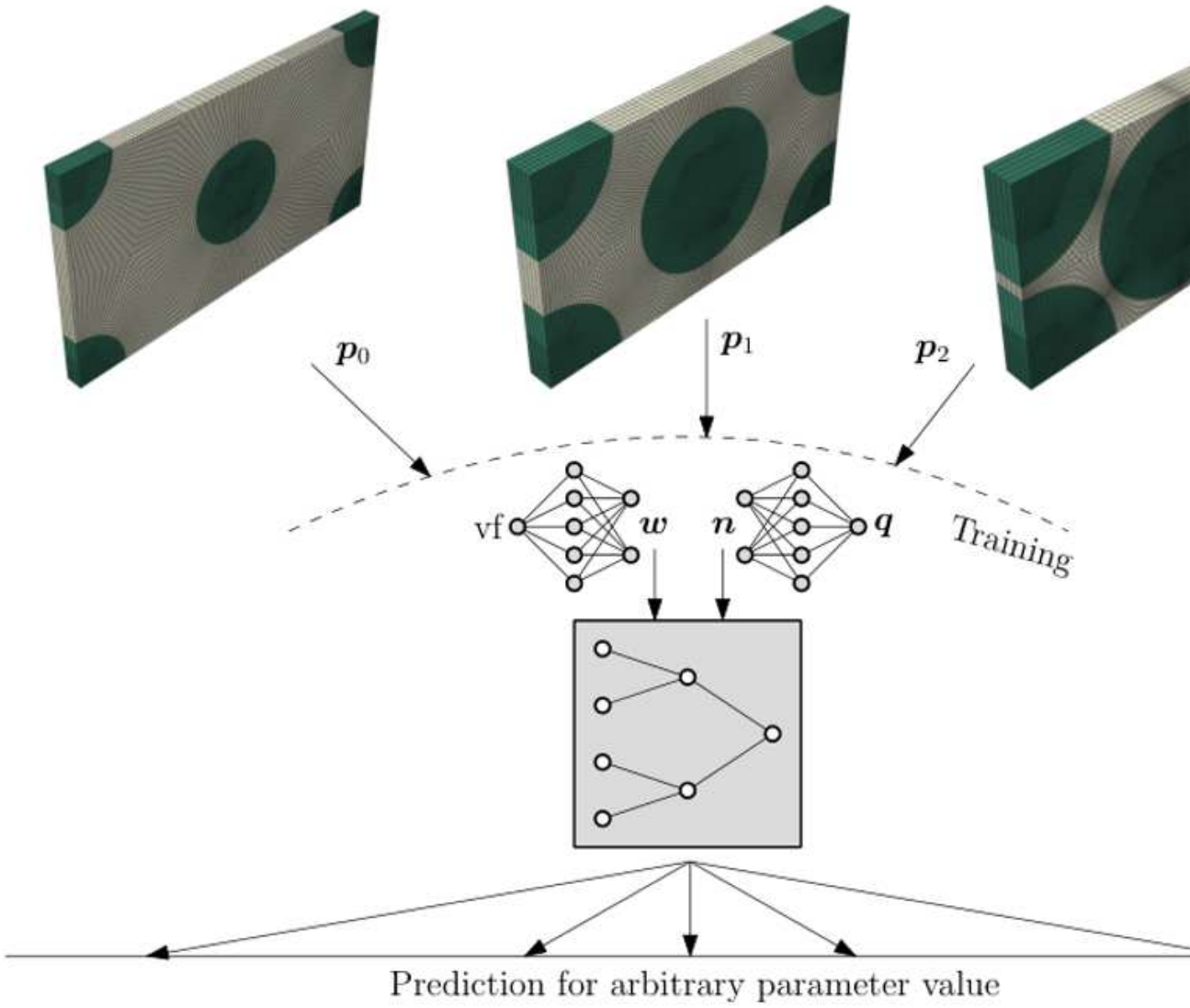


Figure 3: Parametric DMN trained with FE-RVEs of various volume fractions.

Input File Usage: Use the following options to define the total level of the DMN model:

**PARAMETER TABLE*, TYPE="ABQ_DMN_Level"

Microstructure Characteristics

The definition of the DMN model includes the microstructure characteristics and the constituent properties. The microstructure characteristics include the volume fraction, aspect ratio, and the dispersion (described by the orientation tensor) of the fiber. The volume fraction and aspect ratio cannot be dependent on temperature or field variables. You can define the spatial variation of the fiber dispersion by specifying a distribution in the orientation definition (*Defining Fiber Dispersion for Multiscale or Short-Fiber Reinforced Linear Composite Materials*).

You can specify the shape of the fiber in the DMN model when the user type label is DMN_UD.

Input File Usage: Use the following options to define the fiber volume fraction of a unidirectional fiber-reinforced composite:

*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_Volfrac"

Use the following options to define the aspect ratio of the fiber in a short-fiber reinforced composite:

*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_AspectRatio"

Use the following options to define the shape of the fiber:

*PARAMETER TABLE, TYPE="ABQ_MULTSCALE_Shape"

Use the following options to define the orientation tensor of the fiber in a short-fiber reinforced composite:

*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_Oritens"

Constituent Properties

You can define only one matrix and one fiber material in the DMN model in Abaqus. Dedicated property table types are available to define constitutive laws supported for DMN. You must specify labels for these property tables to distinguish between matrix and fiber properties. For the fiber material, only the following properties are supported:

- Isotropic elasticity
- Transversely isotropic elasticity
- Isotropic thermal expansion
- Transversely isotropic thermal expansion

For the matrix material, both linear and nonlinear properties are supported including:

- Isotropic elasticity
- Isotropic thermal expansion
- Linear viscoelasticity defined with Prony series
- Isotropic plasticity with one of the following hardening laws: tabular, power law, or exponential linear law
- Rate-dependent yield including: Power law, Chaboche law, and Prandtl law

Linear Elasticity

The DMN model requires the definition of linear elasticity in both the fiber and the matrix material. You can define linear elasticity in the constituents with dedicated property table types. You can use dedicated labels to indicate whether the elastic response is instantaneous or long-term in the matrix material.

Input File Usage: Use the following option to define the isotropic elasticity in the fiber material:

*PROPERTY TABLE, TYPE"ABQ_MULTSCALE_IsoElas",
LABEL="ABQ_MULTSCALE_Fiber_Elastic

Use the following option to define the transversely isotropic elasticity in the fiber material:

*PROPERTY TABLE, TYPE"ABQ_MULTSCALE_TransIsoElas",
LABEL="ABQ_MULTSCALE_Fiber_Elastic

Use the following option to define the instantaneous isotropic elasticity in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_IsoElas",
LABEL="ABQ_MULTSCALE_Matrix_Elastic

Use the following option to define the long-term isotropic elasticity in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_IsoElas",
LABEL="ABQ_MULTSCALE_Matrix_Elastic_Longterm

Thermal Expansion

You can define thermal expansion in the matrix and fiber material with dedicated property table types.

Input File Usage: Use the following option to define the isotropic thermal expansion in the fiber material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_IsoCTE",
LABEL="ABQ_MULTSCALE_Fiber_CTE

Use the following option to define the transversely isotropic elasticity in the fiber material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_TransIsoCTE",
LABEL="ABQ_MULTSCALE_Fiber_CTE

Use the following option to define the isotropic elasticity in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_IsoCTE",
LABEL="ABQ_MULTSCALE_Matrix_CTE

Linear Viscoelasticity

You can define linear viscoelasticity in the matrix material. The DMN model supports only linear viscoelasticity defined in the time domain with Prony series. You can define the Prony series with dedicated parameter table types.

Input File Usage: Use the following option to define the Prony series in the matrix material:

**PARAMETER TABLE*, TYPE"ABQ_MULTSCALE_Prony",
LABEL="ABQ_MULTSCALE_Matrix_ViscoElastic

Plasticity

You can define isotropic hardening in the matrix material with dedicated property table types. You can specify the hardening with one of the following options:

- define the static yield stress, σ^0 as a tabular function of the equivalent plastic strain, $\bar{\varepsilon}^{pl}$;
- define the static yield stress using a power law, $\sigma^0 = A + B\left(\bar{\varepsilon}^{pl}\right)^n$; or

- define the static yield stress using an exponential linear function, $\sigma^0 = k_0 + k_1 \bar{\varepsilon}^{pl} + k_2 \left(1 - e^{-m\bar{\varepsilon}^{pl}}\right)$.

Input File Usage: Use the following option to define hardening using the tabular function in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_IsoHard", LABEL="ABQ_MULTSCALE_Matrix_Plastic

Use the following option to define the power law hardening in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_PowerHard", LABEL="ABQ_MULTSCALE_Matrix_Plastic

Use the following option to define the exponential linear hardening law in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_ExpLinear", LABEL="ABQ_MULTSCALE_Matrix_Plastic

Rate-Dependent Yield

You can define rate-dependent yield (*Rate-Dependent Yield*) in the matrix material with dedicated property table types. You can define the rate-dependent yield stress $\bar{\sigma}$ as a function of the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^{pl}$, and the static yield stress, σ^0, with one of the following options:

- Power law (*Overstress Power Law*);
- Chaboche rate dependence (*Chaboche Rate Dependence*); or
- Prandtl rate dependence $\dot{\bar{\varepsilon}}^{pl} = \frac{\sigma^0}{\eta}\left(\sinh\left(\frac{\bar{\sigma}}{\beta}\right)\right)^n$, where η, β and n are material parameters.

Input File Usage: Use the following option to define power law rate-dependent yield in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_PowerLaw", LABEL="ABQ_MULTSCALE_Matrix_RateDependent

Use the following option to define the Chaboche rate-dependent yield in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_Chaboche", LABEL="ABQ_MULTSCALE_Matrix_RateDependent

Use the following option to define the Prandtl rate-dependent yield in the matrix material:

**PROPERTY TABLE*, TYPE"ABQ_MULTSCALE_Prandtl", LABEL="ABQ_MULTSCALE_Matrix_RateDependent

Example: Defining a Short Fiber-Reinforced Composite with DMN

This example illustrates defining a short fiber-reinforced composite with DMN.

```
*MATERIAL, USER TYPE=DMN_CHOPPED, NAME=matName
*USER MATERIAL, CONSTANTS=0
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_Volfrac"
Data line to specify the volume fraction of the fiber
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_AspectRatio"
Data line to specify the aspect ratio of the fiber
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_Oritens"
Data line to specify the principal values of the fiber orientation tensor
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_IsoElas",
LABEL="ABQ_MULTSCALE_Matrix_Elastic"
Data line to define the linear elasticity of the matrix material
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_ExpLinear",
LABEL="ABQ_MULTSCALE_Matrix_Plastic"
Data line to define the exponential linear hardening of the matrix material
*PROPERTY TABLE, TYPE="ABQ_MULTSCALE_IsoElas",
LABEL="ABQ_MULTSCALE_Fiber_Elastic"
Data line to define the linear elasticity of the fiber material
```

Elements

The DMN material model can be used only with three-dimensional and plane stress elements that have displacements nodal variables (see *Choosing the Appropriate Element for an Analysis Type*).

Procedures

The DMN material model is supported only in stress analyses.

Output

For DMN material, Abaqus internally create SDVs for specific output.

In addition to the standard output identifiers available in Abaqus (*Abaqus/Standard Output Variable Identifiers*), the following variable has special meaning for the DMN material model:

SDV1
Volume-averaged equivalent plastic strain, $\bar{\varepsilon}^{p}_{avg}$. User label SDV_PEEQ_AVG.

You can also use a dedicated parameter table to request additional micro-level output. When micro-level output is requested, the following variables have special meaning for the DMN material model:

SDV1
11-component of stress in the matrix, S^{m}_{11}.

SDV2
22-component of stress in the matrix, S^{m}_{22}.

SDV3
33-component of stress in the matrix, S^{m}_{33}.

SDV4
12-component of stress in the matrix, S^m_{12}.

SDV5
13-component of stress in the matrix, S^m_{13}.

SDV6
23-component of stress in the matrix, S^m_{23}.

SDV7
11-component of stress in the fiber, S^f_{11}.

SDV8
22-component of stress in the fiber, S^f_{22}.

SDV9
33-component of stress in the fiber, S^f_{33}.

SDV10
12-component of stress in the fiber, S^f_{12}.

SDV11
13-component of stress in the fiber, S^f_{13}.

SDV12
23-component of stress in the fiber, S^f_{23}.

SDV13
Volume-averaged equivalent plastic strain, $\bar{\varepsilon}^p_{avg}$. User label SDV_PEEQ_AVG.

Input File Usage: Use the following option to request micro-level output:

```
*PARAMETER TABLE, TYPE="ABQ_DMN_Controls"
"MicroOutput"
```

References

- Gajek, S., M. Schneider, and T. Böhlke, "On the Micromechanics of Deep Material Networks," *Journal of the Mechanics and Physics of Solids*, vol. 242, 2020.
- Li, T., "Micromechanics-Informed Parametric Deep material network for physics behavior prediction of heterogeneous materials with a Varying Morphology," *Computer Methods in Applied Mechanics and Engineering*, vol. 419, 2024.
- Liu, Z., and C.T. Wu, "Exploring the 3D Architectures of Deep Material Network in Data-Driven Multiscale Mechanics," *Journal of the Mechanics and Physics of Solids*, vol. 127 20-46, 2019.
- Liu, W., C.T. Wu, and M. Koishi, "A Deep Material Network for Multiscale Topology Learning and Accelerated Nonlinear Modeling of Heterogeneous Materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 345 1138-1168, 2019.

Modeling The Electrical and Mechanical Response Of Cardiac Tissue

Special-purpose built-in materials are available for modeling the electrical and mechanical behavior of cardiac tissue.

In this section:

- *Modeling the Electrical Response of Cardiac Tissue*
- *Modeling the Mechanical Response of Cardiac Tissue*

Modeling the Electrical Response of Cardiac Tissue

Products: Abaqus/Standard

References:

- *About Table Collections, Parameter Tables, and Property Tables*
- *Electrical and mechanical modeling of cardiac tissue*

Overview

You can use a special-purpose built-in material to model the electrical behavior of cardiac tissue.
For additional information about how this material model is used in combination with the Living Heart Human model, refer to the *Living Heart Human Model Guide*.

Abaqus/Standard provides a special-purpose framework to model the electrical behavior of cardiac tissue and to account for the coupling between the electric potential and the recovery variable. This capability relies on a built-in implementation of user subroutine *HETVAL* and uses the mathematical analogy between heat and electrical conduction. In this context, thermal quantities are interpreted as their electrical counterparts, and the electrical analysis of the cardiac tissue is run as the heat transfer analysis of an analogous thermally conductive material.

To activate this capability, the definition of the analogous thermally conductive material must include the electric cardiac tissue material user type and parameter tables with a declared type that starts with "`ABQ_CARDIAC_ELEC`." In addition, you must specify the density, thermal conductivity, and specific heat of the analogous material, three state-dependent variables, and heat generation.

A dedicated collection of parameter tables is available to include all the definitions required for this material model. You can use the **abaqus fetch** utility to obtain the file containing the type definitions of the parameter tables used by the built-in material as follows:

```
abaqus fetch job=ABQ_CARDIAC_TISSUE_types.inp
```

Input File Usage: Use the following options to activate the electrical model for cardiac tissue in Abaqus/Standard:

**INCLUDE*, INPUT=ABQ_CARDIAC_TISSUE_types.inp
**MATERIAL*, NAME=`materialName`, USER TYPE=CARDIAC_TISSUE
**DENSITY*
**CONDUCTIVITY*
**SPECIFIC HEAT*
**HEAT GENERATION*
**PARAMETER TABLE*, TYPE=*"ABQ_CARDIAC_ELEC_PARAMETERS_AP"*
…
**DEPVAR*
**PARAMETER TABLE*, TYPE=*"ABQ_CARDIAC_ELEC_ACTIVATION"*
…

Electrical Conduction in Cardiac Tissue

Following the works of *Aliev and Panfilov (1996)* and *Hurtado and Kuhl (2014)*, the electrical conduction equation in cardiac tissue is given by

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \boldsymbol{q}(\phi) = f^{\phi}\left(\phi, r\right)$$

Equation 1

for the electric potential ϕ. The current density $\boldsymbol{q}$ is given by Ohm's law

$$\boldsymbol{q} = -\boldsymbol{D}\nabla\phi$$

Equation 2

in terms of the electrical diffusion tensor $\boldsymbol{D}$.

The source term $f^{\phi}\left(\phi, r\right)$ represents the local action potential profile and is given by

$$f^{\phi} = \frac{\phi_N}{\tau}[c\varphi(\phi)(1-\varphi(\phi))(\varphi(\phi)-\alpha) - r\varphi(\phi)]$$

Equation 3

in terms of

$$\varphi\left(\phi\right) = \frac{\phi-\phi_s}{\phi_N},$$

a normalized electric potential and τ, a time scaling factor. These normalizations are introduced to facilitate the computations when using physiological values for the material parameters. The normalization coefficients are ϕ_s = -80 mV, ϕ_N = 100 mV, and τ = 12.9 ms.

In these equations,

c	is a scaling parameter,
α	is the oscillation threshold, and
r	is the recovery variable.

The recovery variable r is the solution of the evolution equation

$$\dot{r} = f^{r}(\phi, r),$$

Equation 4

where the source term $f^{r}(\phi, r)$ characterizes the slow features of the action potential:

$$f^{r}\left(\phi, r\right) = \frac{1}{\tau}[\gamma + r\overline{\gamma}(\varphi(\phi))]\,[-r - c\varphi(\phi)\,[\varphi(\phi) - b - 1]]$$

with

$$\overline{\gamma}\left(\varphi\right) = \frac{\mu_1}{\mu_2+\varphi},$$

where

γ	is the refractoriness,
b	is a phenomenological scaling parameter, and
μ_1 **and** μ_2	are two parameters that control the shape of the restitution curve.

Analogous Thermal Conduction Formulation

As noted above, the electrical conduction equations (*Equation 1*-*Equation 3*) in cardiac tissue can be recast as the transient heat transfer

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{k} \nabla T) = \dot{Q}$$

Equation 5

for the temperature T in an analogous thermally conductive material with unit density and specific heat, thermal conductivity tensor $\boldsymbol{k}$. *Table 1* shows the correspondence between the electrical quantities pertaining to the cardiac tissue and the thermal quantities relevant to the analogous thermally conductive material.

Table 1: Correspondence between electrical and thermal quantities.

Electrical quantity	Thermal quantity
Electric potential, ϕ	Temperature, T
Electrical diffusion tensor, $\boldsymbol{D}$	Thermal conductivity tensor, $\boldsymbol{k}$
Electrical source term, f^{ϕ}	Heat generation source term, $\dot{Q}$

In addition, the evolution equation (*Equation 4*) for the recovery variable r is recast as

$$\dot{r} = f^{r}(T, r),$$

Equation 6

which is solved in a built-in *HETVAL* subroutine that is used to specify the heat generation source term in *Equation 5*.

The heat equation (*Equation 5*) and evolution equation (*Equation 6*) are solved using the transient heat transfer procedure in Abaqus/Standard.

Input File Usage: Use the following options to define the material parameters of the model:

**MATERIAL*
**DENSITY*
1.0
**CONDUCTIVITY*
$\boldsymbol{k}$
**SPECIFIC HEAT*
1.0
**HEAT GENERATION*
**PARAMETER TABLE*, TYPE=*"ABQ_CARDIAC_ELEC_PARAMETERS_AP"*
α, γ, b, c, μ_1, μ_2

Solution-Dependent State Variables

This special-purpose material model makes use of solution-dependent state variables (SDVs) to specify spatially dependent material parameters and output and to record state variables (see *About User Subroutines and Utilities*). You must specify at least three solution-dependent state variables, as described in *Table 2*.

Table 2: Label and description for each SDV.

SDV	Label	Description
1	GAMMA	The refractoriness γ if it is specified as an initial value for SDV1. Otherwise, SDV1 is equal to 0.
2	ACTTIME	The activation time, t_{act}; that is, the first time when $\phi \geq \phi_{act}$, if an electrical potential activation threshold ϕ_{act} is specified using the parameter table of type *"ABQ_CARDIAC_ELEC_ACTIVATION"*. Otherwise, SDV2 is equal to 0.
3	R	The recovery variable, r.

Input File Usage: Use the following option to allocate space for solution-dependent state variables:

**DEPVAR*
3
1, GAMMA, "Refractoriness"
2, ACTTIME, "Activation_time"
3, R, "Recovery_variable"

Distribution of the Refractoriness

Optionally, you can define the spatial distribution of the refractoriness, γ, by specifying an initial value for SDV1. Positive values specified as initial conditions for SDV1 override the value specified in the parameter table of type *"ABQ_CARDIAC_ELEC_PARAMETERS_AP"*.

Input File Usage: Use the following option to define the spatial distribution of the refractoriness:

**INITIAL CONDITIONS*, TYPE=SOLUTION

Recording the Activation Time

Optionally, you can define an electrical potential activation threshold, ϕ_{act}, to record the activation time, t_{act}, in SDV2; that is, the first time when $\phi \geq \phi_{act}$.

Input File Usage: Use the following option to specify the electrical potential activation threshold:

**PARAMETER TABLE*, TYPE=*"ABQ_CARDIAC_ELEC_ACTIVATION"*
ϕ_{act}

Example

This example illustrates defining the electrical analysis of cardiac tissue. A spatial distribution of the refractoriness and a definition of the electrical potential activation threshold are included for the purpose of the example.

```
*HEADING
*INCLUDE, INPUT=ABQ_CARDIAC_TISSUE_types.inp
*MATERIAL, NAME=materialName, USER TYPE=CARDIAC_TISSUE
*DENSITY
1.0
*CONDUCTIVITY
Data line to specify the thermal conductivity equivalent to the electrical
 diffusion
*SPECIFIC HEAT
1.0
*HEAT GENERATION
*PARAMETER TABLE, TYPE="ABQ_CARDIAC_ELEC_PARAMETERS_AP"
Data line to define the material parameters of the model
*DEPVAR
3
1, GAMMA, "Refractoriness"
2, ACTTIME, "Activation_time"
3, R, "Recovery_variable"
*PARAMETER TABLE, TYPE="ABQ_CARDIAC_ELEC_ACTIVATION"
Data line to define the electrical potential activation threshold
*INITIAL CONDITIONS, TYPE=SOLUTION
Data line to define a spatial distribution of the refractoriness
*STEP
*HEAT TRANSFER, DELTMX
Data line to control incrementation and to specify the total time
*END STEP
```

Procedures

This model is supported only in uncoupled transient heat transfer analyses (see *Uncoupled Heat Transfer Analysis*). It is recommended to use the DELTMX parameter to allow increment cutback when needed.

Elements

This model supports diffusive heat transfer elements (see *Choosing the Appropriate Element for an Analysis Type*).

Output

Thermal quantities have a special meaning for this material model and must be interpreted as their electrical counterparts. For example,

GRADT	Gradient of the electric potential.
HFL	Current density.
NT11	Electric potential.

In addition to the standard output identifiers available in Abaqus/Standard (*Abaqus/Standard Output Variable Identifiers*), the following variables have a special meaning for this material model:

SDV1
Refractoriness, γ, if specified as an initial value for SDV1. Otherwise, SDV1 is equal to 0.

SDV2
Activation time, t_{act}.

SDV3
Recovery variable, r.

References

- Aliev, R. R., and A. V. Panfilov, "A Simple Two-Variable Model of Cardiac Excitation," *Chaos, Solitons & Fractals*, vol. 7, no. 3, pp. 293–301, 1996.
- Hurtado, D. E., and E. Kuhl, "Computational Modelling of Electrocardiograms: Repolarisation and T-Wave Polarity in the Human Heart," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, pp. 986–996, 2014.

Modeling the Mechanical Response of Cardiac Tissue

Products: Abaqus/Explicit

References:

- *About Table Collections, Parameter Tables, and Property Tables*
- *Electrical and mechanical modeling of cardiac tissue*

Overview

You can use a special purpose built-in material to model the mechanical behavior of cardiac tissue.
For additional information about how this material model is used in combination with the Living Heart Human model, refer to the *Living Heart Human Model Guide*.

Abaqus/Explicit provides a built-in material to model the mechanical behavior of cardiac tissue. This model combines a passive tissue response and an active tissue response. The passive tissue response corresponds to the anisotropic hyperelastic model proposed by *Holzapfel and Ogden (2009)*. The active tissue response represents the mechanical loads generated by the electrical activation of cardiac tissue. The tissue passive and active responses are implemented and combined in a built-in implementation of user subroutine *VUANISOHYPER_INV* .

To activate this capability, the material definition must include the mechanical cardiac tissue material user type. In addition, you must specify a user-defined anisotropic hyperelastic material definition, a density value, and four state-dependent variables.

Input File Usage: Use the following options to activate the mechanical model for cardiac tissue in Abaqus/Explicit:

```
*MATERIAL, NAME=materialName,
 USER TYPE=CARDIAC_TISSUE_HYBRID
*DENSITY
*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,
FORMULATION=INVARIANT,
LOCAL DIRECTIONS=2, PROPERTIES=22
*DEPVAR
```

Formulation

The mechanical passive and active responses of cardiac tissue are described in detail below.

Mechanical Passive Tissue Response

The mechanical passive tissue response is given by the Holzapfel-Ogden anisotropic hyperelastic model (see *Holzapfel-Ogden Form*). The strain energy potential is given by

$$U = \frac{a}{2b}\left\{\exp\left[b\left(\overline{I}_1 - 3\right)\right] - 1\right\} + \sum_{\alpha=f,s} \frac{a_\alpha}{2b_\alpha}\left\{\exp\left[b_\alpha\left(\overline{I}_{4(\alpha\alpha)} - 1\right)^2\right] - 1\right\}$$

$$+ \frac{a_{fs}}{2b_{fs}}\left\{\exp\left[b_{fs}\left(\overline{I}_{4(fs)} - \zeta_{12}\right)^2\right] - 1\right\} + \frac{1}{D}\left(\frac{\left(J^{el}\right)^2 - 1}{2} - \ln J^{el}\right),$$

Equation 1

where U is the strain energy per unit of reference volume; the indexes f and s ($f = 1$, $s = 2$) denote the fiber and sheet directions, which are assumed to be aligned with the local directions 1 and 2, respectively; and a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}, and D are material parameters (see *Holzapfel-Ogden Form* for additional details). Unlike the native implementation of the Holzapfel-Ogden model, the cardiac fibers and sheets contribute to *Equation 1* when they are in compression.

If the material is stretched significantly, the second and third exponential terms in the Holzapfel-Ogden strain energy potential might overflow numerically. To avoid this, the exponential terms can be linearized or approximated by polynomial functions when the stretch in the fiber direction reaches a certain value λ_{fL} (see *Linearization of Holzapfel-Ogden Strain Energy Potential* for additional details). The value of the linearization stretch must be $\lambda_{fL} \geq 1$. If you do not specify the linearization stretch or set it to 0, the exponential terms in the strain energy potential are not linearized.

Mechanical Active Tissue Response

The active tissue response intends to capture the Frank-Starling effect; that is, the fact that the strength of the heart's systolic contraction is directly proportional to its diastolic expansion. To do so, the active response accounts for the local changes of the sarcomere length, which affects the stress in the fiber and sheet directions.

Following the work of *Guccione et al. (1993)* and *Walker et al. (2005)*, a time-varying elastance model defines the active stress in the cardiac fiber direction, σ_{af}, as

$$\sigma_{af}\left(t, E_{ff}\right) = \begin{cases} T_{max} H\left(E_{ff}\right) \frac{1-\cos\left(\omega\left(t, E_{ff}\right)\right)}{2} & \text{when } I_{act}(t) > 0 \text{ and } l(E_{ff}) > l_0 \\ 0 & \text{otherwise} \end{cases}$$

Equation 2

where $I_{act}\left(t\right)$ stands for the activation indicator function defined in *Equation 4* and l is the deformed sarcomere length given by

$$l\left(E_{ff}\right) = l_r \sqrt{2E_{ff} + 1}$$

Equation 3

in terms of the stress-free sarcomere length, l_r, and the component E_{ff} of the Lagrangian strain tensor aligned with the local fiber direction.

The active stress is first activated at time $t_{act}^{(1)}$ when the activation field variable $\mathbf{FV_k}$ first reaches the electrical potential activation threshold, ϕ_{act}:

$$t_{act}^{(1)} = \inf\{t \geq 0 | \mathbf{FV_k}(t) = \phi_{act}\}.$$

The active stress remains activated for a duration d_{act} and subsequent activations occur at $t_{act}^{(m)}$, $m \geq 2$, when the activation field variable, $\mathbf{FV_k}$, again reaches the electrical potential activation threshold, ϕ_{act}, and if the active stress is not currently activated:

$$t_{act}^{(m)} = \inf\left\{t \geq t_{act}^{(m-1)} + d_{act} \middle| \mathbf{FV_k}(t) = \phi_{act}\right\},\ m \geq 2.$$

The activation indicator function in *Equation 2* can then be written as

$$I_{act}\left(t\right) = \begin{cases} t_{act}^{(m)} & \text{when } t_{act}^{(m)} \leq t \leq t_{act}^{(m)} + d_{act} \\ 0 & \text{otherwise} \end{cases}$$

Equation 4

in terms of the activation times $t_{act}^{(m)}$ and the activation duration d_{act}.

The time evolution of the active stress during the activation duration $d_{act} = t_0 + t_r\left(E_{ff}\right)$ is governed by

$$\omega\left(t, E_{ff}\right) = \begin{cases} \pi \frac{t - t_{act}^{(m)}}{t_0} & \text{when } t_{act}^{(m)} \le t \le t_{act}^{(m)} + t_0 \\ \pi\left(1 - \frac{t - t_{act}^{(m)} - t_0}{t_r\left(E_{ff}\right)}\right) & \text{when } t_{act}^{(m)} + t_0 \le t \le t_{act}^{(m)} + d_{act} \\ 0 & \text{otherwise} \end{cases}$$

Equation 5

with t_0 the duration to reach peak tension $T_{max} H\left(E_{ff}\right)$ and $t_r\left(E_{ff}\right)$ a strain-dependent relaxation duration given by

$$t_r\left(E_{ff}\right) = c_1 l\left(E_{ff}\right) + c_2$$

Equation 6

with two model parameters c_1 and c_2.

Finally, the Hill equation that describes chemical equilibrium in *Equation 2* is given by

$$H\left(E_{ff}\right) = \frac{Ca_0^2}{Ca_0^2 + ECa_{50}^2\left(E_{ff}\right)}$$

Equation 7

in terms of Ca_0, the peak intercellular calcium concentration and the strain-dependent calcium sensitivity $ECa_{50}\left(E_{ff}\right)$ given by

$$ECa_{50}\left(E_{ff}\right) = \frac{Ca_{0_{max}}}{\sqrt{e^{B\left(l\left(E_{ff}\right) - l_0\right)} - 1}}$$

Equation 8

with $Ca_{0_{max}}$ the maximum intercellular calcium concentration, and B the parameter that governs the shape of peak isometric tension-sarcomere length relation.

Figure 1 shows a schematic for $\mathbf{FV_k}\left(t\right)$ and $I_{act}\left(t\right)$, and *Figure 2* shows a schematic for $\omega\left(t, E_{ff}\right)$ and $\sigma_{af}\left(t, E_{ff}\right)$.

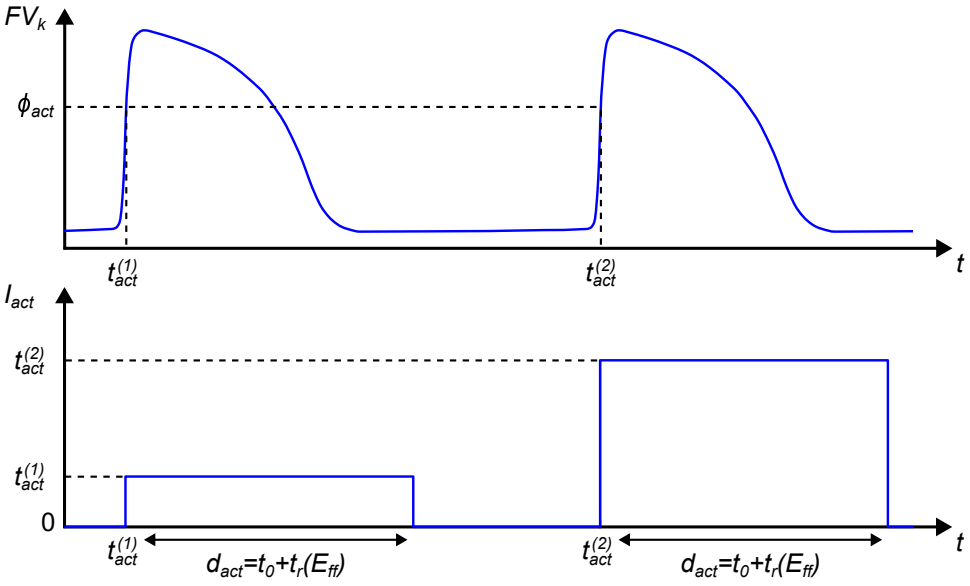


Figure 1: Schematics of the activation field variable and activation indicator function as functions of time.

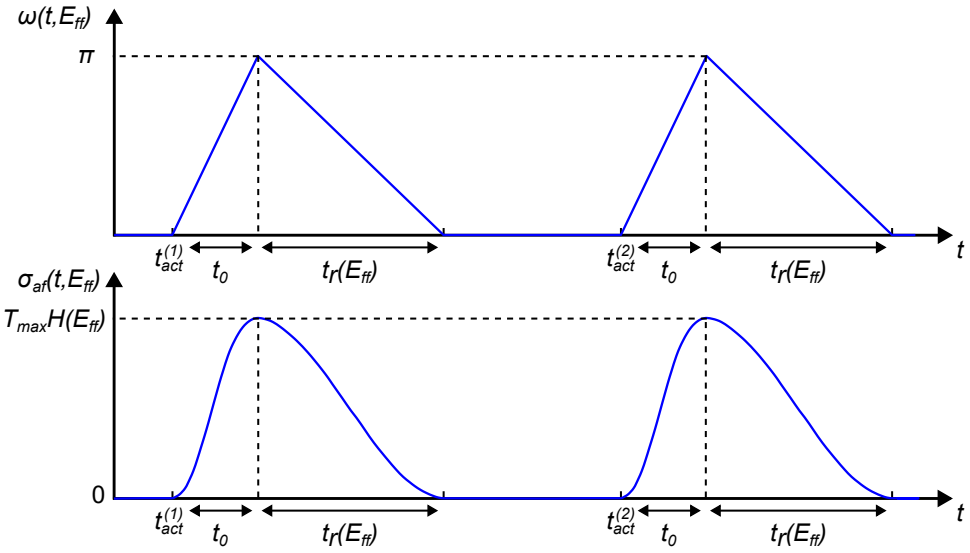


Figure 2: Schematics of $\omega(t, E_{ff})$ and the active stress as functions of time.

Total Tissue Response

The passive and active tissue responses are combined to define the total tissue response. Specifically, the total stress in the fiber direction σ_f is given by

$$\sigma_f = \sigma_{pf} + \sigma_{af}$$

in terms of the passive stress in the fiber direction, σ_{pf}, and the active stress in the fiber direction, σ_{af}. The total stress in the sheet direction, σ_s, is given by

$$\sigma_s = \sigma_{ps} + n\sigma_{af},$$

Equation 9

in terms of the passive stress in the sheet direction, σ_{ps}; the active stress in the fiber direction, σ_{af}; and n, a scalar value less than 1.0 that represents the interaction between the adjacent muscle fibers. The value of n affects the total contractility of the tissue and the degree of twist developed in the tissue during the cardiac cycle.

The 22 parameters that define the combined passive and active tissue responses are defined as follows:

$a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}, D$	are the material parameters for the Holzapfel-Ogden passive response in *Equation 1*,
λ_{fL}	is the linearization stretch for the Holzapfel-Ogden passive response,
$T_{\max}$	is the maximum isometric tension level in *Equation 2*,
ϕ_{act}	is the electrical potential activation threshold in *Equation 2*,
$\mathbf{k}$	is the index for the activation field variable $\mathbf{FV_k}$,
l_0	is the sarcomere length below which there is no activation in *Equation 2*,
l_r	is the stress-free sarcomere length in *Equation 3*,
Ca_0	is the peak intercellular calcium concentration in *Equation 7*,

$Ca_{0_{max}}$ is the maximum intercellular calcium concentration in *Equation 8*,

B is the parameter that governs the shape of peak isometric tension-sarcomere length relation in *Equation 8*,

t_0 is the duration to reach peak tension in *Equation 5*,

c_1 **and** c_2 are parameters for the strain-dependent relaxation duration in *Equation 6*, and

n is the parameter that represents the interaction between the adjacent muscle fibers in *Equation 9*.

Input File Usage: Use the following options to define the material parameters of the model:

**ANISOTROPIC HYPERELASTIC*, DEFINITION=USER,
FORMULATION=INVARIANT,
LOCAL DIRECTIONS=2, PROPERTIES=22

$a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}$

$D, t_0, c_1, c_2, l_0, B, Ca_{0_{max}}, Ca_0$

$T_{\max}, n, \phi_{act}, \lambda_{fL}, l_r, \mathbf{k}$

Damping

Rayleigh mass proportional material damping can be used with this model to simulate the effect of blood mass on cardiac tissue (see *Material Damping* for additional information).

Input File Usage: Use the following option to specify Rayleigh mass proportional material damping:

**DAMPING*, ALPHA=alpha

Viscoelasticity

Isotropic linear viscoelasticity can be used with this capability to model dissipative phenomena (see *Time Domain Viscoelasticity* for additional information).

Input File Usage: Use the following option to specify isotropic linear viscoelasticity:

**VISCOELASTIC*, TIME=PRONY

Predefined Field Variable

This special-purpose material model uses a predefined field variable, $\mathbf{FV_k}$, to trigger the active stress as described above. You must specify the index $\mathbf{k}$ of this activation field variable in the material definition. You can specify the activation field variable directly or you can import it from a previous simulation (see *Predefined Fields* for additional information).

Input File Usage: Use the following option to specify the activation field variable directly:

*FIELD, VARIABLE=k

Use the following option to import the activation field variable from a previous simulation:

*FIELD IMPORT
*EXTERNAL FIELD, FILE=fileName

Solution-Dependent State Variables

The mechanical behavior of cardiac tissue is available as a special-purpose modeling technique that makes use of solution-dependent state variables (SDVs) (see *About User Subroutines and Utilities*) for multiple purposes, such as defining spatially dependent material parameters, state variables, and output. You must specify at least four solution-dependent state variables. *Table 1* provides the meaning of each SDV.

Table 1: Label and description for each SDV.

SDV	Label	Description
1	SIGMA_AF	The active stress in the fiber direction, σ_{af}.
2	STRETCH_F	The stretch in the fiber direction, $\lambda_f = \sqrt{2E_{ff} + 1}$.
3	NOTUSED	This SDV is not currently used by the model.
4	ACTTIME	The indicator function that provides the activation times, $t_{act}^{(m)}$.

Input File Usage: Use the following option to allocate space for solution-dependent state variables:

*DEPVAR
4
1, SIGMA_AF, "Active_fiber_stress"
2, STRETCH_F, "Fiber_stretch"
3, NOTUSED, "NOT_USED"
4, ACTTIME, "Activation_times"

Example

This example illustrates defining the mechanical analysis of cardiac tissue. The activation field variable is imported from a previous simulation for the purpose of the example.

```
*HEADING
*ORIENTATION
Data lines to specify the fiber and sheet orientations
*MATERIAL, NAME=materialName, USER TYPE=CARDIAC_TISSUE_HYBRID
*DENSITY
Data line to specify the density
*ANISOTROPIC HYPERELASTIC, DEFINITION=USER,  FORMULATION=INVARIANT,
LOCAL DIRECTIONS=2, PROPERTIES=22
Data lines to specify the model parameters
*DEPVAR
```

```
4
1, SIGMA_AF, "Active_fiber_stress"
2, STRETCH_F, "Fiber_stretch"
3, NOTUSED, "NOT_USED"
4, ACTTIME, "Activation_times"
*STEP
*DYNAMIC, EXPLICIT
Data line to control incrementation and to specify the total time
*FIELD IMPORT
*EXTERNAL FIELD, FILE=fileName
Data line to import the activation field variable from a previous simulation
*END STEP
```

Procedures

This model is supported only in explicit dynamic analyses (see *Explicit Dynamic Analysis*).

Elements

This model supports all of the elements available in Abaqus/Explicit (see *Choosing the Appropriate Element for an Analysis Type*).

Output

In addition to the standard output identifiers available in Abaqus/Explicit (*Abaqus/Explicit Output Variable Identifiers*), the following variables have a special meaning for this material model:

SDV1

Active stress in the fiber direction, σ_{af}.

SDV2

Fiber stretch, λ_f.

SDV4

Activation time, t_{act}.

References

- Guccione, J. M., L. K. Waldman, and A. D. McCulloch, "Mechanics of Active Contraction in Cardiac Muscle: Part II—Cylindrical Models of the Systolic Left Ventricle," *Journal of Biomechanical Engineering*, vol. 115, pp. 82–80, 1993.
- Holzapfel, G. A., and R. W. Ogden, "A Constitutive Modelling of Passive Myocardium: A Structurally Based Framework for Material Characterization," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, pp. 3445–3475, 2009.
- Walker, J. C., M. B. Ratcliffe, P. Zhang, A. W. Wallace, B. Fata, E. Hsu, D. Saloner, and J. M. Guccione, "MRI-Based Finite-Element Analysis of Left Ventricular Aneurysm," *American Journal of Physiology, Heart and Circulatory Physiology*, vol. 289, pp. H692–H700, 2005.

Parameter Table Type Reference

These parameter table types are used in special-purpose material modeling techniques.

In this section:

- *ABQ_CANN_Coeff*
- *ABQ_CANN_DerivedInvariant*
- *ABQ_CARDIAC_ELEC_ACTIVATION*
- *ABQ_CARDIAC_ELEC_PARAMETERS_AP*
- *ABQ_Cure_HeatGeneration*
- *ABQ_Cure_ReactionKinetics_Kamal*
- *ABQ_DMN_Controls*
- *ABQ_DMN_Level*
- *ABQ_MULTSCALE_Prony*
- *ABQ_MULTSCALE_Shape*
- *ABQ_PHASE_TRANS_Diffusional_TTT_Constants*
- *ABQ_PHASE_TRANS_FieldVariables*
- *ABQ_PHASE_TRANS_GrainSize_Definitions*
- *ABQ_PHASE_TRANS_Martensitic_TTT_Constants*
- *ABQ_PHASE_TRANS_MeltingTemperature*
- *ABQ_PHASE_TRANS_SolidPhases*
- *ABQ_PHASE_TRANS_SolidificationMap_Constants*
- *ABQ_PHASE_TRANS_Transformations*

ABQ_CANN_Coeff

You use this parameter table type to specify a constitutive artificial neural networks (CANNs) model.

You must include a parameter table of type "ABQ_CANN_Coeff" in the material definition to specify a CANNs material model. You must specify seven parameters per line as described below for each contribution to the CANNs strain energy density.

Products: Abaqus/Standard Abaqus/Explicit

- *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_CANN_Coeff", PARAMETERS=7
INTEGER, , "Index n of invariant input X*_n", , ,
INTEGER, , "Index i of activation function in layer 0", , ,
INTEGER, , "Index j of activation function in layer 1", , ,
INTEGER, , "Index k of activation function in layer 2", , ,
FLOAT, , "Weight of layer 0, w_{0,n}", , ,
FLOAT, , "Weight of layer 1, w_{1,n}", , ,
FLOAT, , "Weight of layer 2, w_{2,n}", , ,
```

Parameters

Index of invariant input

Index n of invariant input X^*_n.

Index of activation function in layer 0

Index i of activation function in layer 0.

Index of activation function in layer 1

Index j of activation function in layer 1.

Index of activation function in layer 2

Index k of activation function in layer 2.

Weight of layer 0

Weight $w_{0,n}$ of layer 0.

Weight of layer 1

Weight $w_{1,n}$ of layer 1.

Weight of layer 2

Weight $w_{2,n}$ of layer 2.

ABQ_CANN_DerivedInvariant

You use this parameter table type to specify derived invariant inputs in a constitutive artificial neural networks (CANNs) model.

You can include a parameter table of type `"ABQ_CANN_DerivedInvariant"` in the material definition to specify derived invariant inputs in a CANNs material model. You must specify sixteen parameters per line as described below for each derived invariant.

Products: Abaqus/Standard Abaqus/Explicit

- *Modeling Soft Matter Using Constitutive Artificial Neural Networks (CANNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_CANN_DerivedInvariant", PARAMETERS=16
INTEGER, , "Index m of derived invariant Y*_m", , ,
FLOAT, , "First invariant coefficient, a_m1", , ,
FLOAT, , "Second invariant coefficient, a_m2", , ,
FLOAT, , "Third invariant coefficient, a_m3", , ,
FLOAT, , "Fourth invariant coefficient, a_m4", , ,
FLOAT, , "Fifth invariant coefficient, a_m5", , ,
FLOAT, , "Sixth invariant coefficient, a_m6", , ,
FLOAT, , "Seventh invariant coefficient, a_m7", , ,
FLOAT, , "Eighth invariant coefficient, a_m8", , ,
FLOAT, , "Ninth invariant coefficient, a_m9", , ,
FLOAT, , "Tenth invariant coefficient, a_m10", , ,
FLOAT, , "Eleventh invariant coefficient, a_m11", , ,
FLOAT, , "Twelfth invariant coefficient, a_m12", , ,
FLOAT, , "Thirteenth invariant coefficient, a_m13", , ,
FLOAT, , "Fourteenth invariant coefficient, a_m14", , ,
FLOAT, , "Fifteenth invariant coefficient, a_m15", , ,
```

Parameters

Index of derived invariant
Index m of derived invariant Y^*_m.

First invariant coefficient
Coefficient a_{m1} multiplying X^*_1 in the definition of Y^*_m.

Second invariant coefficient
Coefficient a_{m2} multiplying X^*_2 in the definition of Y^*_m.

Third invariant coefficient
Coefficient a_{m3} multiplying X^*_3 in the definition of Y^*_m.

Fourth invariant coefficient
Coefficient a_{m4} multiplying X^*_4 in the definition of Y^*_m.

Fifth invariant coefficient
Coefficient a_{m5} multiplying X^*_5 in the definition of Y^*_m.

Sixth invariant coefficient
Coefficient a_{m6} multiplying X^*_6 in the definition of Y^*_m.

Seventh invariant coefficient
Coefficient a_{m7} multiplying X^*_7 in the definition of Y^*_m.

Eighth invariant coefficient

Coefficient a_{m8} multiplying X_8^* in the definition of Y_m^*.

Ninth invariant coefficient

Coefficient a_{m9} multiplying X_9^* in the definition of Y_m^*.

Tenth invariant coefficient

Coefficient a_{m10} multiplying X_{10}^* in the definition of Y_m^*.

Eleventh invariant coefficient

Coefficient a_{m11} multiplying X_{11}^* in the definition of Y_m^*.

Twelfth invariant coefficient

Coefficient a_{m12} multiplying X_{12}^* in the definition of Y_m^*.

Thirteenth invariant coefficient

Coefficient a_{m13} multiplying X_{13}^* in the definition of Y_m^*.

Fourteenth invariant coefficient

Coefficient a_{m14} multiplying X_{14}^* in the definition of Y_m^*.

Fifteenth invariant coefficient

Coefficient a_{m15} multiplying X_{15}^* in the definition of Y_m^*.

ABQ_CARDIAC_ELEC_ACTIVATION

You use this parameter table type to specify the electrical potential activation threshold to record the activation time.

You can include a parameter table of type `"ABQ_CARDIAC_ELEC_ACTIVATION"` to specify the electrical potential activation threshold. The activation time is recorded in SDV2.

Products: Abaqus/Standard

- *Modeling the Electrical Response of Cardiac Tissue*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_CARDIAC_ELEC_ACTIVATION", PARAMETERS=1
FLOAT, ,"Electrical potential activation threshold", , ,
```

Parameters

Electrical potential activation threshold

Electrical potential activation threshold, ϕ_{act}.

ABQ_CARDIAC_ELEC_PARAMETERS_AP

You use this parameter table type to specify the electrical parameters of cardiac tissue.

You must include a parameter table of type `"ABQ_CARDIAC_ELEC_PARAMETERS_AP"` in the material definition to specify the electrical parameters of cardiac tissue. You must specify six parameters as described below.

Products: Abaqus/Standard

- *Modeling the Electrical Response of Cardiac Tissue*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_CARDIAC_ELEC_PARAMETERS_AP", PARAMETERS=6
FLOAT, ,"Oscillation threshold", , ,
FLOAT, ,"Refractoriness", , ,
FLOAT, ,"Phenomenological scaling parameter", , ,
FLOAT, ,"Scaling parameter for the action potential profile", , ,
FLOAT, ,"Restitution curve shape parameter 1", , ,
FLOAT, ,"Restitution curve shape parameter 2", , ,
```

Parameters

Oscillation threshold	Oscillation threshold, α.
Refractoriness	Refractoriness, γ.
Phenomenological scaling parameter	Phenomenological scaling parameter, b.
Scaling parameter for the action potential profile	Scaling parameter for the action potential profile, c.
Restitution curve shape parameter 1	Restitution curve shape parameter, μ_1.
Restitution curve shape parameter 2	Restitution curve shape parameter, μ_2.

ABQ_Cure_HeatGeneration

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this parameter table type to specify the specific heat of reaction and the initial density for the cure model.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_Cure_HeatGeneration", PARAMETERS=2
FLOAT, , "Specific Heat of Reaction Q", , ,
FLOAT, , "Density ρ0", , ,
```

Parameters

Specific Heat of Reaction	Specific heat of the reaction, Q.
Density	Initial density of the material, ρ_0.

ABQ_Cure_ReactionKinetics_Kamal

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this parameter table type to specify the cure kinetics in the Kamal form.

You must specify five parameters per line as described below for each term in the Kamal form. In addition, you must specify the absolute zero temperature and universal gas constant for the model (**PHYSICAL CONSTANTS*).

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_Cure_ReactionKinetics_Kamal", PARAMETERS=5
FLOAT, , "Rate Constant Z_i", , ,
FLOAT, , "Activation Energy E_i", , ,
FLOAT, , "Reaction Constant m_i", , ,
FLOAT, , "Reaction Constant n_i", , ,
FLOAT, , "Kamal Constant b_i", , ,
```

Parameters

Rate Constant
Rate constant, Z_i, for the i^{th} term.

Activation Energy
Activation energy, E_i, for the i^{th} term.

Reaction Constant m_i
Reaction constant, m_i, for the i^{th} term.

Reaction Constant n_i
Reaction constant, n_i, for the i^{th} term.

Kamal Constant
Kamal constant, b_i, for the i^{th} term.

ABQ_DMN_Controls

You use this parameter table type to request micro-level output.

Products: Abaqus/Standard

• *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME=ABQ_DMN_Controls, PARAMETERS=1
STRING, , "Control String", ,"MicroOutput",
```

Parameters

Control String	String to activate capabilities specific to DMN. Set to "MicroOutput" to request micro-level output.

ABQ_DMN_Level

You use this parameter table type to specify the total number of levels in the DMN model.

Products: Abaqus/Standard

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_DMN_Level", PARAMETERS=1
INTEGER, , "Number of Levels", , ,
```

Parameters

Number of Levels	Total number of levels in the DMN model.

ABQ_MULTSCALE_Prony

You use this parameter table type to specify the Prony series parameters. The Prony series are used to define the isotropic linear viscoelasticity of the constituent.

Products: Abaqus/Standard

- *Time Domain Viscoelasticity*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_MULTSCALE_Prony", PARAMETERS=3
FLOAT, , "Shear modulus relaxation ratio", , ,
FLOAT, , "Bulk modulus relaxation ratio", , ,
FLOAT, , "Shear relaxation time", , ,
```

Parameters

Shear modulus relaxation ratio	Modulus ratio in the Prony series expansion of the shear relaxation modulus, $\overline{g}_i^P$.
Bulk modulus relaxation ratio	Modulus ratio in the Prony series expansion of the bulk relaxation modulus, $\overline{k}_i^P$.
Shear relaxation time	Relaxation time for the Prony series expansion, τ_i.

ABQ_MULTSCALE_Shape

You use this parameter table type to specify the shape of the fiber in the DMN model.

Products: Abaqus/Standard

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_MULTSCALE_Shape", PARAMETERS=1
STRING, , "Shape", ,"Sphere"|"Prolate",
```

Parameters

Shape Shape of the inclusion in the composite.

Set to `Sphere` if the composite is reinforced with spherical inclusions.

Set to `Prolate` if the composite is reinforced with short fibers.

ABQ_PHASE_TRANS_Diffusional_TTT_Constants

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to specify the constants for the time-temperature-transformation (TTT) diagram for a diffusional transformation.

You must include a parameter table of type `"ABQ_PHASE_TRANS_Diffusional_TTT_Constants"` in the material definition to specify the constants for the time-temperature-transformation (TTT) diagram for a diffusional transformation whose input option is set to `TTT` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The parameter table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*`NameOfTransformation`*`_TTT_Constants"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

The parameter table of type `"ABQ_PHASE_TRANS_Diffusional_TTT_Constants"` can define only one set of data.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_Diffusional_TTT_Constants",
PARAMETERS=2
 FLOAT, , "Starting Proportion", , ,
 FLOAT, , "Ending Proportion", , ,
```

Parameters

Starting Proportion	Relative total volume fraction of child phases at the start of the transformation. This value must be between 0 and 1.
Ending Proportion	Relative total volume fraction of child phases at the end of the transformation. This value must be between 0 and 1.

ABQ_PHASE_TRANS_FieldVariables

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to assign field variables to specify material data as functions of the microstructure state.

You must include a parameter table of type `"ABQ_PHASE_TRANS_FieldVariables"` in the material definition to assign field variables to specify material data as functions of the microstructure state. The parameter table can have multiple sets of data; each set assigns one field variable to a given microstructure characteristic.

Abaqus automatically evaluates the field variable with the corresponding microstructure state before evaluating the material properties.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_FieldVariables", PARAMETERS=2

 INTEGER, , "Field Variable Index", , ,
 STRING, , "Data Option", , RLS|RFPHASEi|GRAINSIZE|FGRAINCOLUMNAR,
```

Parameters

Field Variable Index Field variable number.

Data Option Type of microstructure state to assign to the field variable.

Set to `FGRAINCOLUMNAR` if the field variable represents the volume fraction of columnar grain.

Set to `GRAINSIZE` if the field variable represents the size of the grain.

Set to `RFPHASE`*i* if the field variable represents the volume fraction ratio of solid-phase i ($i > 1$). The volume fraction ratio is defined as $r_i = \frac{f_i}{\sum_{k=1}^{i} f_k}$.

Set to `RLS` if the field variable represents the material physical state.

ABQ_PHASE_TRANS_GrainSize_Definitions

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to define constants of the grain-growth law for modeling grain growth.

You can include a parameter table of type `"ABQ_PHASE_TRANS_GrainSize_Definitions"` in the material definition to define constants of the grain-growth law for modeling grain growth. Only one set of data can be defined.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_GrainSize_Definitions",
PARAMETERS=4
 FLOAT, , "Initial Grain Size", , ,
 STRING, , "Grain-Growth Temperature Condition", , INTERVAL|MIN|MAX|NONE,
 FLOAT, -1E36, "Lower Bound (Grain-Growth Temperature Range)", , ,
 FLOAT, 1E36, "Upper Bound (Grain-Growth Temperature Range)", , ,
```

Parameters

Initial Grain Size

Initial grain size on solidification.

Grain-Growth Temperature Condition

Type of temperature condition that controls grain growth.

Set to `INTERVAL` to specify a range of temperature for grain growth; that is, the grain grows when $\theta_1 \leq \theta \leq \theta_2$.

Set to `MIN` to specify a minimum temperature, θ_1, above which grain can grow; that is, grain grows when $\theta \geq \theta_1$.

Set to `MAX` to specify a maximum temperature, θ_2, below which grain can grow; that is, grain grows when $\theta \leq \theta_2$.

Set to `NONE` if grain grows regardless of the temperature.

Lower Bound (Grain-Growth Temperature Range)

Minimum temperature above which grain can grow, θ_1. Leave blank if the grain-growth temperature condition is set to `MAX` or `NONE`.

Upper Bound (Grain-Growth Temperature Range)

Maximum temperature below which grain can grow, θ_2. Leave blank if the grain-growth temperature condition is set to `MIN` or `NONE`.

ABQ_PHASE_TRANS_Martensitic_TTT_Constants

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to specify constants for the time-temperature-transformation (TTT) diagram for a martensitic transformation.

You can include a parameter table of type `"ABQ_PHASE_TRANS_Martensitic_TTT_Constants"` in the material definition to specify constants for the time-temperature-transformation (TTT) diagram for a martensitic transformation whose input option is set to `TTT` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The parameter table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*NameOfTransformation*`_TTT_Constants"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

The parameter table of type `"ABQ_PHASE_TRANS_Martensitic_TTT_Constants"` can define only one set of data.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_Martensitic_TTT_Constants",
PARAMETERS=2
 FLOAT, , "Starting Proportion", , ,
 FLOAT, , "Ending Proportion", , ,
```

Parameters

Starting Proportion	Relative total volume fraction of parent phases at the start of the transformation. This value must be between 0 and 1.
Ending Proportion	Relative total volume fraction of parent phases at the end of the transformation. This value must be between 0 and 1.

ABQ_PHASE_TRANS_MeltingTemperature

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to specify a melting temperature range for modeling material physical raw-liquid-solid (RLS) state transition.

You must include a parameter table of type `"ABQ_PHASE_TRANS_MeltingTemperature"` in the material definition to specify a melting temperature range for modeling material physical raw-liquid-solid (RLS) state transition. Only one set of data can be defined.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_MeltingTemperature",
PARAMETERS=2
 FLOAT, , "Solidus Temperature", , ,
 FLOAT, , "Liquidus Temperature", , ,
```

Parameters

Solidus Temperature	Temperature below which the material is fully solid.
Liquidus Temperature	Temperature above which the material is fully liquid.

ABQ_PHASE_TRANS_SolidPhases

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to define the different solid phases of the material and their initial volume fraction on solidification.

You must include a parameter table of type `"ABQ_PHASE_TRANS_SolidPhases"` in the material definition to define the different solid phases of the material and their initial volume fraction on solidification. Each table can have multiple sets of data; each set defines one solid phase.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_SolidPhases", PARAMETERS=2
 STRING, , "Solid Phase Name", , ,
 FLOAT, , "Initial Fraction upon Solidification", , ,
```

Parameters

Solid Phase Name	User-defined name of the solid phase.
Initial Fraction upon Solidification	Initial volume fraction on solidification of the solid phase.

ABQ_PHASE_TRANS_SolidificationMap_Constants

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to specify constants of the solidification map for modeling grain morphology.

You must include a parameter table of type `"ABQ_PHASE_TRANS_SolidificationMap_Constants"` in the material definition to specify constants of the solidification map for modeling grain morphology. Only one set of data can be defined.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_SolidificationMap_Constants",
 PARAMETERS=2
 FLOAT, , "Critical Solidification Rate", , ,
 FLOAT, , "Constant Grain Size LogR-LogG Slope", , ,
```

Parameters

Critical Solidification Rate	Solidification rate below which the grain is fully columnar.
Constant Grain Size LogR-LogG Slope	LogR-LogG slope along which the grain size is constant.

ABQ_PHASE_TRANS_Transformations

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this parameter table type to define the solid-phase transformations to consider for a given material.

You must include a parameter table of type "ABQ_PHASE_TRANS_Transformations" in the material definition to define the solid-phase transformations to consider for a given material. The table can have multiple sets of data; each set defines one solid-phase transformation.

Parameter Table Type Definition

```
*PARAMETER TABLE TYPE, NAME="ABQ_PHASE_TRANS_Transformations", PARAMETERS=18

 STRING, , "Solid-Phase Transformation Name", , ,
 STRING, , "Temperature Rate Condition", , INTERVAL|MIN|MAX|NONE,
 FLOAT, -1E36, "Lower Bound (Temperature Rate Range)", , ,
 FLOAT, 1E36, "Upper Bound (Temperature Rate Range)", , ,
 STRING, , "Temperature Condition", , INTERVAL|MIN|MAX|NONE,
 FLOAT, -1E36, "Lower Bound (Temperature Range)", , ,
 FLOAT, 1E36, "Upper Bound (Temperature Range", , ,
 STRING, , "Transformation Type", , DIFFUSIONAL|MARTENSITIC,
 STRING, , "Input Option", , TTT|JMA|KM,
 STRING, , "Reversible Transformation", , NO|YES,
 INTEGER, , "Parent Phases", , ,
 INTEGER, , "Child Phases", , ,
 STRING, " ", "Parent Solid Phase 1 Name", , ,
 STRING, " ", "Parent Solid Phase 2 Name", , ,
 STRING, " ", "Parent Solid Phase 3 Name", , ,
 STRING, " ", "Child Solid Phase 1 Name", , ,
 STRING, " ", "Child Solid Phase 2 Name", , ,
 STRING, " ", "Child Solid Phase 3 Name", , ,
```

Parameters

Solid-Phase Transformation Name

User-defined name of the solid-phase transformation.

Temperature Rate Condition

Type of temperature rate condition that controls the transformation.

Set to INTERVAL to specify a range of temperature rate within which the transformation can occur; that is, the transformation occurs when $\dot{\theta}_1 \leq \dot{\theta} \leq \dot{\theta}_2$.

Set to MIN to specify a minimum temperature rate, $\dot{\theta}_1$, above which the transformation can occur; that is, the transformation occurs when $\dot{\theta} \geq \dot{\theta}_1$.

Set to MAX to specify a maximum temperature rate, $\dot{\theta}_2$, below which the transformation can occur; that is, the transformation occurs when $\dot{\theta} \leq \dot{\theta}_2$.

Set to NONE if the transformation occurs regardless of the temperature rate.

Lower Bound (Temperature Rate Range)
Minimum temperature rate above which the transformation can occur, $\dot{\theta}_1$. Leave blank if the temperature rate condition is set to MAX or NONE.

Upper Bound (Temperature Rate Range)
Maximum temperature rate below which the transformation can occur, $\dot{\theta}_2$. Leave blank if the temperature rate condition is set to MIN or NONE.

Temperature Condition
Type of temperature condition that controls the transformation.

Set to INTERVAL to specify a range of temperatures within which the transformation can occur; that is, the transformation occurs when $\theta_1 \leq \theta \leq \theta_2$.

Set to MIN to specify a minimum temperature, θ_1, above which the transformation can occur; that is, the transformation occurs when $\theta \geq \theta_1$.

Set to MAX to specify a maximum temperature, θ_2, below which the transformation can occur; that is, the transformation occurs when $\theta \leq \theta_2$.

Set to NONE if the transformation occurs regardless of the temperature.

Lower Bound (Temperature Range)
Minimum temperature above which the transformation can occur, θ_1. Leave blank if the temperature condition is set to MAX or NONE.

Upper Bound (Temperature Range)
Maximum temperature below which the transformation can occur, θ_2. Leave blank if the temperature condition is set to MIN or NONE.

Transformation Type
Type of the transformation.

Set to DIFFUSIONAL if the transformation is a diffusional transformation.

Set to MARTENSITIC if the transformation is a martensitic transformation.

Input Option
Defines the option used to define the kinetics of the transformation.

Set to TTT if a time-temperature-transformation (TTT) diagram defines the kinetics of the transformation.

Set to JMA if the transformation is diffusional and the coefficients of the Johnson-Mehl-Avrami (JMA) model define its kinetics.

Set to KM if the transformation is martensitic and the coefficients of the Koistinen-Marburger (KM) model define its kinetics.

Reversible Transformation
Flag indicating whether the transformation is reversible. This parameter is applicable only for diffusional transformations.

Set to NO if the diffusional transformation is irreversible.

Set to YES if the diffusional transformation is reversible.

Leave blank if the transformation is a martensitic transformation.

Parent Phases
Number of parent phases of the transformation, n_p. The value can be 1, 2, or 3.

Child Phases
Number of child phases of the transformation, n_c. The value can be 1, 2, or 3.

Parent Solid Phase 1 Name
User-defined name of the first parent phase, p_1. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*.

Parent Solid Phase 2 Name
User-defined name of the second parent phase, p_2. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*. Leave blank if the number of parent phases is less than 2.

Parent Solid Phase 3 Name
User-defined name of the third parent phase, p_3. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*. Leave blank if the number of child phases is less than 3.

Child Solid Phase 1 Name
User-defined name of the first child phase, c_1. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*.

Child Solid Phase 2 Name
User-defined name of the second child phase, c_2. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*. Leave blank if the number of child phases is less than 2.

Child Solid Phase 2 Name
User-defined name of the third child phase, c_3. The corresponding phase must be defined in the parameter table of type *ABQ_PHASE_TRANS_SolidPhases*. Leave blank if the number of child phases is less than 3.

Property Table Type Reference

These property table types are used in special-purpose material modeling techniques.

In this section:

- *ABQ_Cure_ReactionKinetics_MaxConversion*
- *ABQ_Cure_ReactionKinetics_Tabular*
- *ABQ_Cure_ShrinkageCoeff_Aniso*
- *ABQ_Cure_ShrinkageCoeff_Iso*
- *ABQ_Cure_ShrinkageCoeff_Ortho*
- *ABQ_Cure_ShrinkageCoeff_Vol*
- *ABQ_MULTSCALE_AspectRatio*
- *ABQ_MULTSCALE_Chaboche*
- *ABQ_MULTSCALE_ExpLinearHard*
- *ABQ_MULTSCALE_IsoCTE*
- *ABQ_MULTSCALE_IsoElas*
- *ABQ_MULTSCALE_IsoHard*
- *ABQ_MULTSCALE_Oritens*
- *ABQ_MULTSCALE_PowerHard*
- *ABQ_MULTSCALE_PowerLaw*
- *ABQ_MULTSCALE_Prandtl*
- *ABQ_MULTSCALE_TransIsoCTE*
- *ABQ_MULTSCALE_TransIsoElas*
- *ABQ_MULTSCALE_Volfrac*
- *ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients*
- *ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac*
- *ABQ_PHASE_TRANS_Diffusional_TTT_Diagram*
- *ABQ_PHASE_TRANS_GrainSize_Coefficients*
- *ABQ_PHASE_TRANS_Martensitic_KM_Coefficients*
- *ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac*
- *ABQ_PHASE_TRANS_Martensitic_TTT_Diagram*
- *ABQ_PHASE_TRANS_SolidificationMap_Diagram*

ABQ_Cure_ReactionKinetics_MaxConversion

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify the maximum value of the conversion (or maximum degree of cure).

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ReactionKinetics_MaxConversion",
PROPERTIES=1
"Maximum Conversion $\alpha_{max}$"
```

Properties

Maximum Conversion	Maximum value of conversion (degree of cure) that can be reached during the curing process, α_{max}.

ABQ_Cure_ReactionKinetics_Tabular

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify the cure kinetics in the tabular form.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ReactionKinetics_Tabular", PROPERTIES=1,
 INDEPENDENT VARIABLES=1
"Conversion Rate $\dot{\alpha}$"
"Conversion $\alpha$"
```

Properties

Conversion Rate Conversion rate (rate of degree of cure), $\dot{\alpha}$.

Independent Variables

Conversion Conversion (degree of cure), α.

ABQ_Cure_ShrinkageCoeff_Aniso

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify anisotropic shrinkage coefficients for the cure model.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ShrinkageCoeff_Aniso", PROPERTIES=6
"Shrinkage Coefficient γ11"
"Shrinkage Coefficient γ22"
"Shrinkage Coefficient γ33"
"Shrinkage Coefficient γ12"
"Shrinkage Coefficient γ13"
"Shrinkage Coefficient γ23"
```

Properties

Shrinkage Coefficient γ_{11}	Anisotropic shrinkage coefficient, γ_{11}.
Shrinkage Coefficient γ_{22}	Anisotropic shrinkage coefficient, γ_{22}.
Shrinkage Coefficient γ_{33}	Anisotropic shrinkage coefficient, γ_{33}.
Shrinkage Coefficient γ_{12}	Anisotropic shrinkage coefficient, γ_{12}.
Shrinkage Coefficient γ_{13}	Anisotropic shrinkage coefficient, γ_{13}.
Shrinkage Coefficient γ_{23}	Anisotropic shrinkage coefficient, γ_{23}.

ABQ_Cure_ShrinkageCoeff_Iso

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify the isotropic shrinkage coefficient for the cure model.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ShrinkageCoeff_Iso", PROPERTIES=1
"Shrinkage Coefficient γ"
```

Properties

Shrinkage Coefficient γ	Isotropic shrinkage coefficient, γ.

ABQ_Cure_ShrinkageCoeff_Ortho

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify orthotropic coefficients of shrinkage for the cure model.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ShrinkageCoeff_Ortho", PROPERTIES=3
"Shrinkage Coefficient γ11"
"Shrinkage Coefficient γ22"
"Shrinkage Coefficient γ33"
```

Properties

Shrinkage Coefficient γ_{11}	Orthotropic shrinkage coefficient, γ_{11}.
Shrinkage Coefficient γ_{22}	Orthotropic shrinkage coefficient, γ_{22}.
Shrinkage Coefficient γ_{33}	Orthotropic shrinkage coefficient, γ_{33}.

ABQ_Cure_ShrinkageCoeff_Vol

Products: Abaqus/Standard

References:

- *Modeling the Cure Process in Thermosetting Polymers*

Overview

You use this property table type to specify the volumetric shrinkage coefficient for the cure model.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_Cure_ShrinkageCoeff_Vol", PROPERTIES=1
"Shrinkage Coefficient γvol"
```

Properties

Shrinkage Coefficient γ_{vol}	Volumetric shrinkage coefficient, γ_{vol}.

ABQ_MULTSCALE_AspectRatio

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to specify the aspect ratio of the inclusion.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_AspectRatio", PROPERTIES=1
"Aspect Ratio"
```

Properties

Aspect Ratio Aspect ratio of the inclusion, AR.

ABQ_MULTSCALE_Chaboche

Products: Abaqus/Standard

References:

- *Rate-Dependent Yield*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the Chaboche rate-dependent yield of the constituent.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_Chaboche", PROPERTIES=3
"Reference Strain"
"Viscosity Parameter"
"Rate Sensitivity Exponent"
```

Properties

Reference Strain	Reference strain, $\dot{\varepsilon}_0$.
Viscosity Parameter	Viscosity parameter, K.
Rate Sensitivity Exponent	Rate sensitivity exponent, n.

ABQ_MULTSCALE_ExpLinearHard

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the exponential linear hardening law of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_ExpLinearHard", PROPERTIES=4
"Initial Yield Stress"
"Linear Hardening Parameter"
"Exponential Hardening Parameter"
"Hardening Exponent"
```

Properties

Initial Yield Stress	Yield stress at zero plastic strain, σ_0.
Linear Hardening Parameter	Parameter of the linear term in the hardening law, k_1.
Exponential Hardening Parameter	Parameter of the exponential term in the hardening law, k_2.
Hardening Exponent	Hardening exponent, m.

ABQ_MULTSCALE_IsoCTE

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define isotropic thermal expansion of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_IsoCTE", PROPERTIES=1
"Thermal Expansion Coefficient"
```

Properties

Thermal Expansion Coefficient	Isotropic thermal expansion coefficient, α.

ABQ_MULTSCALE_IsoElas

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define isotropic elasticity of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_IsoElas", PROPERTIES=2
"Young's Modulus"
"Poisson's Ratio"
```

Properties

Young's Modulus	Young's modulus, E.
Poisson's Ratio	Poisson's ratio, ν.

ABQ_MULTSCALE_IsoHard

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the isotropic hardening of the constituents in the tabular form.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_IsoHard", PROPERTIES=1, INDEPENDENT
 VARIABLES=1
"Yield Stress"
"Plastic Strain"
```

Properties

Yield Stress Yield stress, $\bar{\sigma}$.

Independent Variables

Plastic Strain Equivalent plastic strain, $\bar{\varepsilon}^{pl}$.

ABQ_MULTSCALE_Oritens

Products: Abaqus/Standard

References:

- *Specifying a Second-Order Orientation Tensor in Abaqus/Standard*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the second-order orientation tensor of the inclusion.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_Oritens", PROPERTIES=6
"a11"
"a22"
"a33"
"a12"
"a13"
"a23"
```

Properties

a11 11-component of the orientation tensor, a_{11}.

a22 22-component of the orientation tensor, a_{22}.

a33 33-component of the orientation tensor, a_{33}.

a12 12-component of the orientation tensor, a_{12}.

a13 13-component of the orientation tensor, a_{13}.

a23 23-component of the orientation tensor, a_{23}.

ABQ_MULTSCALE_PowerHard

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the power law hardening of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_PowerHard", PROPERTIES=3
"Initial Yield Stress"
"Strength Coefficient k"
"Hardening Exponent n"
```

Properties

Initial Yield Stress	Yield stress at zero plastic strain, σ_0.
Strength Coefficient k	Strength coefficient, k.
Hardening Exponent n	Hardening exponent, n.

ABQ_MULTSCALE_PowerLaw

Products: Abaqus/Standard

References:

- *Rate-Dependent Yield*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the power law rate-dependent yield of the constituent.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_PowerLaw", PROPERTIES=2
"Reference Strain"
"Power Law Exponent"
```

Properties

Reference Strain	Reference strain rate, $\dot{\varepsilon}_0$.
Power Law Exponent	Power law exponent, n.

ABQ_MULTSCALE_Prandtl

Products: Abaqus/Standard

References:

- *Rate-Dependent Yield*
- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define the Prandtl rate-dependent yield of the constituent.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_Prandtl", PROPERTIES=3
"Viscosity Parameter η"
"Viscosity Parameter β"
"Rate Sensitivity Exponent"
```

Properties

Viscosity Parameter η	Viscosity parameter, η.
Viscosity Parameter β	Viscosity parameter, β.
Rate Sensitivity Exponent	Rate sensitivity exponent, n.

ABQ_MULTSCALE_TransIsoCTE

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define transversely isotropic thermal expansion of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_TransIsoCTE", PROPERTIES=2
"Longitudinal Thermal Expansion Coefficient"
"Transverse Thermal Expansion Coefficient"
```

Properties

Longitudinal Thermal Expansion Coefficient	Thermal expansion coefficient in the longitudinal direction, α_{11}.
Transverse Thermal Expansion Coefficient	Thermal expansion coefficient in the transverse direction, α_{22}.

ABQ_MULTSCALE_TransIsoElas

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to define transversely isotropic elasticity of the constituents.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_TransIsoElas", PROPERTIES=5
"Longitudinal Modulus"
"Transverse Modulus"
"In-plane Poisson's Ratio"
"Out-of-plane Poisson's Ratio"
"In-plane Shear Modulus"
```

Properties

Longitudinal Modulus
Modulus in the longitudinal direction, E_1.

Transverse Modulus
Modulus in the transverse direction, E_2.

In-plane Poisson's Ratio
In-plane Poisson's ratio, ν_{12}.

Out-of-plane Poisson's Ratio
Out-of-plane Poisson's ratio, ν_{23}.

In-plane Shear Modulus
In-plane shear modulus, G_{12}.

ABQ_MULTSCALE_Volfrac

Products: Abaqus/Standard

References:

- *Modeling Multiscale Materials with Deep Material Networks (DMNs)*

Overview

You use this property table type to specify the volume fraction of the inclusion.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_MULTSCALE_Volfrac", PROPERTIES=1
"Volume Fraction"
```

Properties

Volume Fraction — Volume fraction of the inclusion, V_f.

ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the coefficients of the Johnson-Mehl-Avrami (JMA) model for a diffusional transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients"` in the material definition to specify the coefficients of the Johnson-Mehl-Avrami (JMA) model for a diffusional transformation whose input option is set to `JMA` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*NameOfTransformation*`_JMA_Coefficients"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Diffusional_JMA_Coefficients",
 PROPERTIES=2
 "JMA Coefficient k"
 "JMA Coefficient n"
```

Properties

JMA Coefficient k	Reaction rate constant, k, of the JMA model.
JMA Coefficient n	Avrami exponent, n, of the JMA model.

ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the equilibrium proportion of parent phases for a diffusional transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac"` in the material definition to specify the equilibrium proportion of parent phases for a diffusional transformation.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*NameOfTransformation*`_ParentEquiFrac"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Diffusional_ParentEquiFrac",
PROPERTIES=1
 "Parent Equilibrium Proportion"
```

Properties

Parent Equilibrium Proportion Equilibrium relative total volume fraction of parent phases under isothermal condition, $f_{p,0}^{eq}$.

ABQ_PHASE_TRANS_Diffusional_TTT_Diagram

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the starting and ending curves of the time-temperature-transformation (TTT) diagram for a diffusional transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Diffusional_TTT_Diagram"` in the material definition to specify the starting and ending curves of the time-temperature-transformation (TTT) diagram for a diffusional transformation whose input option is set to `TTT` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*`NameOfTransformation`*`_TTT_Diagram"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Diffusional_TTT_Diagram",
PROPERTIES=2
 "Start Time"
 "End Time"
```

Properties

Start Time	Time at the start of the transformation.
End Time	Time at the end of the transformation.

ABQ_PHASE_TRANS_GrainSize_Coefficients

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the coefficients of the grain-growth law for modeling grain growth.

You must include a property table of type "ABQ_PHASE_TRANS_GrainSize_Coefficients" in the material definition to specify the coefficients of the grain-growth law for modeling grain growth.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_GrainSize_Coefficients",
PROPERTIES=3
 "Grain-Growth Coefficient k"
 "Grain-Growth Coefficient n"
 "Grain-Growth Coefficient Q"
```

Properties

Grain-Growth Coefficient k	Coefficient k of the parabolic grain-growth law.
Grain-Growth Coefficient n	Exponent n of the parabolic grain-growth law.
Grain-Growth Coefficient Q	Active energy, Q, of the parabolic grain-growth law.

ABQ_PHASE_TRANS_Martensitic_KM_Coefficients

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the coefficients of the Koistinen-Marburger (KM) model for a martensitic transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Martensitic_KM_Coefficients"` in the material definition to specify the coefficients of the Koistinen-Marburger (KM) model for a martensitic transformation whose input option is set to `KM` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*`NameOfTransformation`*`_KM_Coefficients"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Martensitic_KM_Coefficients",
PROPERTIES=1
 "KM Coefficient γ"
```

Properties

KM Coefficient γ	Coefficient γ of the KM model.

ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the retained proportion of parent phases for a martensitic transformation as a function of the volume fraction at the beginning of the transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac"` in the material definition to specify the retained proportion of parent phases for a martensitic transformation as a function of the volume fraction at the beginning of the transformation.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*NameOfTransformation*`_ParentRetainedFrac"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Martensitic_ParentRetainedFrac",
 PROPERTIES=1,
INDEPENDENT VARIABLES=1
 "Retained Proportion"
 "Starting Proportion"
```

Properties

Retained Proportion — Relative total volume fraction of parent phases retained at the end of the transformation, f_{pr}.

Independent Variables

Starting Proportion — Relative total volume fraction of parent phases at the beginning of the transformation, $f_p(\theta_0)$.

ABQ_PHASE_TRANS_Martensitic_TTT_Diagram

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the starting and ending curves of the time-temperature-transformation (TTT) diagram for a martensitic transformation.

You must include a property table of type `"ABQ_PHASE_TRANS_Martensitic_TTT_Diagram"` in the material definition to specify the starting and ending curves of the time-temperature-transformation (TTT) diagram for a martensitic transformation whose input option is set to `TTT` in the parameter table of type `"ABQ_PHASE_TRANS_Transformations"`.

The property table must have a unique label of `"ABQ_PHASE_TRANS_Trans_`*NameOfTransformation*`_TTT_Diagram"`, where *NameOfTransformation* is the name of the transformation defined in the parameter table of type *"ABQ_PHASE_TRANS_Transformations"*.

Property Table Type Definition

```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_Diffusional_TTT_Diagram",
PROPERTIES=2
 "Start Temperature"
 "End Temperature"
```

Properties

Start Temperature — Temperature at the start of the transformation, M_s.

End Temperature — Temperature at the end of the transformation, M_f.

ABQ_PHASE_TRANS_SolidificationMap_Diagram

Products: Abaqus/Standard

References:

- *Modeling Metallurgical Phase Transformations*

Overview

You use this property table type to specify the equiaxial and columnar boundary lines in the solidification map for modeling grain morphology.

You must include two property tables of type `"ABQ_PHASE_TRANS_SolidificationMap_Diagram"` in the material definition to specify the equiaxial and columnar boundary lines in the solidification map for modeling grain morphology. The property tables must have unique labels of `"ABQ_PHASE_TRANS_SolidificationMap_Diagram_Equiaxied"` and `"ABQ_PHASE_TRANS_SolidificationMap_Diagram_Columnar"`, respectively.

Property Table Type Definition

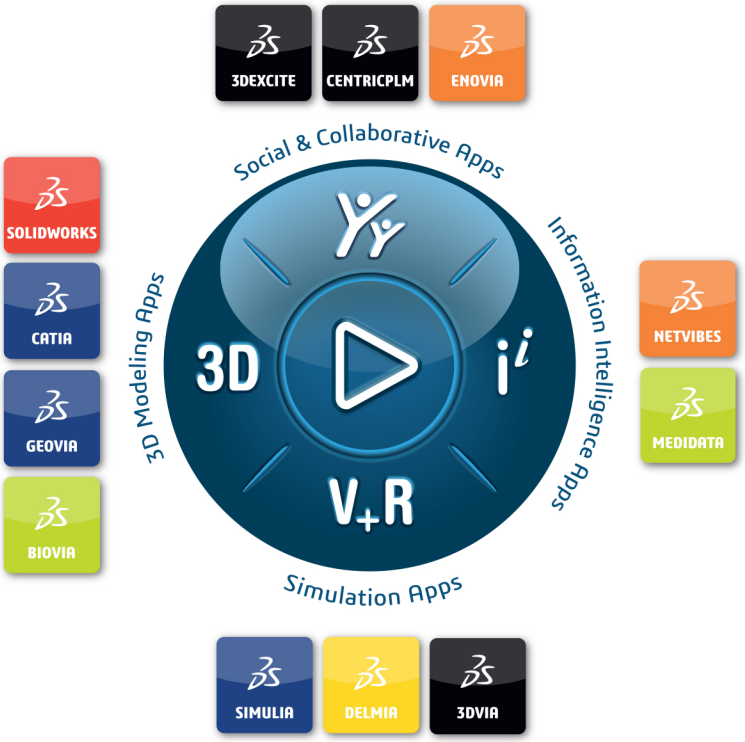
```
*PROPERTY TABLE TYPE, NAME="ABQ_PHASE_TRANS_SolidificationMap_Diagram",
PROPERTIES=1, INDEPENDENT VARIABLES=1
 "LogG"
 "LogR"
```

Properties

LogG Decimal logarithm of the thermal gradient magnitude.

Independent Variables

LogR Decimal logarithm of the solidification rate.



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.