SANDIA REPORT

SAND2017-9758
Unlimited Release
Printed September 11, 2017

Sierra/SolidMechanics 4.46 Example Problems Manual

SIERRA Solid Mechanics Team
Computational Solid Mechanics and Structural Dynamics Department
Engineering Sciences Center

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology and Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2017-9758
Unlimited Release
Printed: September 11, 2017

Sierra/SolidMechanics 4.46 Example Problems Manual

SIERRA Solid Mechanics Team
Computational Solid Mechanics and Structural Dynamics Department
Engineering Sciences Center
Sandia National Laboratories
Box 5800
Albuquerque, NM 87185-0380

Abstract

Presented in this document are tests that exist in the Sierra Solid Mechanics example problem suite. The purpose of these examples is to showcase common and advanced code capabilities. Note that many other regression and verification tests exist in the Sierra/SM test suite that have not been included in this manual.

Acknowledgments

This document is the result of the collective effort of a number of individuals. This document was originally written primarily by Steven Gomez, Jason Ivey and Patrick Suszko, and initially reviewed by Kevin Long, Kendall H. Pierson and Michael Tupek.

The current core development team responsible for the Sierra/SolidMechanics codes includes: Nathan K. Crane, Gabriel J. de Frias, San Le, David J. Littlewood, Mark T. Merewether, Matthew D. Mosby, Kendall H. Pierson, Julia A. Plews, Vicki L. Porter, Timothy R. Shelton, Jesse D. Thomas, Michael R. Tupek, Michael G. Veilleux, and Patrick G. Xavier.

Contents

List of Figures

List of Tables

Introduction

The Sierra/SM Example Problem Manual is divided into chapters that represent related capabilities. The tests listed in each chapter verify some aspect of that suite of capabilities and are automatically run nightly. The test files for these problems may be found in the Sierra regression test repository. See also the example problems at http://compsim.sandia.gov/compsim/Team_SM_Content/html_src/docs.php?ref=examples.

Chapter 1

Contact

1.1 Newton Cradle

Product: Sierra/Solid Mechanics - Explicit Analysis

1.1.1 Problem Description

This example demonstrates the conservation of momentum and kinetic energy (through basic Newtonian mechanics) within an explicit dynamic analysis with contact. There are currently three geometric cases available for running: the dropping of one, two, and three balls. These cases can be tested through commenting/uncommenting the Genesis files of interest referenced inside the input deck. These configurations can be seen in Figure 1.1. The 5 ball-chain system contains a default initial configuration of one raised ball.

The balls are given an initial rotational displacement through their geometrical location specified in Cubit. The at rest balls touch at an initial position as fine as the mesh generated.

1.1.2 Loading and Boundary Conditions

The pendulum wires are defined as a truss section with each of the five balls containing an inner rigid body core. Connected in a v-shaped manner, the uppermost 'ceiling' node of the wires define the reference location to each of the five inner rigid body volumes. These rigid bodies are only allowed to rotate along the z-axis and translate along the x-axis and y-axis. Controlling the rigid body displacements in this manner allows one to bypass making the spherical nodeset rigid, hence restricting movement.

A constant and uniform gravitational load is applied to the system.

1.1.3 Material Model

Assuming small deformations and simple linear elastic behavior, the outermost material for each respective sphere is defined in an elastic model. Modulus of Elasticity values were picked out of convenience, i.e., stiff enough for minimal elastic deformation and compliant enough to minimize computational expense for the explicit analysis.

Iterations performed to optimize these parameters conceded the outer sphere's Young's Modulus at approximately 10^4 times less than that of steel. The inner spheres of the model were defined as rigid bodies.

The following material properties were used during analysis:

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 1.1: Newton Cradle Materials

Newton Cradle		
Young's Modulus: Outer Sphere	E	200×10^5
Young's Modulus: Core	E	Rigid
Young's Modulus: Wire/String	E	100
Poisson's Ratio	ν	0.3
Density	ρ	7.48×10^3

1.1.4 Finite Element Model

To eliminate rigid body contact of the outer spheres, mesh creation of this model required independent nodeset specification for inner and outer spherical volumes (see figure 1.2). The overlap of volumes was eliminated by first creating inner and outer spheres of radii 1 and 0.25 units respectively, followed by elimination/recreation of the inner sphere volumes. Both inner and outer spheres were then meshed using a standard four noded tetrahedral mesh.

The wires were given a truss element type.

1.1.5 Feature Tested

Explicit contact and rigid bodies.

1.1.6 Results and Discussion

In an idealized Newton Cradle exhibiting perfectly elastic collisions, the Total Kinetic Energy before and after subsequent collisions would be equal: no energy stored in the steel balls would be lost to inelastic processes. In reality, the balls are elastic, so some energy is stored in them as elastic Strain Energy and Internal Energy of the balls themselves. Moreover, artificial bulk viscosity is used to prevent high frequency responses from the available energy in the system. Viscosity is dissipated and removes Total Energy from the system, hence we do not expect to perfectly conserve Kinetic Energy.

The Internal Energy is calculated as the stress times strain rate integrated over time, while the Strain Energy is calculated as the elastic portion of the stress times strain rate. The difference in

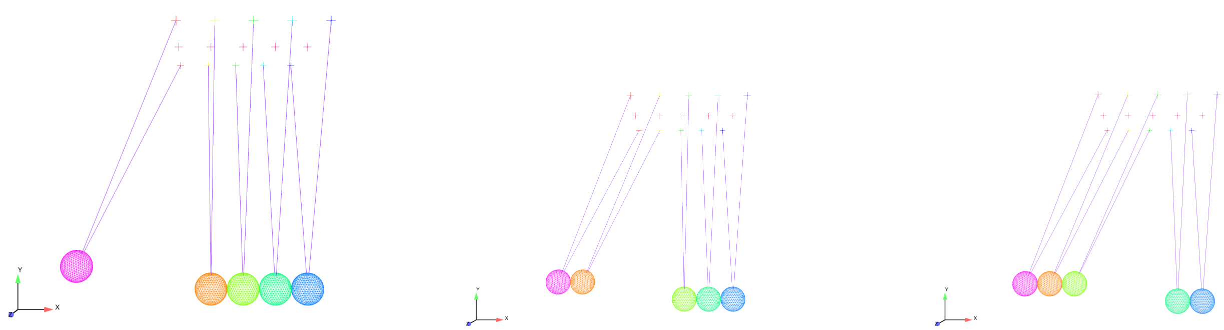
Internal Energy and Strain Energy normalized by the Maximum Initial Potential Energy can be seen in Figure 1.3. At sphere-to-sphere contact initiation, steep rates of change in Internal Energy and Strain Energy are present, followed by an approximately constant tiering when the opposing ball rebounds. This stair-step cycle continues with subsequent oscillations of the Newton Cradle. System energy conservation, including Kinetic, Internal, and Potential Energy, is presented in Figure 1.4. The increasing Internal Energy over the simulation can be attributed to bulk viscosity.

At 20 seconds of test run time under the configured geometries and loading, the Newton Cradle will swing for approximately 2 periods. The run time for energy dissipation of the system can be carried out as desired through input deck specification.

Table 1.2: Abbreviations Used in Results

Energy Variables	
Kinetic Energy	KE
Internal Energy	IE
Potential Energy	PE
Strain Energy	SE
Total Initial Energy	TIE

For input deck see Appendix A.1.



(a) 1 ball **(b)** 2 balls **(c)** 3 balls

Figure 1.1: Initial Configurations

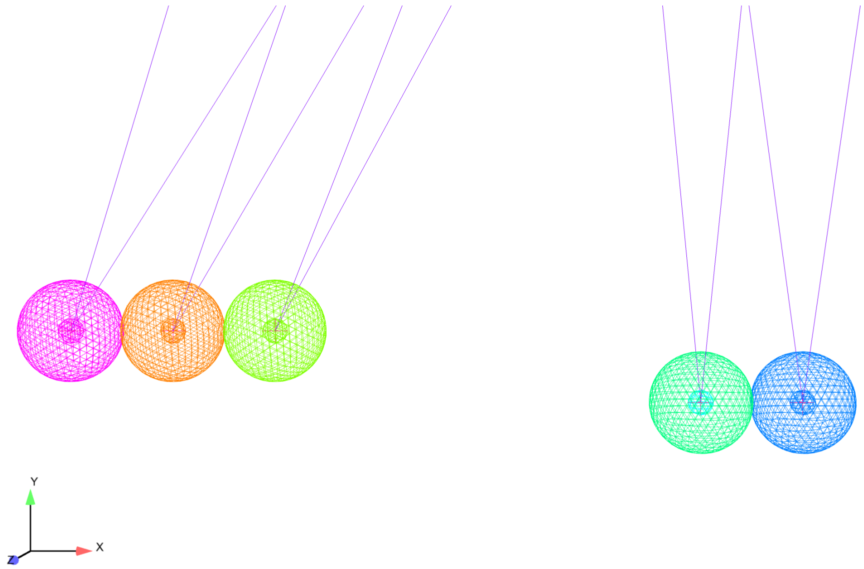


Figure 1.2: Rigid Body Highlight

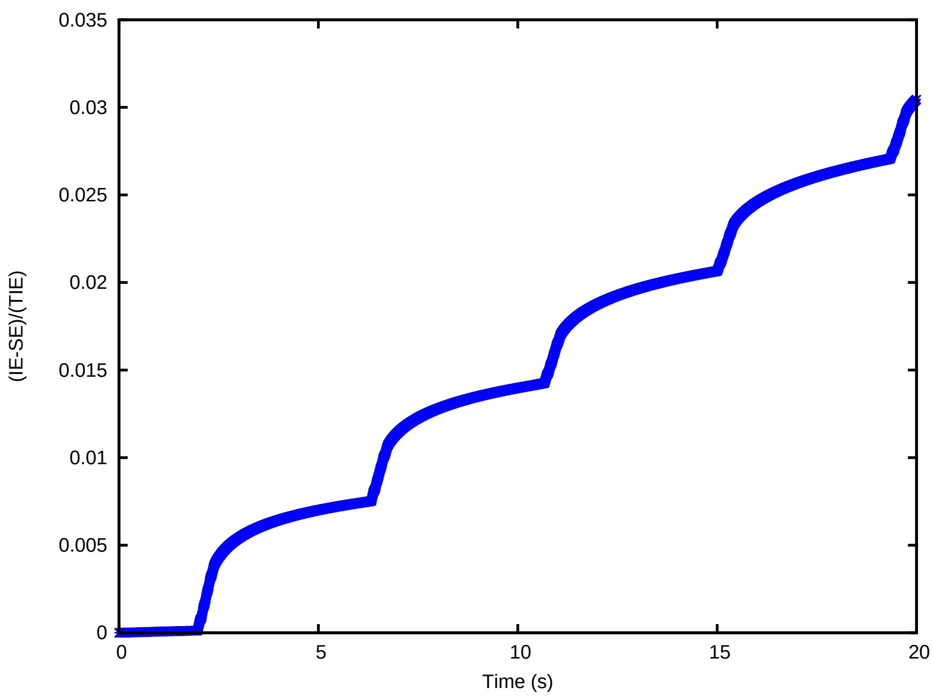


Figure 1.3: Normalized Dissipation Energy

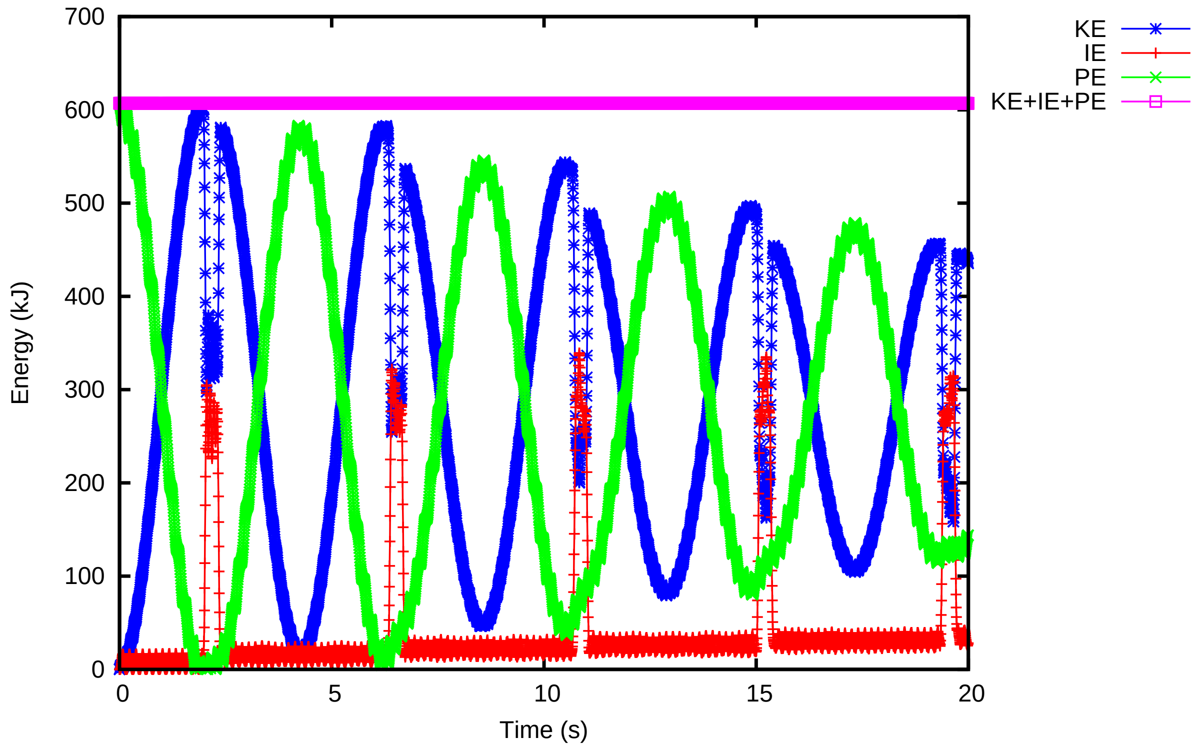


Figure 1.4: Energy History

1.2 Bullet Collision

Product: Sierra/Solid Mechanics - Explicit Analysis.

1.2.1 Problem Description

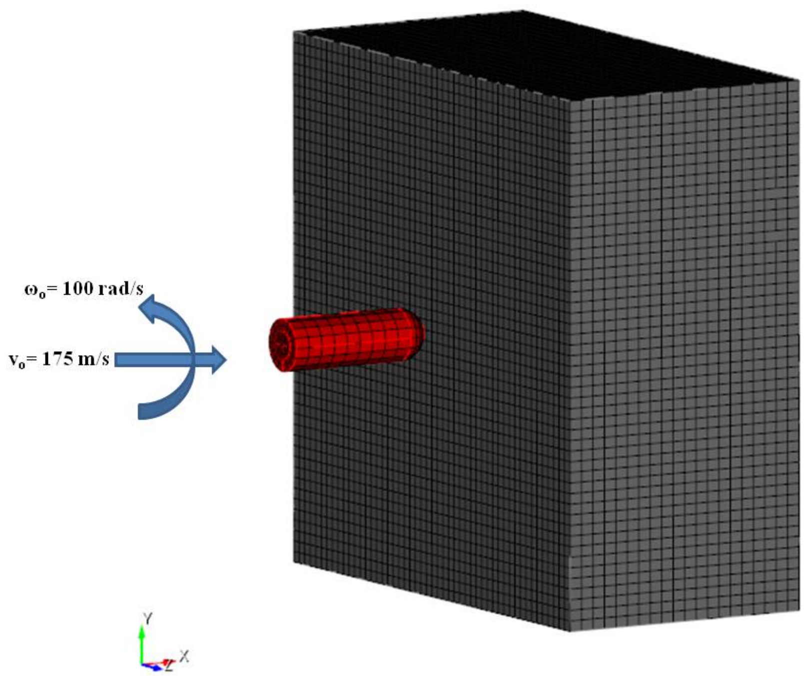


Figure 1.5: Initial Setup

The primary purpose of this example problem is to demonstrate both the analytic functions and the user defined output. In the problem, a bullet is given initial angular and translational velocities. The bullet then collides with a block, and the non-dimensional torque is output. The initial configuration can be seen in Figure 1.5.

1.2.2 Loading and Boundary Conditions

The bullet is constrained to translation along the X axis, and rotation about the X axis. The surfaces of the block in the XY and XZ planes are given a fixed displacement in the X Y and Z directions. The block is still allowed to deform when it is hit by the bullet. Contact is enforced by setting general contact ON and creating a constant friction model.

1.2.3 Material Model

The bullet is modeled as an elastic plastic body and is given material properties similar to steel. The block is given arbitrary material properties and is also modeled as an elastic plastic body. The material models can be seen in Table 1.3 and Table 1.4.

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Material Properties		
Young's Modulus	E	350×10^9
Poisson's Ratio	ν	0.3333
Density	ρ	10^4
Yield Stress	σ_{yield}	4.5×10^8

Table 1.3: Bullet Material Properties

Material Properties		
Young's Modulus	E	170×10^6
Poisson's Ratio	ν	0.15
Density	ρ	2×10^3
Yield Stress	σ_{yield}	2.0×10^6

Table 1.4: Block Material Properties

1.2.4 Finite Element Model

This problem contains just under 7000 elements, 6200 of which are associated with the block. There are just over 800 elements in the bullet. Figure 1.5 shows the mesh generated for the problem.

1.2.5 Feature Tested

The user defined output and analytic functions are the main features tested in this problem. Secondary features include explicit contact and the use of aprepro variables.

1.2.6 Results and Discussion

After initial rotational and translational velocities are applied, the bullet collides with the block. This can be seen in Figure 1.6. Due to its rotation, a reaction torque is applied to the bullet by the block. The non-dimensional form of the torque is found using analytic functions and a user defined output. The equations used to find the non-dimensional torque are based on the Hertz solution. While the torque is calculated for the entirety of the problem, it is found using contact and is thus zero for parts of the problem. Figure 1.7 shows the graph of the non-dimensional torque. The initial equations for the calculation contained two issues. First, the sign of the contact reaction force normal to the contact surface, defined as a variable P, may or may not be negative

Variables	
Contact Reaction Force	P
Poisson's Ratio	ν
Contact Radius	R_c
Bullet Radius	R_b
Non-dimensional Torque	T
Young's Modulus of Elasticity	E
Friction Coefficient	μ
Reaction Force on far end of Bullet	Ty_top

Table 1.5: Table of Variables

depending on where the global origin is set. If P is negative, the equation for the contact radius, R_c, will contain a negative cubed root, as shown below. Table 1.5 contains a list of the variables used in the following calculations. Compute global P as the sum of nodal contact forces in the x direction,

$$P = \sum \text{Force\_Contact(x)}. \tag{1.1}$$

Compute global R_c from the expression,

$$R_c = \frac{3}{4}^{(1/3)} \times \left((-1 + \nu^2) \times P \times \frac{R_b}{E} \right)^{(1/3)}. \tag{1.2}$$

The user defined output cannot calculate the cubed root of a negative number. To account for this case, the absolute value is taken inside the cubed root. The updated equation for the contact radius is,

$$R_c = \frac{3}{4}^{(1/3)} \times \left(\frac{((-1 + \nu^2) \times P \times \frac{R_b}{E})}{\text{abs}((-1 + \nu^2) \times P \times \frac{R_b}{E})} \right) \times \left(\text{abs}((-1 + \nu^2) \times P \times \frac{R_b}{E}) \right)^{(1/3)}. \tag{1.3}$$

The second issue is that the torque and contact radius calculations are based on contact. When the bullet and the concrete slab are not in contact, P is zero. This causes both the contact radius calculation above and the non-dimensional torque calculation below to have division by zero.

Compute global T from the expression,

$$T = \text{abs}\left(\frac{\text{Ty\_top}}{(\mu \times P \times R_c)} \right). \tag{1.4}$$

While the simulation will still run, the non-dimensional torque and contact radius cannot be graphed. To visualize the results, a small constant perturbation is introduced in both equations. The perturbation only affects the results during contact, and it is shown below.

Compute global R_c from the expression,

$$R_c = \frac{3}{4}^{(1/3)} \times \left(\frac{((-1+\nu^2) \times P \times \frac{R_b}{E} + 0.0001)}{\mathrm{abs}((-1+\nu^2) \times P \times \frac{R_b}{E} + 0.0001)} \right) \times \left(\mathrm{abs}((-1+\nu^2) \times P \times \frac{R_b}{E}) \right)^{(1/3)}. \tag{1.5}$$

Compute global T from the expression,

$$T = \mathrm{abs}\left(\frac{Ty\_top}{(\mu \times P \times R_c + 0.0001)} \right). \tag{1.6}$$

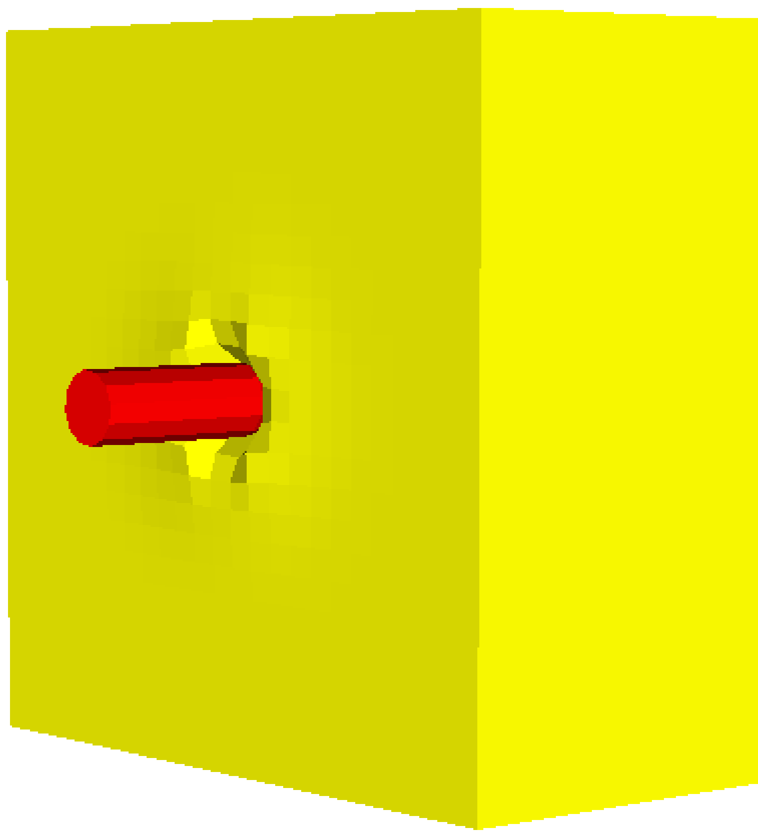


Figure 1.6: Bullet in contact with block

For input deck see Appendix A.2.

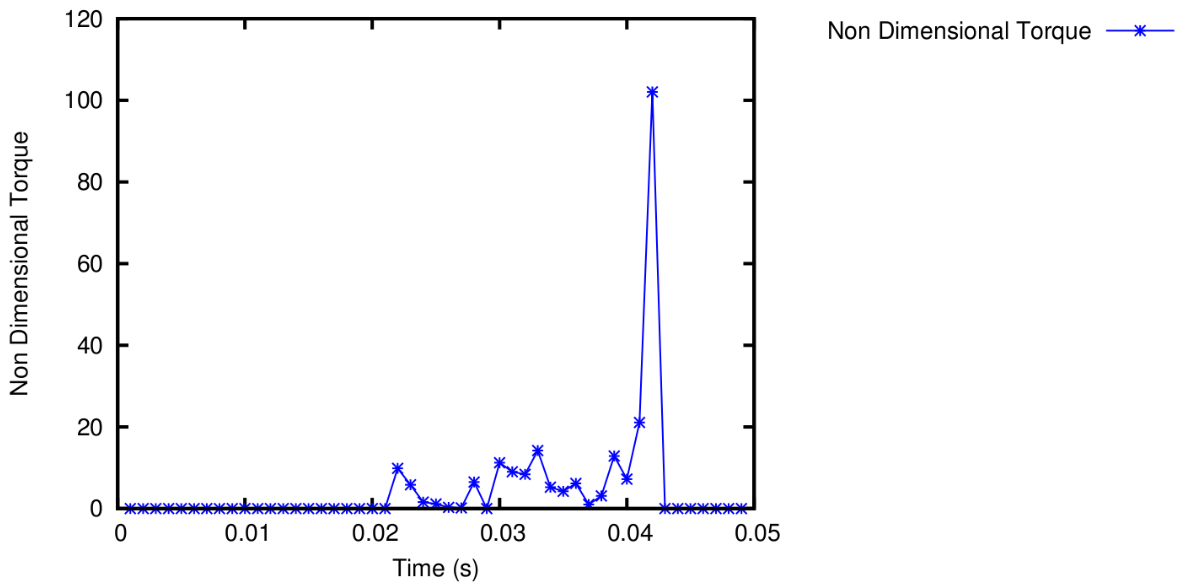


Figure 1.7: Non-dimensional Torque. See equation 1.6.

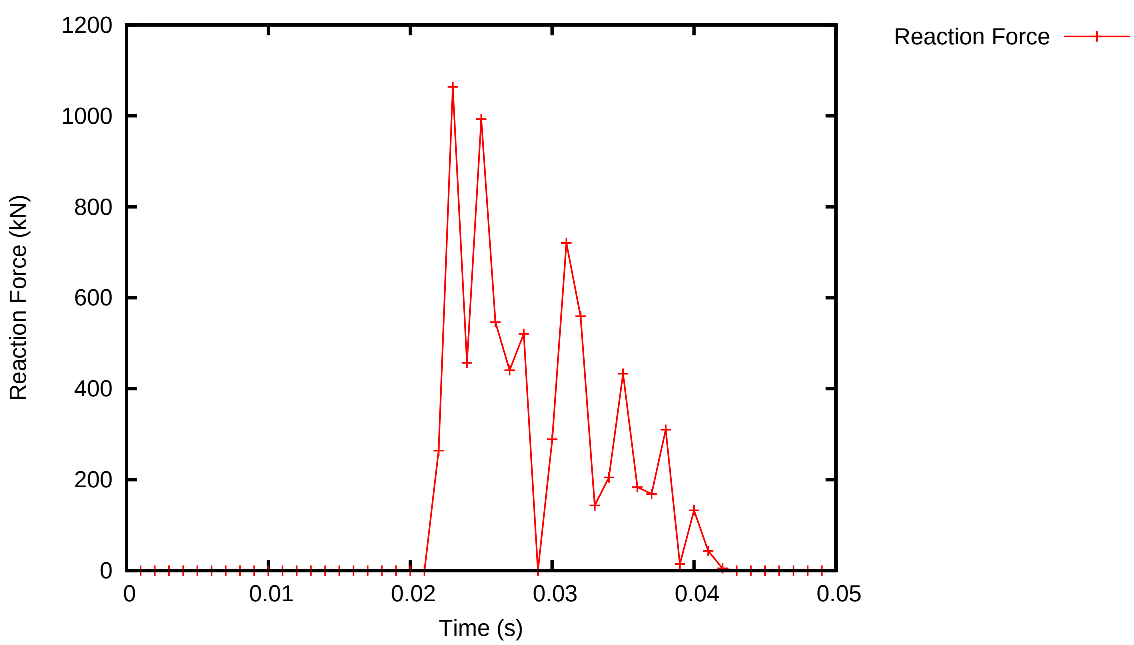


Figure 1.8: Reaction Force. See equation 1.1.

1.3 Analytic Planes

Product: Sierra/Solid Mechanics

1.3.1 Problem Description

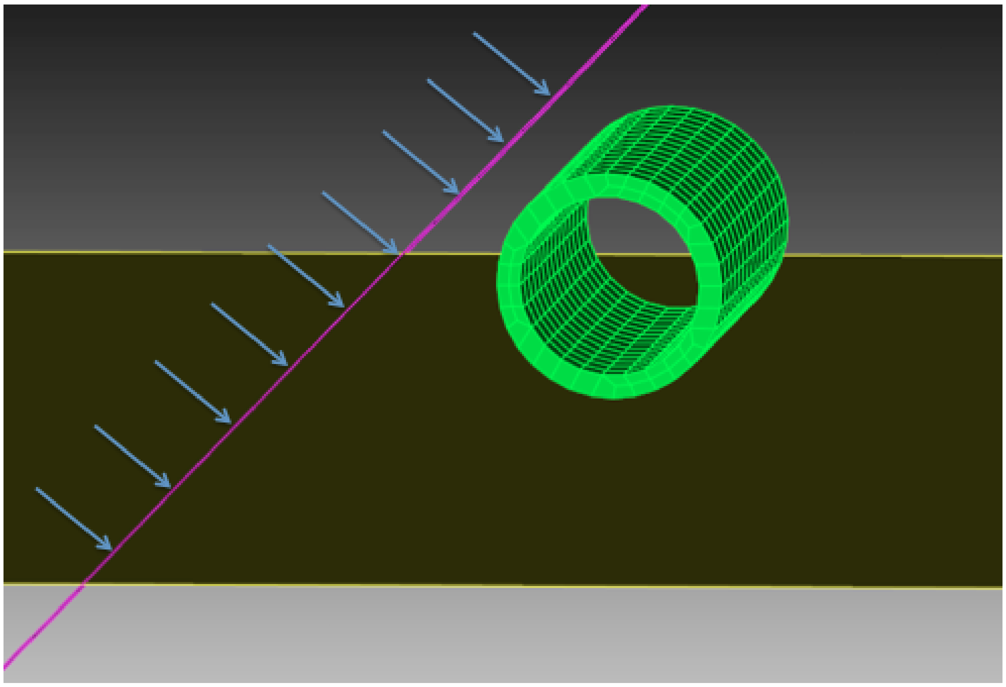


Figure 1.9: Analytic Surfaces problem set-up.

The purpose of the following numerical example is to display how to setup a problem using analytic surfaces. Consider a cylinder hovering above an analytic surface and to the right of an angled analytic surface.

1.3.2 Loading and Boundary Conditions

A prescribed displacement is applied to the angled analytic surface in the upper left, and a fixed displacement is applied to the analytic surface below the cylinder. To see how the analytic surfaces are created, how the boundary conditions are applied, and how contact is setup, see Appendix A.3.

1.3.3 Feature Tested

Analytic surfaces with DASH contact.

1.3.4 Results and Discussion

As can be seen in Figure 1.10 below, the angled analytic surface is the first to contact the cylinder. Then, as the cylinder displaces, the cylinder then contacts the lower analytic plane and ricochet's off to the right (Figure 1.11).

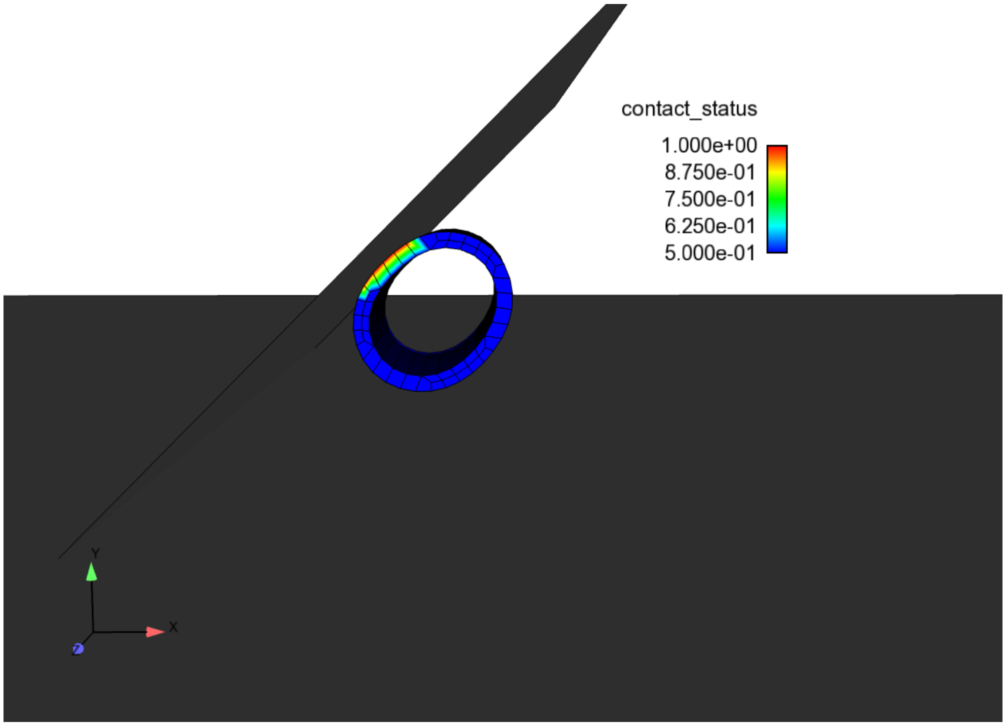


Figure 1.10: Contact with Angled Plane.

For input deck see Appendix A.3.

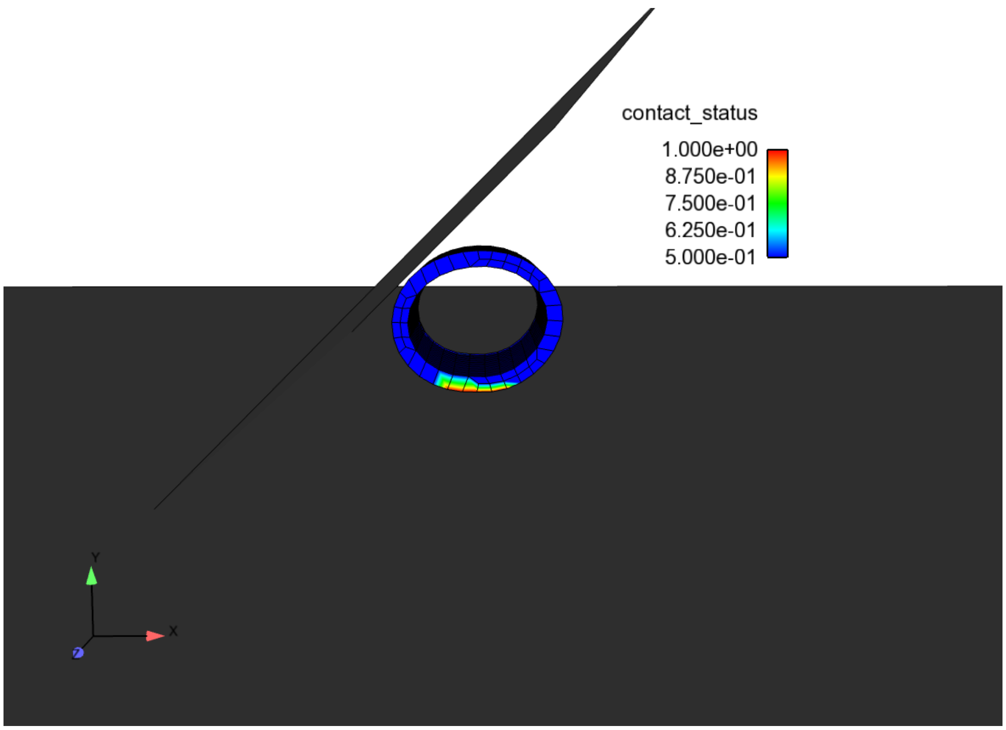


Figure 1.11: Contact with Lower Plane.

1.4 Curved Surface Friction Behavior

Product: Sierra/Solid Mechanics - Explicit Analysis

1.4.1 Problem Description

The purpose of this problem was to examine the frictional contact behavior of explicit analysis in Sierra, specifically that of curved contact surfaces. The scenario of this problem was this: a cylindrical mass is placed atop a slope of a given angle and allowed to roll down it. As it rolls, slip may occur at the contact surface if the frictional force is not large enough compared to the cylinder's acceleration along the slope. Naturally, when the slope is horizontal or almost horizontal, the cylinder should "stick" to the surface and roll without slipping if it rolls at all, but when the slope is oriented near vertical, the cylinder should hardly spin at all, and will experience large slip values as it moves along the surface. With this in mind, the slip at the contact surface was measured for several slope angles and several frictional coefficients, then compared to ideal behavior to determine the accuracy of the frictional model (Figure 1.16).

1.4.2 Mesh Model Setup

The cylinder was modeled with a radius of 0.2m, and a length of 0.2m. A conversion study was performed to determine when the model mesh was fine enough that the slip data obtained from the model had begun to converge on an accurate solution without requiring undue processing time. The results are shown in Figure 1.12, and a mesh refinement level of 5 was used as a result. Note: the mesh refinement level refers to the auto size setting in Cubit's meshing commands.
The slope was modeled as a rectangular block 10m long, 0.5m wide, and 0.1m thick. Mesh elements were chosen as 0.1m cubes to balance the need for a fine mesh with the need for fast run times (Figure 1.13).

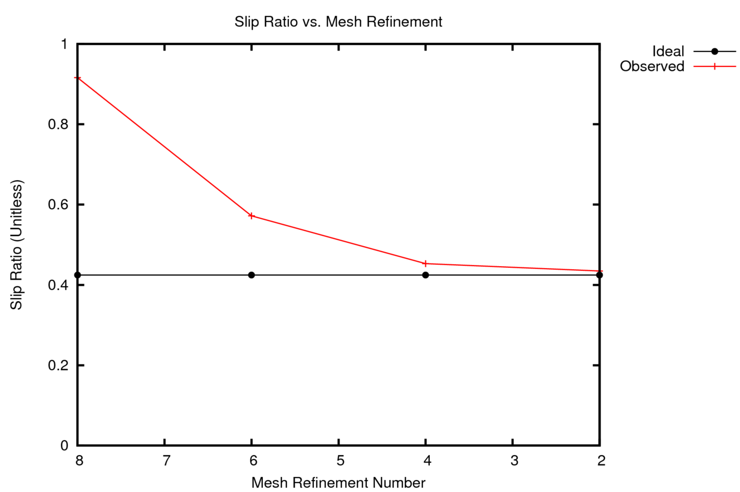


Figure 1.12: Mesh Convergence

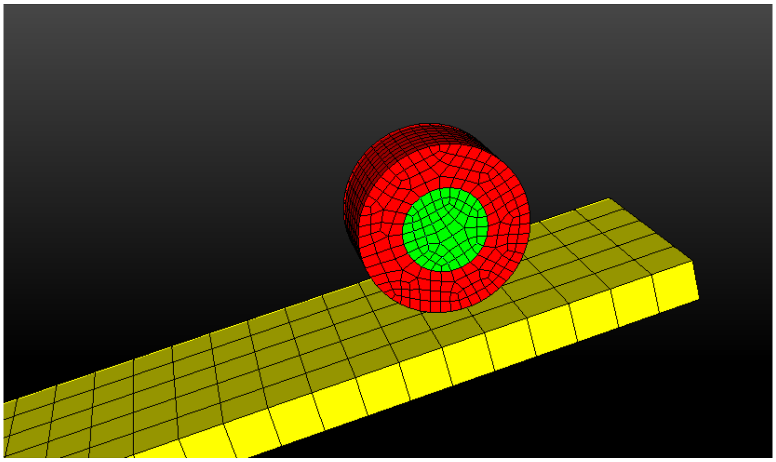


Figure 1.13: Mesh Refinement

1.4.3 Boundary Conditions and General Problem Setup

The slope was constrained to zero displacement in each direction, while the cylinder had no constraints on its movement, and was introduced with zero initial velocity. Because the rotational velocity of the cylinder was needed to determine its slip magnitude, it was first modeled as a rigid body to access the rigid body output variables. This resulted in poor results at friction coefficients above 0.25 (Figure 1.14), so the cylinder was instead modeled as a non-rigid cylinder with a rigid body core that had been merged with it. This resulted in more accurate results across the spectrum of examined friction coefficients, especially at higher settings (Figure 1.15). Note: The slope and cylinder were not part of the same block entity. The slope was not modeled as a rigid body, but due to its zero displacement boundary condition, it did not deform or experience stresses anywhere.

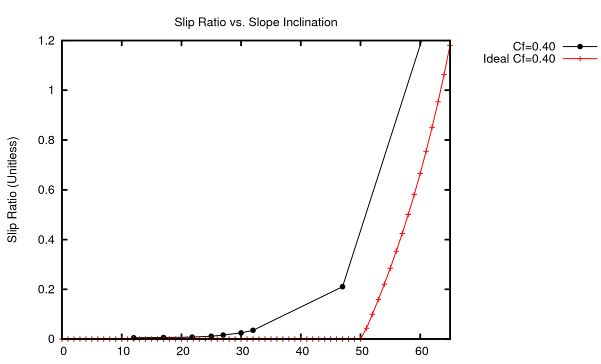


Figure 1.14: Rigid Body Contact

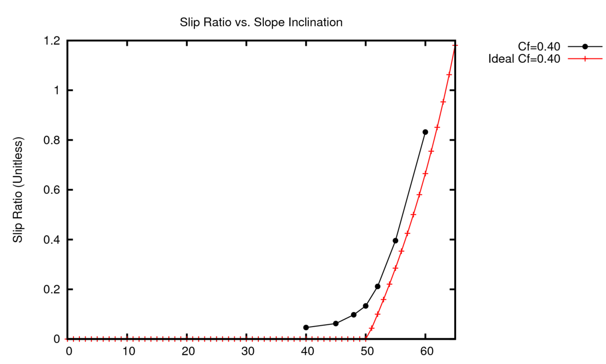


Figure 1.15: No Rigid Body Contact

Metric units were used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 1.6: Material Properties

Aluminum		
Young's Modulus:	E	68.9×10^6
Poisson's Ratio	ν	0.33
Density	ρ	2720
Yield Stress		276×10^6
Hardening Modulus		0.0

Contact was enforced as general contact with "skin all blocks" set to on. Gravity was enforced as a constant of $9.81 m/s^2$ in the negative y direction. Friction was modeled as a constant coefficient of friction which varied from test to test. Slope angle was determined by rotating both volumes in cubit by a prescribed angle.

1.4.4 Feature Tested

Curved surface frictional contact behavior.

1.4.5 Ideal Behavior Model

Ideal behavior was based on a basic physics model manually calculated as shown.

Nomenclature					
Friction Coefficient	(C_f)	Translational Velocity	(V)	Normal Force	(f_N)
Slope Angle	(θ)	Translational Acceleration	(a)	Frictional Force	(f_f)
Gravitational Constant	(g)	Rotational Velocity	(ω)	Gravitational Force	(f_g)
Cylinder Rotational Inertia	(I)	Rotational Acceleration	(α)		
Mass of Cylinder	(m)	Slip Ratio	(S_R)		
radius of Cylinder	(r)				

$$S_R = \frac{V}{\omega r} - 1 \tag{1.7}$$

$$S_R \neq 0 \;\; iff \;\; \alpha r \neq a \tag{1.8}$$

$$\alpha = \frac{f_f r}{I} = \frac{2 f_f}{mr} \tag{1.9}$$

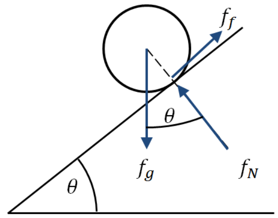


Figure 1.16: Basic Physics Model

$$I = \frac{1}{2}mr^2 \tag{1.10}$$

$$f_f = C_f f_N = C_f f_g cos(\theta) \tag{1.11}$$

$$a = (f_g sin(\theta) - f_f)/m = f_g(sin(\theta) - C_f cos(\theta))/m \tag{1.12}$$

$$\therefore \alpha r = a \quad becomes: \quad 2r(\frac{C_f f_g cos(\theta)}{mr}) = f_g \frac{sin(\theta) - C_f cos(\theta)}{m} \tag{1.13}$$

$$\therefore, \ S_R = 0 \quad when \quad \theta <= tan^{-1}(3C_f) \tag{1.14}$$

From this we see that there is a threshold beyond which the frictional force cannot overcome the gravitational force in the direction of motion. This threshold is defined as: $\theta = tan^{-1}(3C_f)$.

After the threshold has been reached, slip occurs. To measure slip relative to velocity, the slip ratio (S_R) is defined as $S_R = V/(\omega * r) - 1$, and is positive when the cylinder's tangential velocity is less than its translational velocity ($\omega * r < V$). The ideal slip ratio behavior was calculated as:

$$V = \int a dt = tg(sin(\theta) - C_f cos(\theta)) \tag{1.15}$$

$$\omega = \int \alpha dt = 2\frac{tg}{r} C_f cos(\theta) \tag{1.16}$$

$$S_R = \frac{tgr(sin(\theta) - C_f cos(\theta))}{2tgrC_f cos(\theta)} - 1 = \frac{1}{2}(\frac{tan(\theta)}{C_f} - 3) \tag{1.17}$$

From this, we can see that the slip ratio does not very with time as the cylinder moves down the slope, and we have an established ideal to rate results from the analysis.

1.4.6 Results and Discussion

The following graphs show the slip ratios of various frictional coefficients near their respective threshold angles, shown side by side with their ideal behavior. Figures(1.19 - 1.26) show the observed and ideal behavior of each friction coefficient, consolidated into Figure 1.17 and Figure 1.18. Note: Each data point represents a separate simulation.

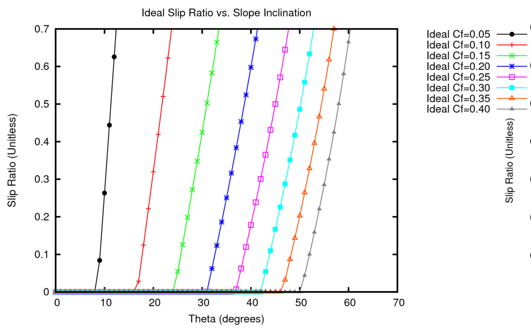


Figure 1.17: Predicted Slip

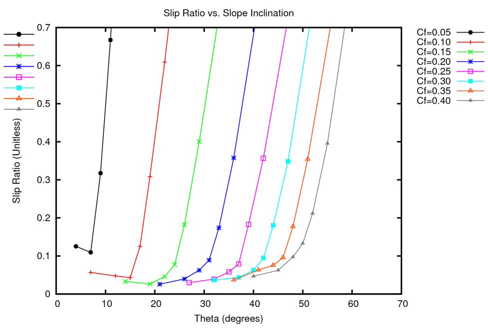


Figure 1.18: Observed Slip

1.4.7 Conclusion

Frictional contact with curved surfaces appears to behave as expected in explicit analysis, with results that converge to the ideal solution as mesh refinement increases. Defining an interior volume as a rigid body successfully produces accurate results without restricting access to rigid body output variables such as angular velocity.

For input deck see Appendix A.4

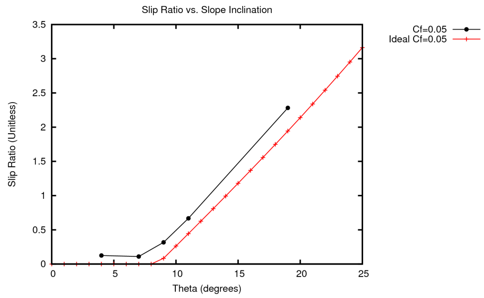


Figure 1.19: Coefficient 0.05

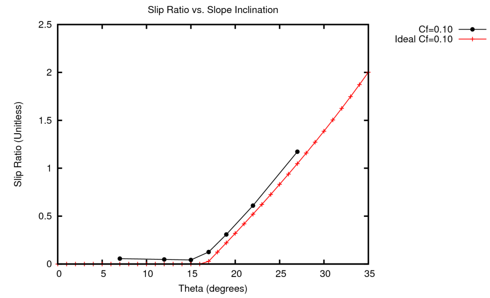


Figure 1.20: Coefficient 0.10

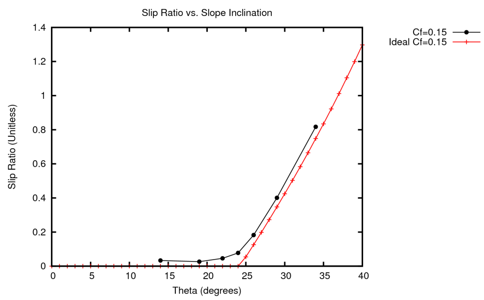


Figure 1.21: Coefficient 0.15

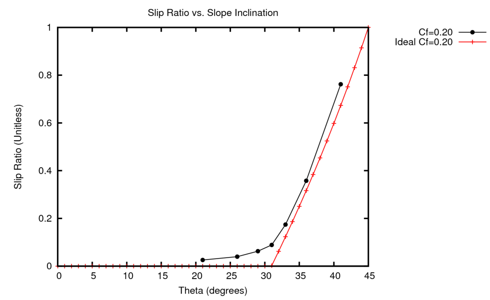


Figure 1.22: Coefficient 0.20

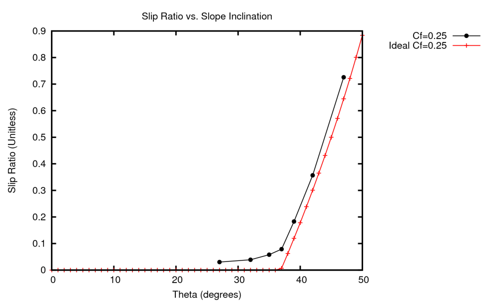


Figure 1.23: Coefficient 0.25

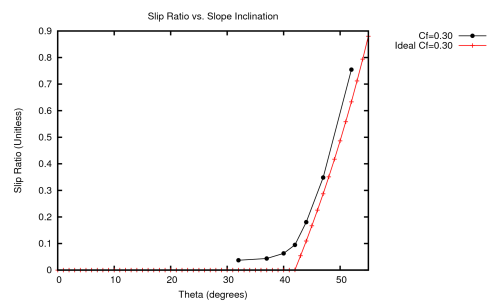


Figure 1.24: Coefficient 0.30

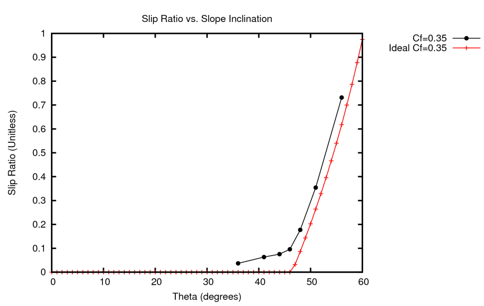


Figure 1.25: Coefficient 0.35

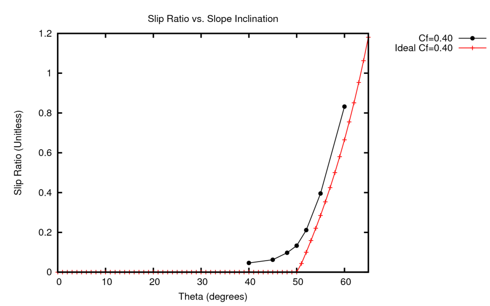


Figure 1.26: Coefficient 0.40

1.5 Plate Indentation

Product: Sierra/Solid Mechanics - Implicit Analysis

1.5.1 Problem Description

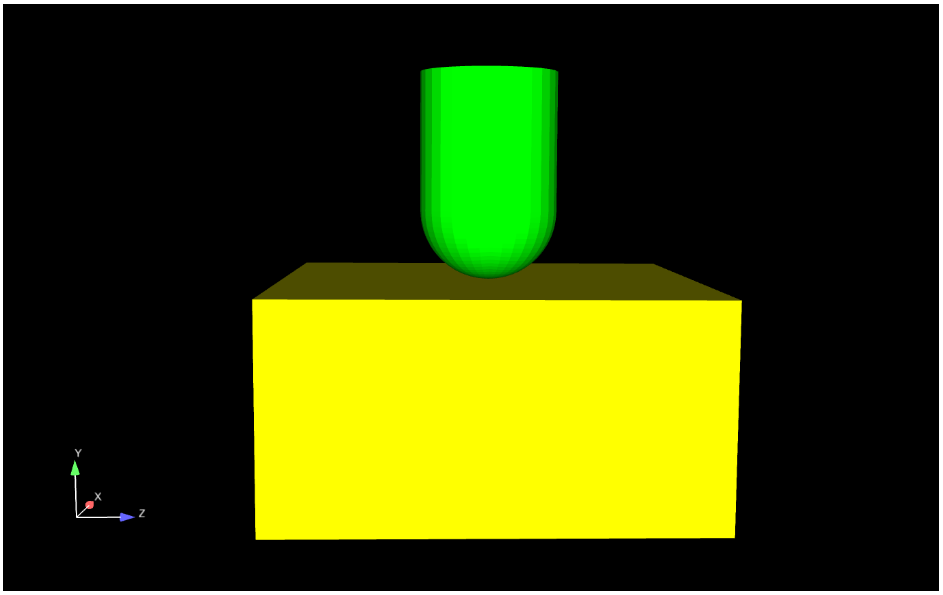


Figure 1.27: Thick plate indentation problem.

This test was originally created to replicate a problem from the ABAQUS Example Problem Manual. In the problem, a punch is given a displacement which creates a deep indentation into a thick, malleable plate. However, the size of the displacement could not be replicated so the problem was modified to incorporate only a fraction of the original displacement. The other adjustments to the ABAQUS problem are that instead of the punch displacing, it is now fixed, with the plate pushing up on the punch, and that the problem now tests the full geometry instead of a quarter of the total geometry. The initial configuration can be seen in Figure 1.27.

1.5.2 Loading and Boundary Conditions

This example problem contains very few and simple geometries. The bottom surface of the plate is given a prescribed displacement which pushes the plate up into the punch. In addition, the top surface of the punch is fixed in the y direction to ensure the contact between the plate and the punch causes deformation, not displacement. Lastly, a master slave relation is used to define contact between the punch and the plate, respectively.

1.5.3 Material Model

The plate uses an elastic material model and is given the properties of a crushable foam. The Young's Modulus for the punch is set very high to mimic a rigid body. Properties for the crushable foam were obtained from the ABAQUS manual.

1.5.4 Finite Element Model

The total model contains just under 170000 hex elements, with the plate containing 150000 and the punch containing almost 20000. Figure 1.28 shows the graded mesh used for the plate.

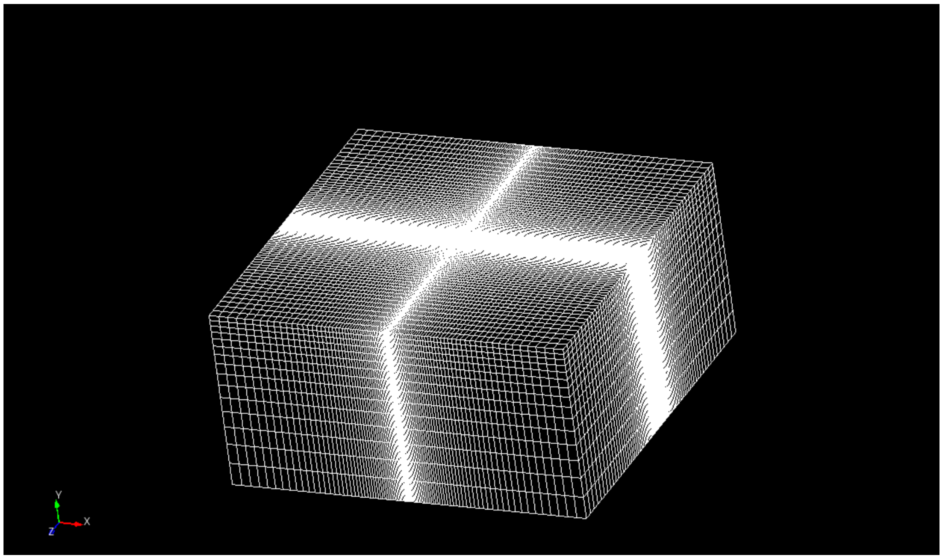


Figure 1.28: Graded Mesh

1.5.5 Feature Tested

The primary featured tested in this problem is implicit contact

1.5.6 Results and Discussion

Figure 1.29 shows the system at it's final step, displaying the maximum indentation the plate undergoes. Figure 1.30 displays 1/4 of the plate, which better shows the contact behavior. When behaving correctly, the plate will not show any wave patterns, and will be as smooth as the mesh allows it. A finer mesh will lend a smoother surface, eventually reaching a perfect curve as the as the mesh intervals go to zero. Figures 1.31 and 1.32 show the strain developed by the compression of the plate.

For input deck see Appendix A.5

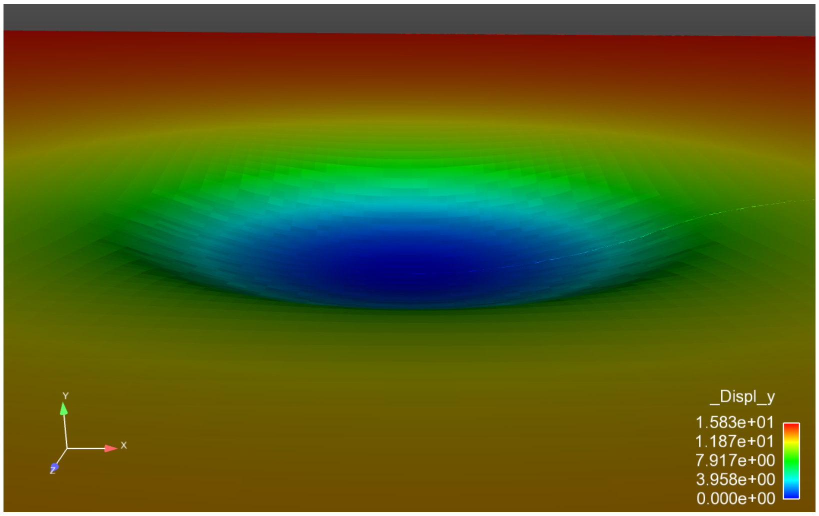


Figure 1.29: Final displacement.

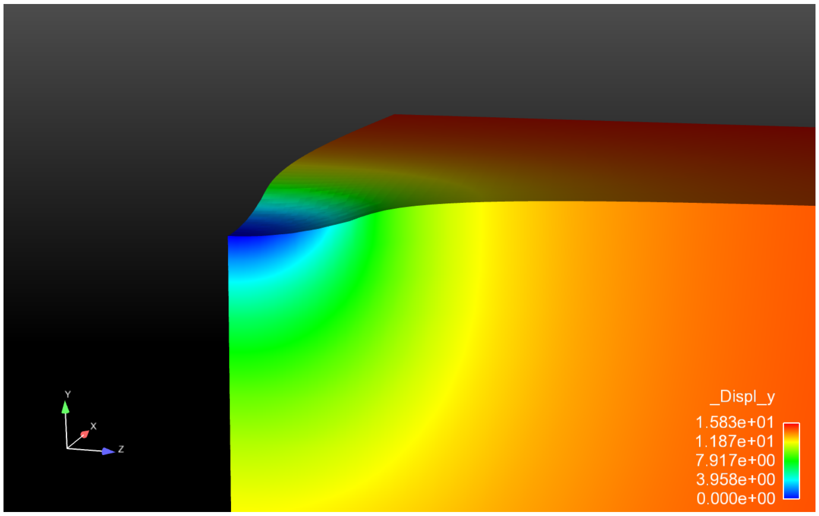


Figure 1.30: Final displacement.

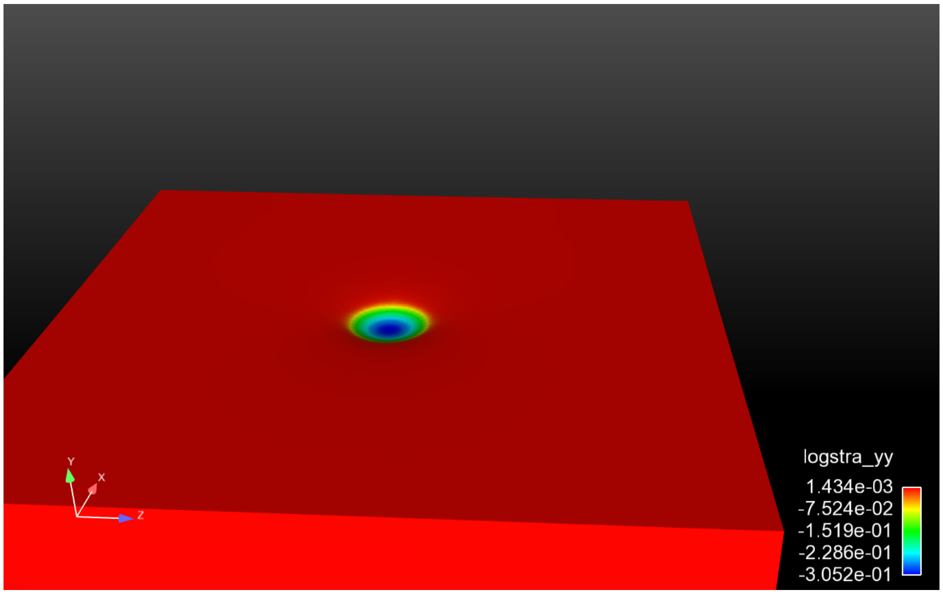


Figure 1.31: Final strain.

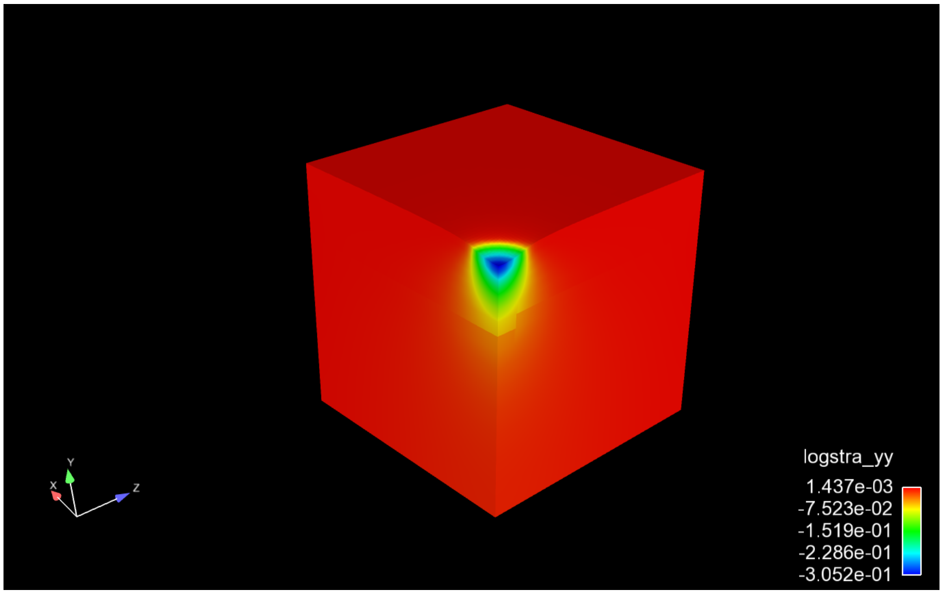


Figure 1.32: Final strain.

Chapter 2

XFEM

Warning: Support for XFEM in Sierra/SM is currently at an experimental level. As such, not all features may be fully implemented or tested and the analyst should use this capability with caution.

2.1 Angled Crack Cylinder

Product: Sierra/Solid Mechanics - Explicit Analysis

2.1.1 Problem Description

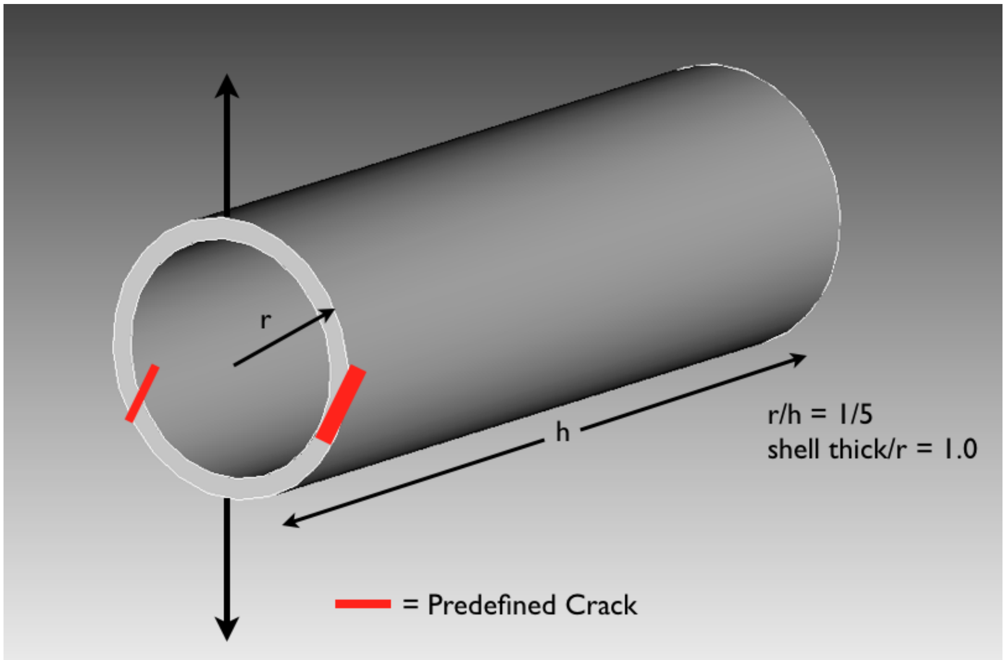


Figure 2.1: Angled crack cylinder problem set-up.

The purpose of the following numerical example is to display the X-FEM cutting by prescribed object and planar crack growth capabilities for 2-D shell elements. Consider a hollow cylinder under uniform tensile loading at the rim of the cylinder and an angled prescribed crack (Figure 2.1).

As the force increases as the top and bottom of the cylinder are pulled apart, a stress concentration forms at the crack tip. Once the stress concentration reaches the crack growth stress, the crack grows across that particular element. This problem is run using explicit dynamics.

2.1.2 Loading and Boundary Conditions

A prescribed displacement is applied to the top and bottom of the rim of the cylinder at a linear rate. The prescribed object used for cutting is a disk whose midpoint is placed at the end of the cylinder, halfway between the center of the cylinder and the bottom rim. The crack growth parameter that was used was chosen for this problem is maximum principal stress.

2.1.3 Material Model

An elastic-plastic material model is used, and the material properties can be found in Table 2.1 below.

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 2.1: Cylinder Material

Cylinder		
Young's Modulus:	E	1.0×10^9
Poisson's Ratio	ν	0.25
Density	ρ	2.61×10^-4
Yield Stress	σ_{yield}	36,000
Hardening Modulus	H	0.0
Beta	β	1.0

2.1.4 Finite Element Model

The elements that were used for this simulation were Belytschko-Tsay shell elements.

2.1.5 Feature Tested

X-FEM cut by prescribed object, and planar crack growth.

2.1.6 Results and Discussion

As can be seen in Figure 2.2 below, the prescribed crack grows when the user-specified maximum principal stress is reached. The crack propagates up the geometry in a planar fashion, and a sliver of the cylinder is cut off. As the displacement is applied, the geometry separates into two separate pieces. In Figure 2.3, the triangles over the geometry are not elements but visualization surfaces; the element used is still the Belytschko-Tsay four node shell.

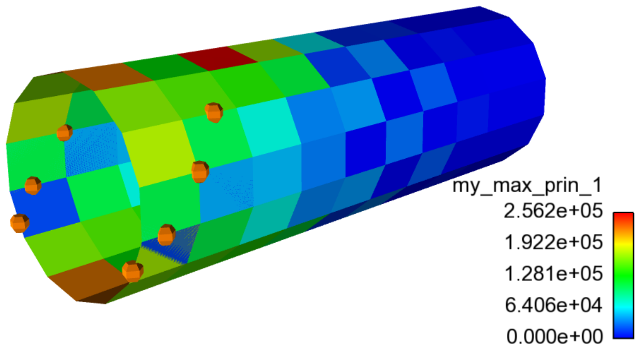


Figure 2.2: Planar crack growth.

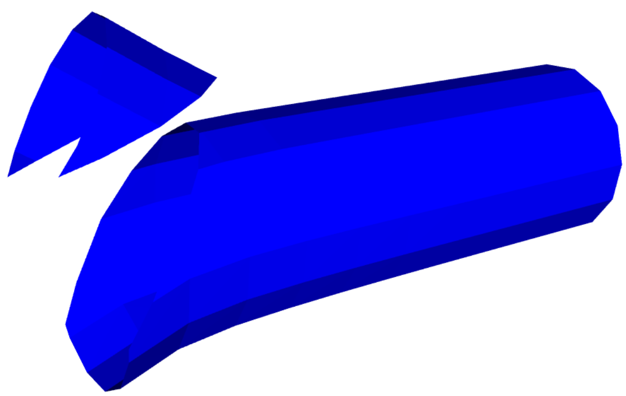


Figure 2.3: Piece of cylinder is cut off and separates.

For input deck see Appendix A.6.

2.2 Plate with Multiple Holes

Product: Sierra/Solid Mechanics - Explicit Analysis

2.2.1 Problem Description

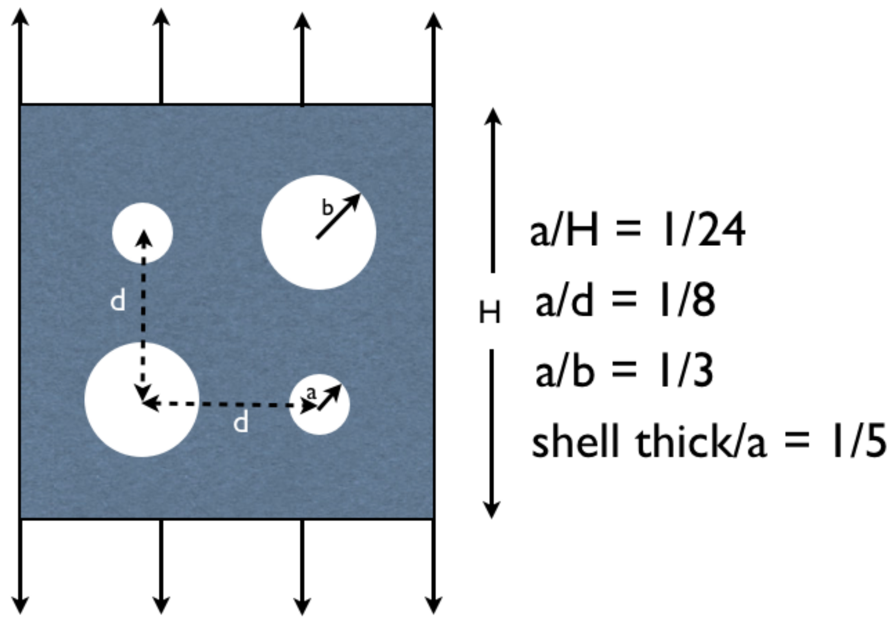


Figure 2.4: Plate with multiple holes problem set-up.

The purpose of the following numerical examples is to display the X-FEM nucleation and branching capabilities for 2-D shell elements. Consider a plate with a multiple holes under uniform tensile loading (Figure 2.4). As the force increases as the top and bottom are pulled apart, a stress concentration forms at the sides of both large holes. Once the stress concentration reaches the fracture stress, a crack nucleates at these locations. The cracks grow in multiple different modes, and the cracks branch when their user specified branching criteria is reached.

2.2.2 Loading and Boundary Conditions

A force is applied to the top and bottom of the plate at a linear rate. The nucleation method that was chosen for this problem is element based nucleation and the nucleation and branching failure parameter is maximum principal stress.

2.2.3 Material Model

An elastic-plastic material model is used, and the material properties can be found in Table 2.2 below.

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 2.2: Plate with Multiple Holes Materials

Plate with Hole		
Young's Modulus:	E	210×10^3
Poisson's Ratio	ν	0.3
Density	ρ	0.0078
Yield Stress		360
Hardening Modulus		50×10^3
Beta	β	0.75

2.2.4 Finite Element Model

The elements that were used for this simulation were Belytschko-Tsay shell elements.

2.2.5 Feature Tested

X-FEM crack nucleation, piecewise-linear crack growth, and crack branching in shells under dynamic conditions.

2.2.6 Results and Discussion

As can be seen in Figure 2.5 below, the stress concentration just before crack nucleation is the highest around the 2 large holes. Once crack nucleation is initiated, stress waves propagate throughout the geometry as shown in Figure 2.6. The cracks propagate in multiple different fashions, with other cracks nucleating in the geometry as well as crack branching occurring (Figure 2.7).

For input deck see Appendix A.7.

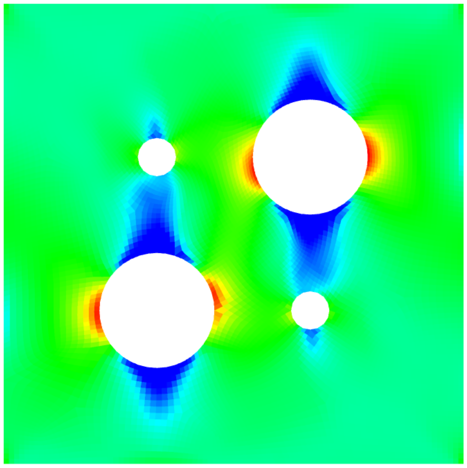


Figure 2.5: Multi holes before nucleation.

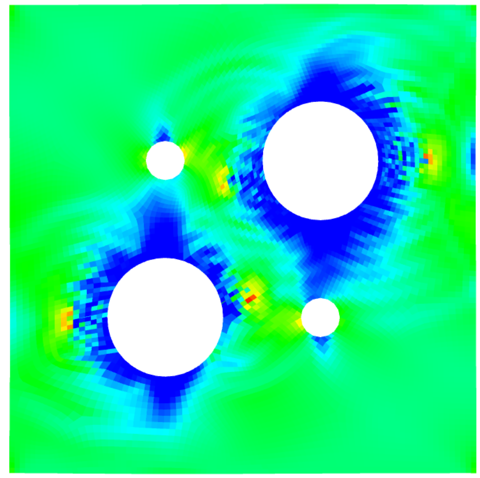


Figure 2.6: Stress waves after nucleation.

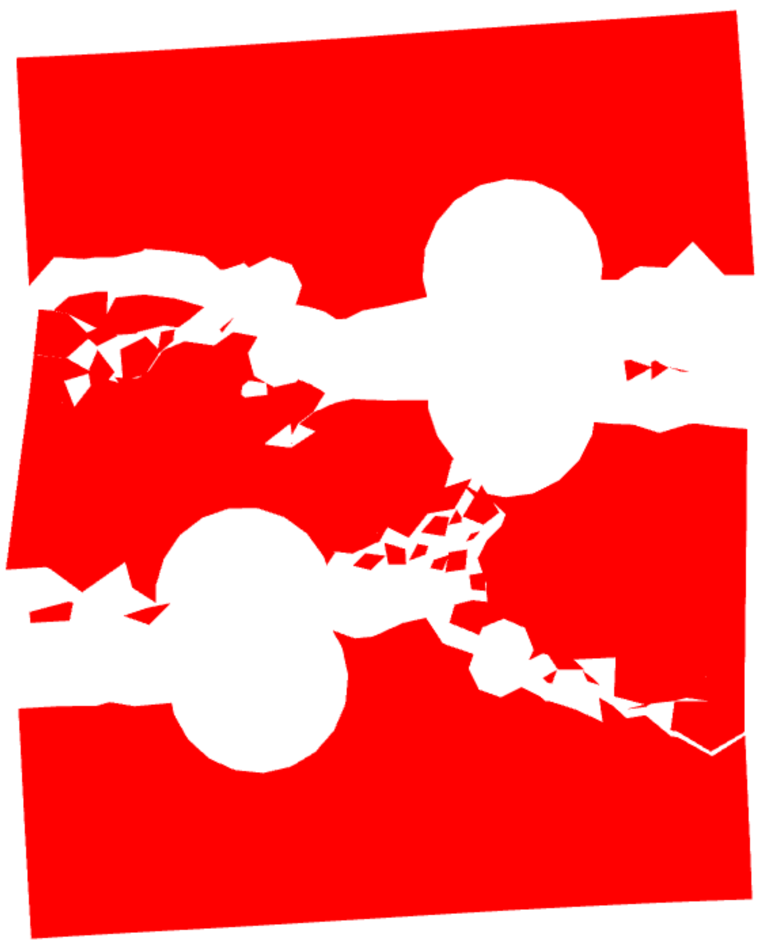


Figure 2.7: Plate with Multiple Holes Snapshots

Chapter 3

General/Other

3.1 Stress Strain Plate

Product: Sierra/Solid Mechanics - Implicit Analysis

3.1.1 Problem Description

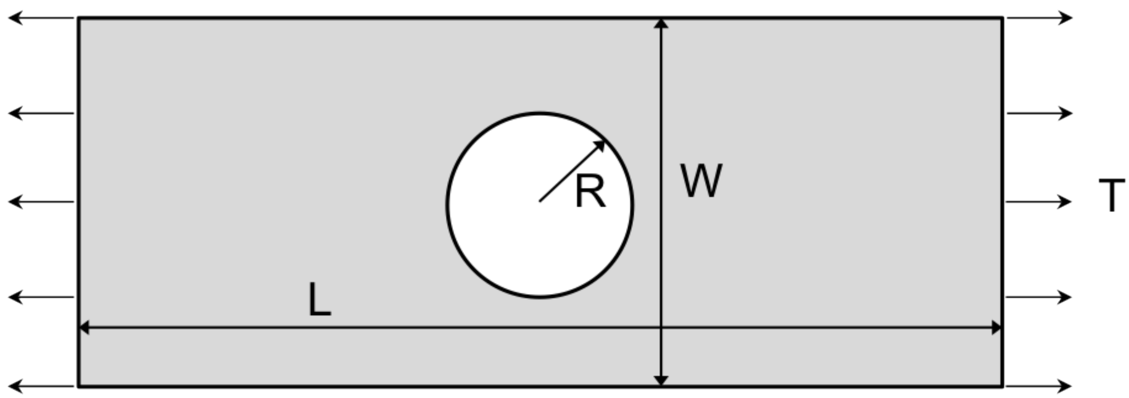


Figure 3.1: Plate with hole problem definition.

The purpose of this problem is to exemplify how to apply boundary conditions that approximate plane strain and plane stress conditions in a three-dimensional model. A plate with a hole under uniform tensile loading is considered (Figure 3.1) with length $L = 10.0$, width $W = 4.0$, variable thickness $2t$, and hole radius $R = 1.0$. The plane strain condition means the out-of-plane strain components are negligible, and the plane stress condition means the out-of-plane stress components are negligible. The former is representative of stresses at mid-thickness of a 'thick' plate and the latter is representative of stresses through the thickness of a 'thin' plate. This problem is run using implicit quasi-statics.

3.1.2 Loading and Boundary Conditions

For computational simplicity, only one-eighth of the plate is modeled (Figure 3.2). A tensile traction of magnitude $T = 1.0 \times 10^4$ is applied on the positive-x face of the plate, and symmetry

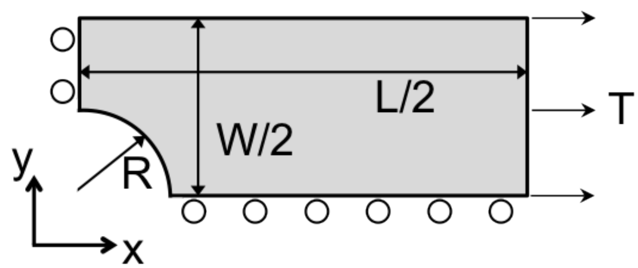


Figure 3.2: Plate with hole model.

boundary conditions are applied on the negative-x, negative-y, and negative-z faces of the plate. For the positive-z face of the plate, the boundaries conditions shown in Table 3.1, below, are considered. The intended out-of-plane behavior, plane stress or plane strain, is noted for each positive-z boundary condition.

Table 3.1: Plate with hole BC's on positive-z face

BC TYPE	DIRECTION	MAGNITUDE	OUT-OF-PLANE BEHAVIOR
Displacement	z	0.0	Plane Strain
Pressure	normal	0.0	Plane Stress
Traction	z	0.0	Plane Stress
Force	z	0.0	Plane Stress
Free DOF	all	—	Plane Stress

3.1.3 Material Model

The elastic material model given in Table 3.2, below, is used.

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

3.1.4 Finite Element Model

The elements used for all simulations are uniform gradient hexahedron elements. Two meshes, *mesh1* and *mesh2*, are considered (Figure 3.3). *mesh1* has a plate half-thickness $t = 0.05$; *mesh2* has $t = 0.025$ and approximately half the element size of *mesh1*. Both meshes have one element through the thickness direction, so the element size differs between the two meshes to maintain the same element aspect ratio. The plate thickness is decreased from *mesh1* to *mesh2* to evaluate the affect of thickness on the out-of-plane stresses.

Table 3.2: Plate With hole materials

Material Properties		
Young's Modulus:	E	200×10^9
Poisson's Ratio	ν	0.3
Density	ρ	1.0

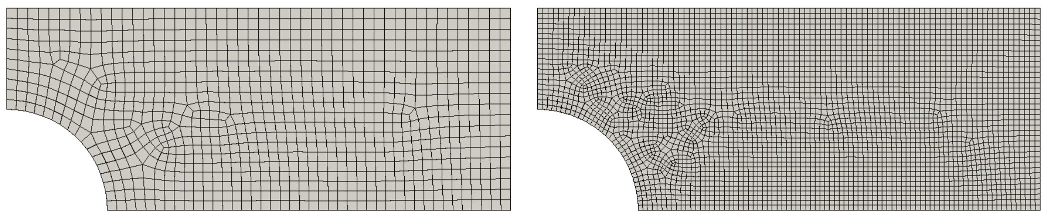


Figure 3.3: Plate with hole meshes.

3.1.5 Feature Tested

Plane stress/strain boundary conditions.

3.1.6 Results and Discussion

As can be seen in Figure 3.4, the plane strain condition is represented in both meshes by prescribing zero displacements in the out-of-plane direction (the z-direction). As can be seen in Figure 3.5, the accuracy of the plane stress increases as the plate thickness decreases, as expected. Each of the plane stress boundary conditions in Table 3.1 show a similar level of accuracy. For these plane stress approximations, the maximum absolute value of the out-of-plane stress is three orders of magnitude less than the applied traction, and negligible a sufficient distance from the hole.

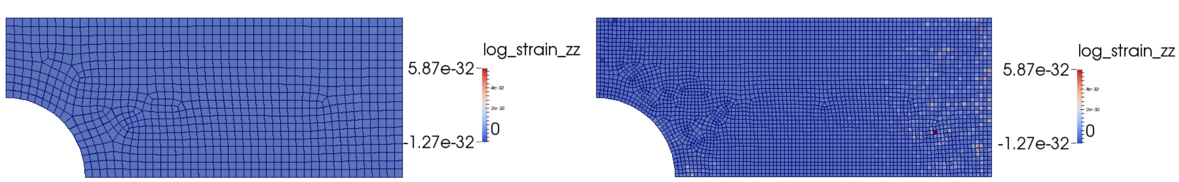


Figure 3.4: Plate with hole results for zero z-displacement prescribed on positive-z face.

For input deck see Appendix A.8.

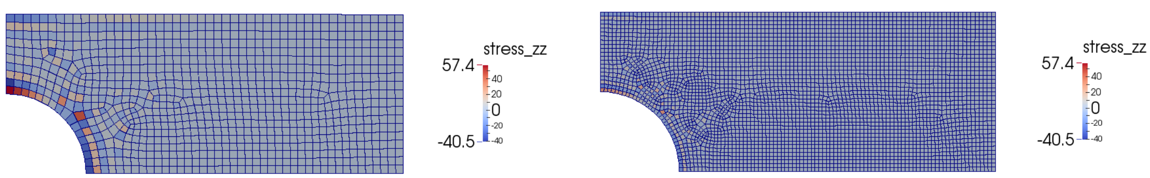


Figure 3.5: Plate with hole results for zero pressure prescribed on positive-z face.

3.2 Bolt Preload

Product: Sierra/Solid Mechanics - Implicit Analysis

3.2.1 Problem Description

This example demonstrates the process of preloading a bolt in four manners: thermal strain, artificial strain, prescribed displacement, and a spring. In reality a bolt and nut could be used to clamp components of a joint by the application of a preload. This could be applied through the shaft by a bolt head and nut, thereby bounding respective surfaces together. For simplicity, this example implements a single loading block, a bolt head, and a theoretical nut of matching dimensions. The loading block for these test cases can be seen in Figure 3.6 and the assembly diagrams can be seen in Figure 3.7.

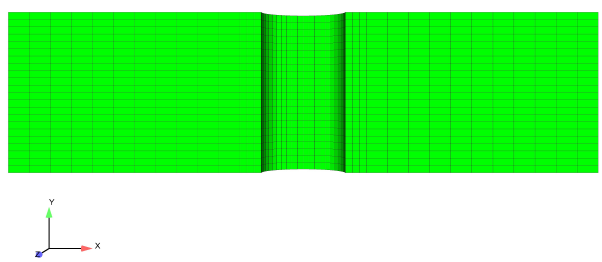
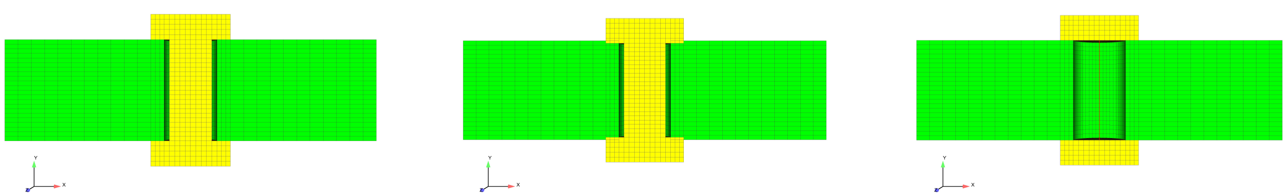


Figure 3.6: Loading Block for the Four Preloading Cases



(a) Thermal and Artificial Strain **(b)** Prescribed Displacement (mesh overlap) **(c)** Spring

Figure 3.7: Bolt Assembly Diagram for the Four Preloading Cases

In the first case, a preload is simulated through a thermal strain on a bolt. The entire bolt is cooled to -10 Kelvin at which point an orthotropic thermal engineering strain of 0.05 is applied to the bolted joint along the longitudinal axial direction of the bolt. Both ends of the bolt flanges lay flush with the joint. The isometric view of this preloading can be seen in Figure 3.8.

In the second case, a preload analysis is simulated by defining an artificial strain to a bolt. The bolt is prescribed an anisotropic strain aligned with the global X, Y and Z axes. Both ends of the bolt flanges lay flush with the joint. The isometric view of this preloading case is identical to that of the thermal case and can be seen in Figure 3.8.

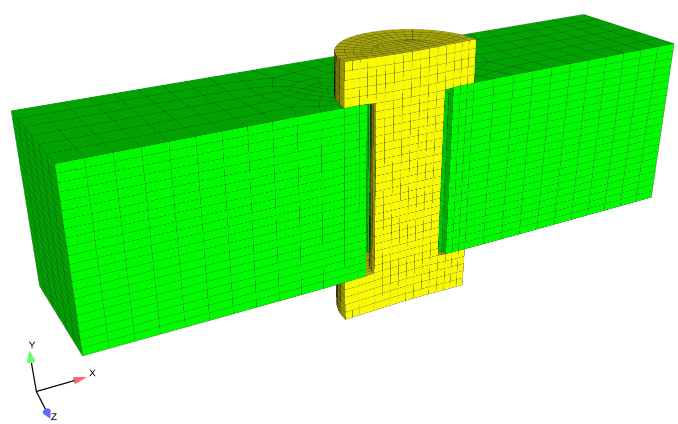
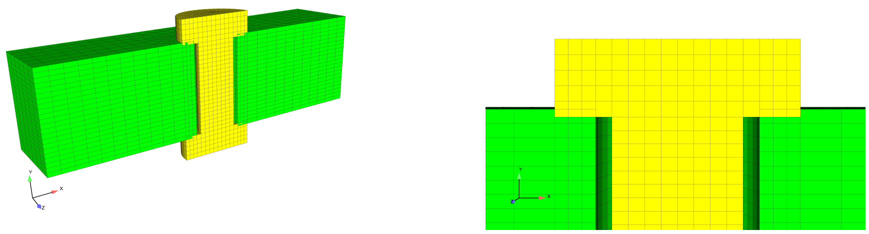


Figure 3.8: Preload test case: Thermal and Artificial Strain

In the third case, a prescribed displacement is applied to a bolt to simulate preload. The model is set-up in a 'stress free' condition with both ends of the bolt overlapping the joint by approximately half an element's length. With contact turned off on the bottom and top surfaces, the bolt is pulled into place using a prescribed displacement. After the bolt flanges are displaced enough to lay flush with the joint, contact on the top and bottom surfaces is turned on and the artificial force is released. The isometric view of this preloading case can be seen in Figure 3.9.



(a) Prescribed Displacement **(b)** Close Up Of Mesh Overlap

Figure 3.9: Preload test case: Prescribed

In the fourth case, a preload analysis is simulated by defining a spring section on a bolt. In this analysis the middle section of a bolt is replaced with a preloaded two node spring of equivalent cross sectional area to the thermal and prescribed displacement cases. The spring section can be adjusted to provided the desired preload in the bolt material. Both ends of the bolt flanges lay flush with the joint. The isometric view of this preloading case can be seen in Figure 3.10.

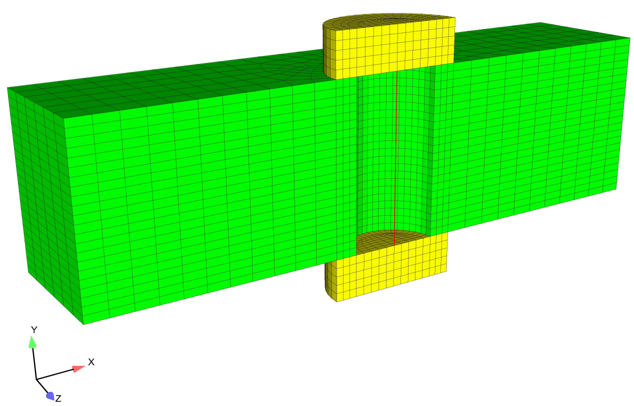


Figure 3.10: Preload test case: Spring

3.2.2 Loading and Boundary Conditions

In the prescribed displacement, thermal strain, and artificial strain cases a fixed displacement is set on the cross sectional front face of the bolt in addition to the front face of the joint block. This allows for a symmetric model. The upper and lower outer portion of bolt flanges are fixed in all directions. Master and Slave contact syntax is defined between the upper and lower inner flanges with the block.

The feti equation solver uses a damping coefficient of 0.0001 in the four load cases.

3.2.3 Material Model

In the thermal preload case, an orthotropic thermal strain field is applied in the material specification command block; the thermal strain is a general material property and not part of a constitutive model such as Elasticity. It is required that the orthotropic thermal strain in the X, Y, and Z be placed inside of material the command block. This example demonstrates thermal strain along the y-axis, setting the change in strain with decreasing temperature to zero along the x and z axis.

An anisotropic strain is applied in the artificial strain bolt preload condition. If desired, an isotropic thermal or artificial strain can be applied in the material specification command block. This would demonstrate equal physical properties along all axes.

The following material properties were used during analysis:

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 3.3: Bolt Materials

Bolt Preload		
Young's Modulus: Block and Bolt	E	200×10^9
Poisson's Ratio	ν	0.29
Density	ρ	7.89×10^3

3.2.4 Finite Element Model

For all four cases, the bolt and outer block implement a hex mesh, while the spring was created using a bar. Various webcuts are defined to obtain precise nodeset and block specifications across the different load cases.

3.2.5 Results and Discussion

Results obtained from the thermal, spring, prescribed displacement, and artificial strain bolt preload cases were calculated using input to allow similar analytical solutions. The σ_{yy}, σ_{xx}, and σ_{xy} stresses for the four loading conditions can be seen in Figures 3.11, 3.12, and 3.13. The σ_{yy} stress is aligned with the bolt axis.

In addition, mesh generation in the four cases were created to mimic geometries, yet also satisfy the appropriate loading conditions. For example, the central cross section of the thermal bolt was replaced with a two-noded bar and a representative rigid body cross sectional area in the spring case.

Table 3.4: Use Case Summary

Preloads	
Thermal	Well used for problems that are insensitive to temperature. Straightforward set up.
Artificial	Well used for problems that are sensitive to temperature. Straightforward set up.
Prescribed	A higher degree of setup difficulty, but most realistic if mesh overlap is present.
Spring	If a problem force is well known, it can be put directly into spring. Moderate set up.

For input deck see Appendix A.9.

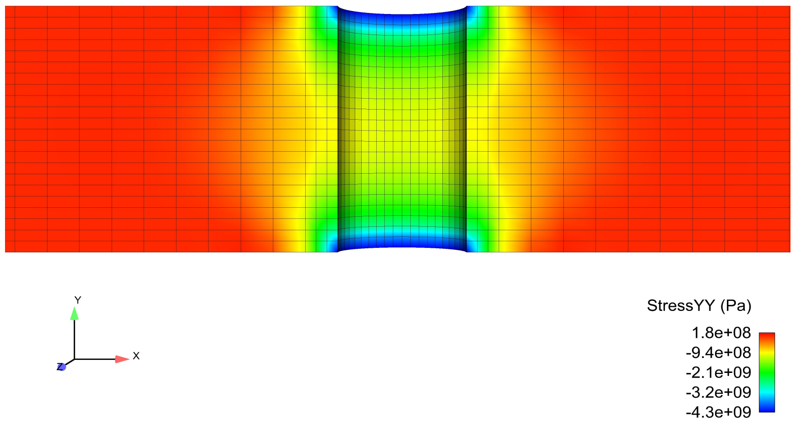


Figure 3.11: Bolt Preload: σ_{yy}

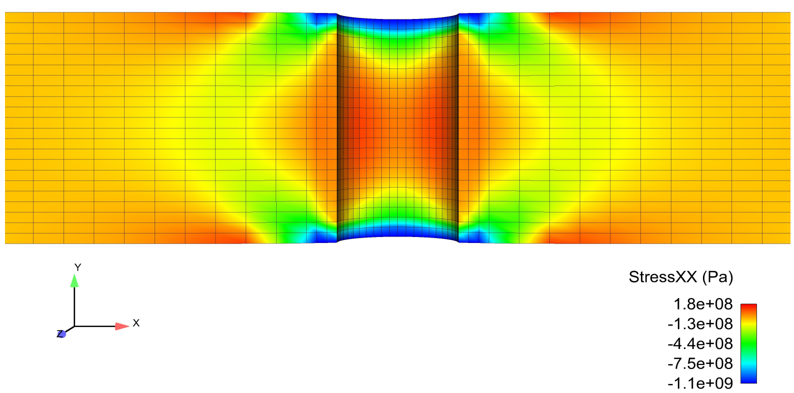


Figure 3.12: Bolt Preload: σ_{xx}

For input deck see Appendix A.9.

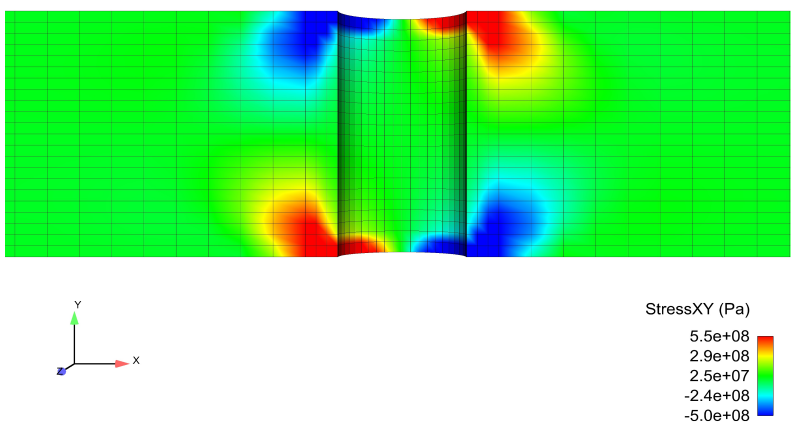


Figure 3.13: Bolt Preload: σ_{xy}

3.3 Automated Adaptive Preloading

3.3.1 Problem Description

This example demonstrates how an automated preload many be applied in an analysis to meet some specific target conditions. There are two analyses presented here. In the first analysis the artificial strain to achieve target clamping forces in a set of bolts is determined. In the second case the target force required to deform a nonlinear part a specified amount is determined.

3.3.2 Bolt Preload Problem

The mesh for the bolt preload analysis is shown in Figure 3.14. Each of the three bolts (blue, violet, and red) are are embedded in a fixture block (gray). An artificial strain will be applied to shrink a portion of each bolt (green). The purpose of the preload is to find the correct artificial strain such that each bolt has the correct target clamping force as would be produced by tightening the bolt with a calibrated torque wrench.

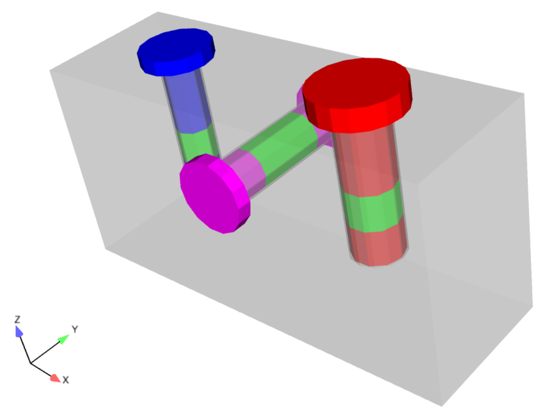


Figure 3.14: Bolt Preload Mesh

The material model for the bolt preload case is a simple elastic model. The bolts interact with the fixture block via contact. The bottom surface of the blue and red bolts are tied into the fixture block. All other contacts are frictional Coulomb contact.

The bolt model is loaded slowly in explicit transient dynamics in way to enable a mostly quasistatic response. A viscous damping block with a small velocity damping coefficient is used to damp out high frequency response and accelerate convergence to the quasistatic solution.

The artificial strain is solved for by combining a number of specialized capabilities with a library user subroutine. Ultimately the actual artificial strain is applied by two 'begin artificial strain' blocks. Blocks 201 (blue bolt) and 401 (red bolt) are artificially shrunk in the z direction while block 301 (violet bolt) is shrunk in the y direction. The actual shrinkage is controlled by the function 'bolt_preload' which is a simple function that just sets the artificial strain to the element value of the variable 'applied_strain'.

The per element applied strain is defined by a set of user output blocks each running a 'aupst_preload_solver' subroutine. The 'aupst_preload_solver' subroutine is described in more detail in the Sierra/SM user manual chapter on the user subroutine library. Effectively this subroutine will adaptively update the element variable 'applied_strain' until the target criteria is matched.

In this case the target criteria is the global value of the internal force in each bolt (found by the magnitude of the internal reaction within the bolt.) The subroutine takes several parameters. The 'target_value' parameter is the target axial preload force in the bolt. The 'initial_guess' parameter is the an initial guess of the strain required to reach the target force. The closer the initial guess is to the final correct value the faster the solution will be reached. Generally the initial guess should be on the low side to avoid accidentally overshooting the correct solution and causing yield. The 'iteration_time' parameter controls how long each 'load step' of the preload solver predictor corrector algorithm will be. The iteration time must be large enough that system is able to obtain at least approximate equilibrium within each iteration step. The 'target_variable' parameters defines the variable that drives solution. The preload is complete when the value of the 'target_variable' reaches the 'target_value'. The 'working_variable' parameter defines where the output of the subroutine will be stored. For this example the subroutine is setting the 'applied_strain' variable on each element to be read by the artificial strain block. Finally the user subroutine requires some persistent state data to function 'bolt_preload_state' which is defined an a 'begin user variable' command block.

3.3.2.1 Results and Discussion

The resultant forces in the bolts as a function of time are shown in Figure 3.15. It can be seen that the bolt forces asymptote up to the correct target values. Additionally the load, evaluate, update cycles of produced by the aupst_preload_solver subroutine are apparent.

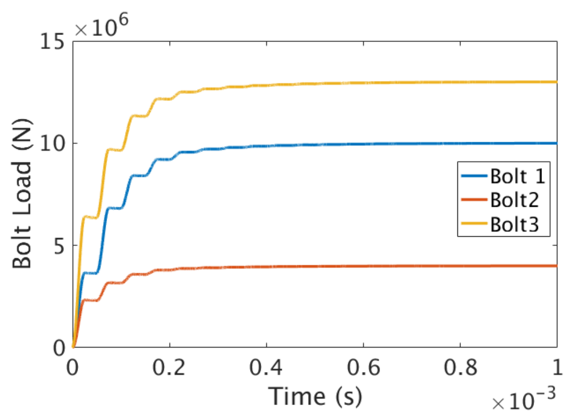


Figure 3.15: Bolt Preload Results

3.3.3 Wishbone Problem

The mesh for the wishbone preload analysis is shown in Figure 3.16. The purpose of the preload is to find the correct pin forces such that the body is stretched to the correct pin-to-pin length so that it can be placed on another component.

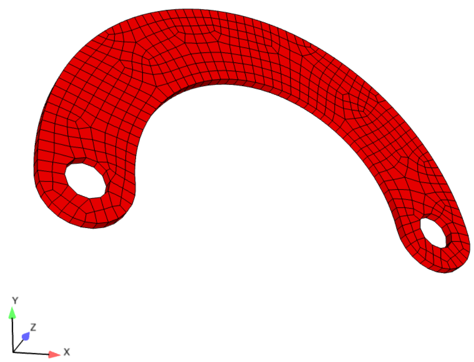


Figure 3.16: Wishbone Preload Mesh

The material model for the bolt preload case is a elastic plastic. It is expected that the preload will yield the material making this problem very non-linear.

For this case the model is solved with implicit statics. A handful of additional symmetry constraints are applied to ensure the model remains statically determinate.

As in the bolt preload case the artificial strain is solved for by combining a number of specialized capabilities with a library user subroutine. Ultimately the actual pin force is applied by two 'begin distributed force' blocks which apply equal and opposite -X forces to the left pin hole and +X forces to the right pin hole. The distributed force blocks access the function 'solved_force' which just sets the net force on each pin equal to a global variable that will be defined by an 'aupst_preload_solver' subroutine.

In this case the target variable for the preload solver is the net displacement between the left and right pin holes and the 'initial_guess' is a guess of the required force to achieve the displacement. As in the bolt case this initial guess needs to be in the right ball park, but should generally be a low-ball estimate to avoid overshooting the actual solution and causing excessive yield. As in the bolt preload case the user subroutine requires an externally defined state variable field. The only difference in the wishbone case is the user subroutine is operating on global quantities (in the bolt case the user subroutine was operating on element quantities.)

3.3.3.1 Results and Discussion

The forces applied to the pin holes as a function of time are shown in Figure 3.17 and the resultant displacement in Figure 3.18. It can be seen that the pin-to-pin displacement asymptotes to the correct value. Finally the force displacement curve (applied_force vs. curDisp) for the wishbone is plotted in Figure 3.19. The force displacement curve shows the non-linear response of the wishbone and that the preload is effectively terminated at the correct displacement value.

For input deck see Appendix A.10

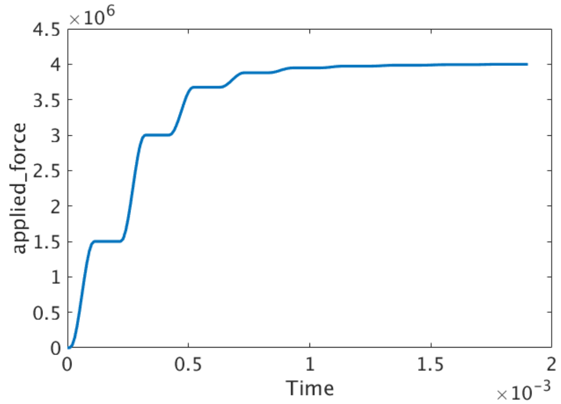


Figure 3.17: Wishbone Force Results

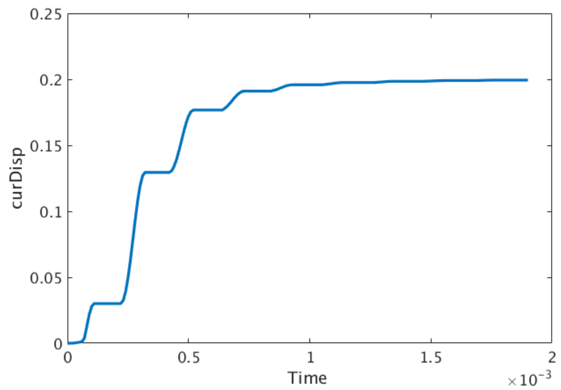


Figure 3.18: Wishbone Displacements Results

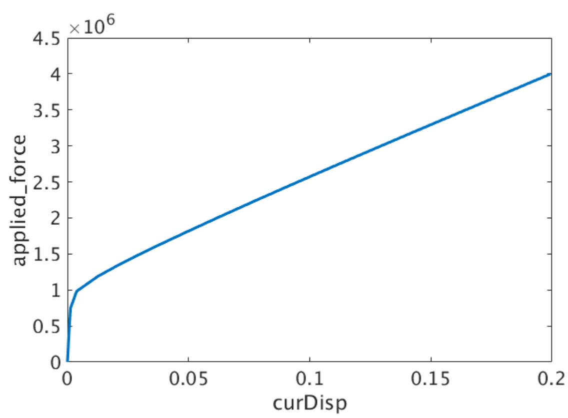


Figure 3.19: Wishbone Force Displacement Curve

3.4 Overlap Removal Methods

3.4.1 Problem Description

This example demonstrates the process of removing overlap from two rings using two different methods: overlap removal and artificial stain coupled with general contact.

In the first case, the two rings have a small overlap due to the inner ring being slightly larger than the inner radius of the outer ring. This overlap removal block will be placed directly into the contact block. In figure 3.20 the left side is showing the model before the overlap is removed while the right is showing the resulting model after the removal.

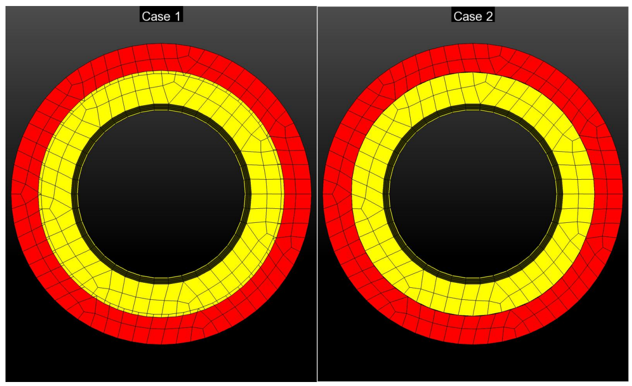


Figure 3.20: Small overlap and results after overlap removal

In the second case, a large overlap will be removed from the rings using an artificial strain in the radial direction for the first time period as shown in figure 3.21. Then contact is activated and the artificial strain is removed as you can see on the right side of the plot in figure 3.21. In figure 3.22 the left side is showing the model before anything is done and the right is showing how the model will look after the removal method.

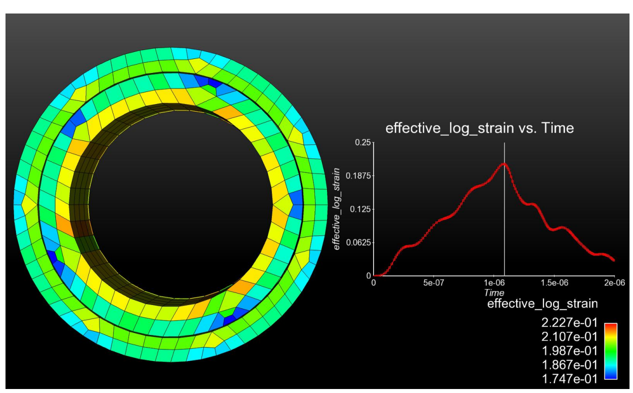


Figure 3.21: Rings under strain with strain vs time plot

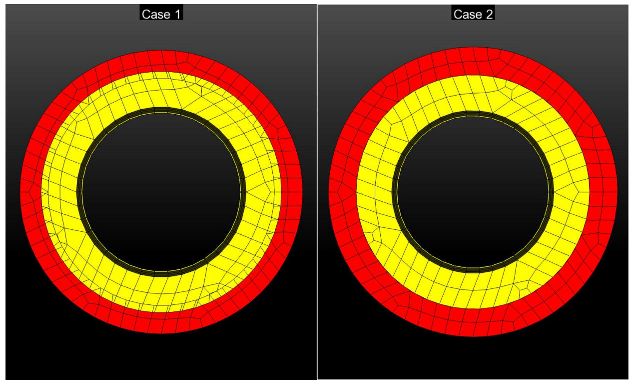


Figure 3.22: Large overlap and results after overlap removal method

3.4.2 Boundary Conditions

No boundary conditions where applied to these models.

3.4.3 Material Model

In both of these cases an elastic orthotropic material model was used with isotropic properties. This material model was used so that the material would have a defined cylindrical coordinate system so that the strain can be applied in the radial direction.

The following material properties were used during analysis:

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 3.5: Ring Material

Overlapping Rings		
Young's Modulus: Block and Bolt	E	64×10^9
Poisson's Ratio	ν	0.20
Density	ρ	0.5

3.4.4 Finite Element Model

For the small overlap case the inside ring was made to have an outside radius of 0.205 while the outer ring has an inner radius of 0.20. For the larger overlap the case the inside ring has an outer radius of 0.2125 while the outer ring has the same inner radius. Both of these cases were created using a simple hex mesh.

3.4.5 Results and Discussion

Results obtained from the overlap removal case showed that no stress or strain was applied to the system in order to remove the overlap, however this method could handle more than an overlap of 50 percent of your smallest element. In the case where strain was applied larger amounts of overlap could be removed, however a resulting stress and strain value is added to the system causing the rings to change from their original forms. In figure 3.23 the max principle stress of the resulting model is zero, while in figure 3.24 both of the rings are experiencing large values of stress.

Table 3.6: Use Case Summary

Methods	
Overlap Removal	Simple to use for small amounts of overlap. Straightforward set up.
Artificial Strain and General contact	Works for large overlap but leaves a stress and deformation of original setup. Moderate set up.

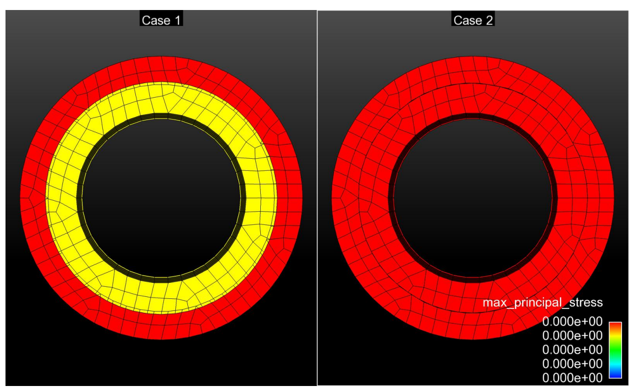


Figure 3.23: Stresses experienced after overlap removal

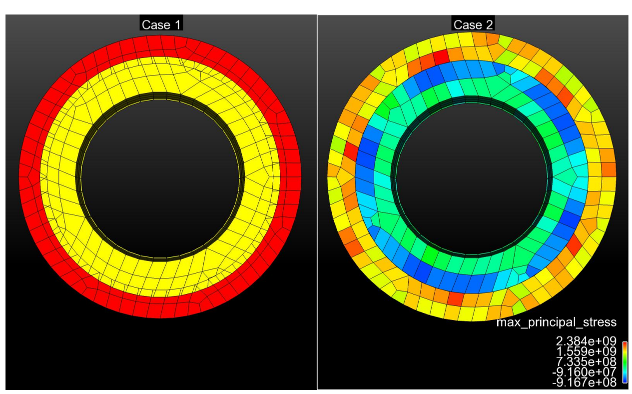


Figure 3.24: Stresses experienced after strain and contact is applied

For input deck see Appendix A.11

3.5 Remeshing

3.5.1 Problem Description

This example demonstrates the process of remeshing a part as the elements are experiencing abundant stretching. A bar with a very slight initial taper is pulled in tension to induce localized stretching, *i.e.* necking.

3.5.2 Boundary Conditions

For simplicity, the model is simulated with only one eighth of the bar. A fixed displacement is applied to the bottom and the interior faces to represent symmetry, and a prescribed velocity is applied to the top surface of the bar.

3.5.3 Material Model

The BCJ_MEM material model is used along with the metric units shown below.

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: newtons
Temperature: Kelvin

3.5.4 Finite Element Model

There is currently no automated method for remeshing a part as the elements are experiencing abundant stretching. In order to fix this problem a multistep process is used in the input deck with a termination time based off the global max eqps times the run number. Restart data is stored on every load step and read back into the old mesh after remeshing in order to transfer variables from this old mesh to the new mesh. The new mesh is created by reading the current exodus file of the old mesh into Cubit with applied deformations at the final timestep. This old mesh is then deleted and remeshed to remove stretching elements and saved as the next mesh in the sequence to prevent overwriting data. The input deck is then be altered to run with the next mesh in the series and to have a start time equal to the ending time of the previous run so that the restart output is used from the previous run. Exodus files for each mesh in the sequence are saved with unique names so that the entire simulation is stored for post-processing. This process is repeated multiple times until the part is completely stretched.

3.5.5 Results and Discussion

If remeshing is not performed for this simulation, the elements will overstretch and invert as seen in figure 3.26 below. This is fixed in figure 3.27 below, in which remeshing captures the proper necking as the strain increases. The remeshing will also help improve the way that localized softening is simulated as seen in figure 3.28. It can also be seen in figure 3.29 that the eqps is increasing after each run showing that all of the variables are being transferred from run to run. An L_2 projection transfer is performed from the current configuration (current coordinates) of the old

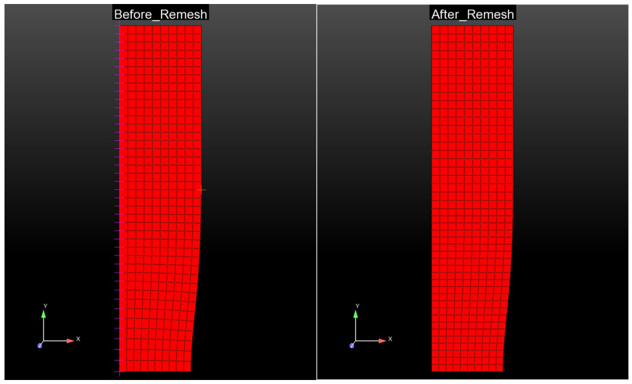


Figure 3.25: Mesh before recreation and after

mesh to the original configuration (model coordinates) of the new mesh to effectively update the reference frame after every remesh. This is necessary for improved accuracy when modeling large deformations such as this simple example.

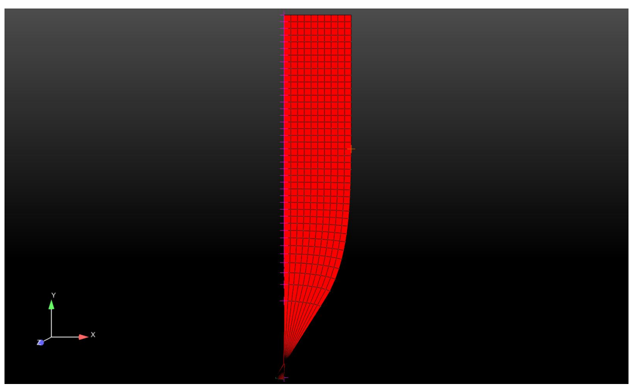


Figure 3.26: Mesh after stretching without remeshing

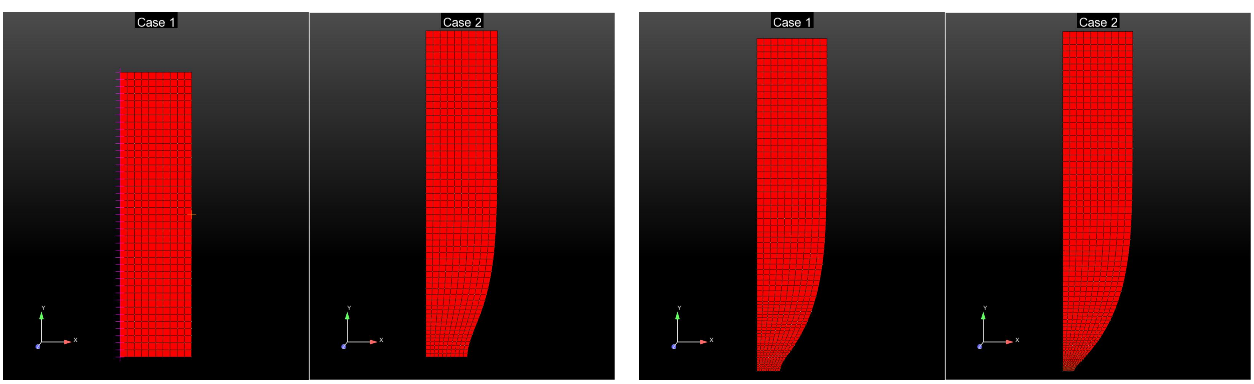


Figure 3.27: Different meshes throughout this process

For input deck see Appendix A.12.

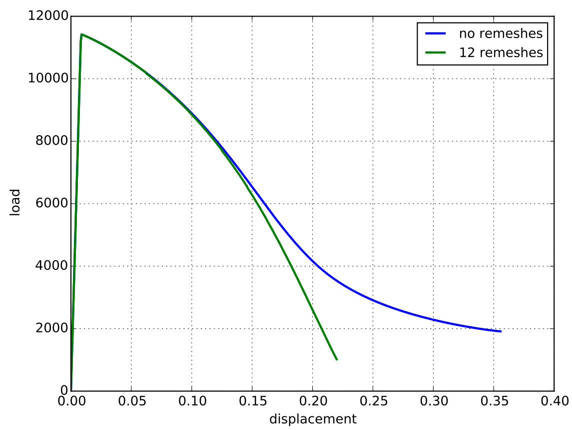


Figure 3.28: Displacement vs Load plot 12 remeshing and no remeshing

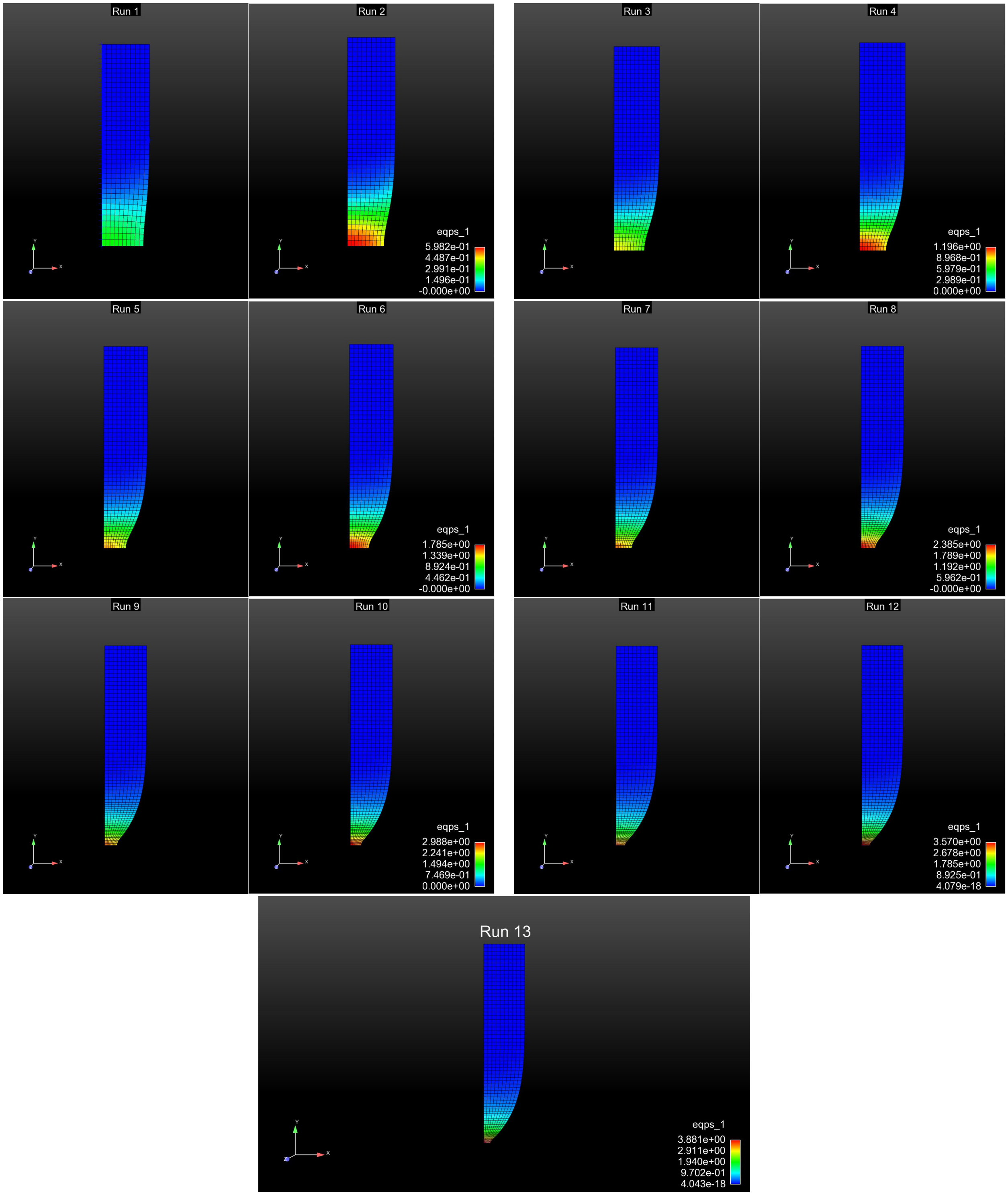


Figure 3.29: Meshes after each run with eqps values showing

3.6 Frame Indifference

Product: Sierra/Solid Mechanics - Implicit Analysis

3.6.1 Problem Description

The Frame Indifference Test requires the constitutive model to be self-consistent under superimposed rigid rotation and translations. This feature is tested through applying a uni-axial artificial strain to a single cube-shaped element, followed by an arbitrary rotation.

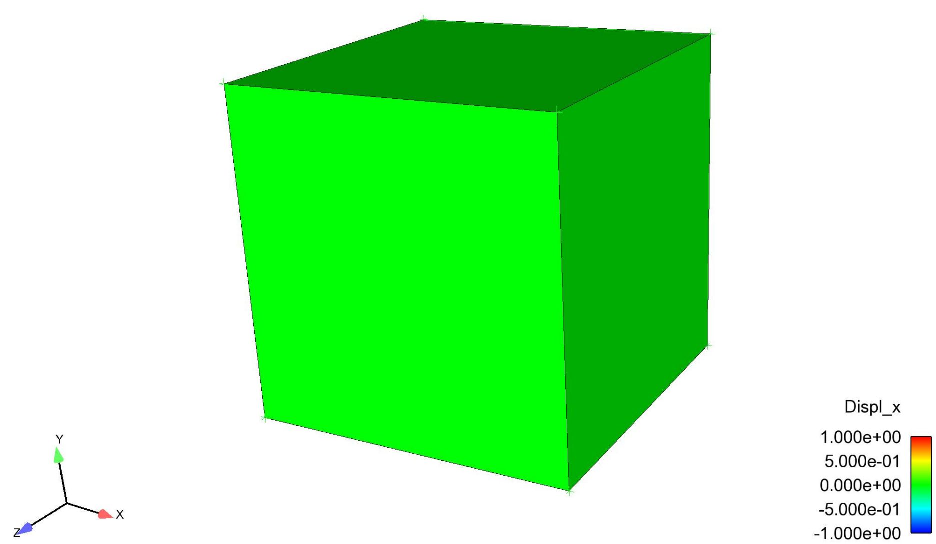


Figure 3.30: Initial Configuration

3.6.2 Loading and Boundary Conditions

Initially, the block is given an artificial strain in the positive X direction. During the application of artificial strain all nodes are fixed from moving in the X, Y, and Z directions, which creates an internal element stress. Once the artificial strain reaches its maximum value, the block rotates 90° about the Z axis. Rotation is achieved with a prescribed velocity function using a cylindrical axis; the fixed displacement boundary condition remains on all nodes during rotation. This prevents the block from deforming, but not from rotating. Figure 3.31 shows the block halfway through its rotation and Figure 3.32 shows the complete 90° rotation.

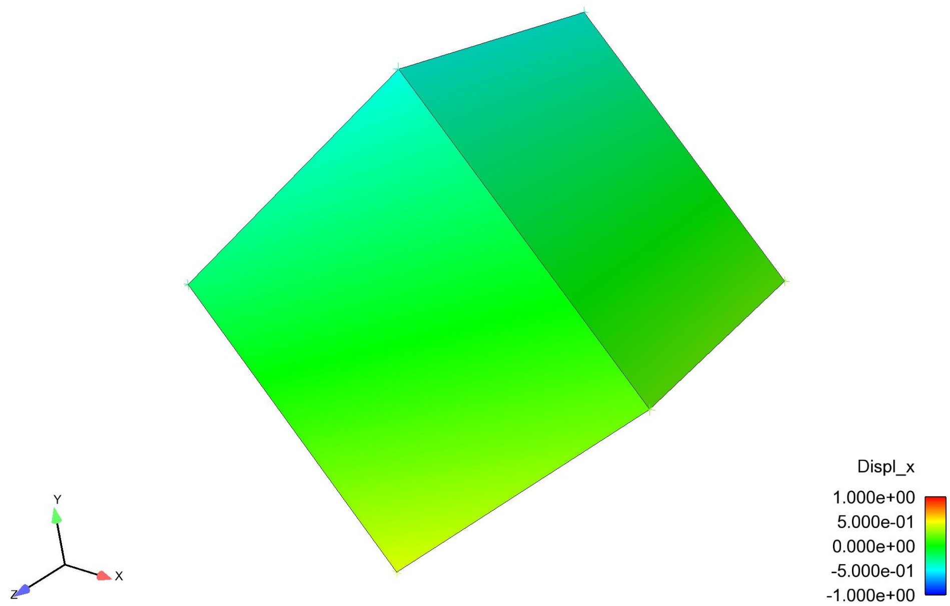


Figure 3.31: Midpoint of Block Rotation

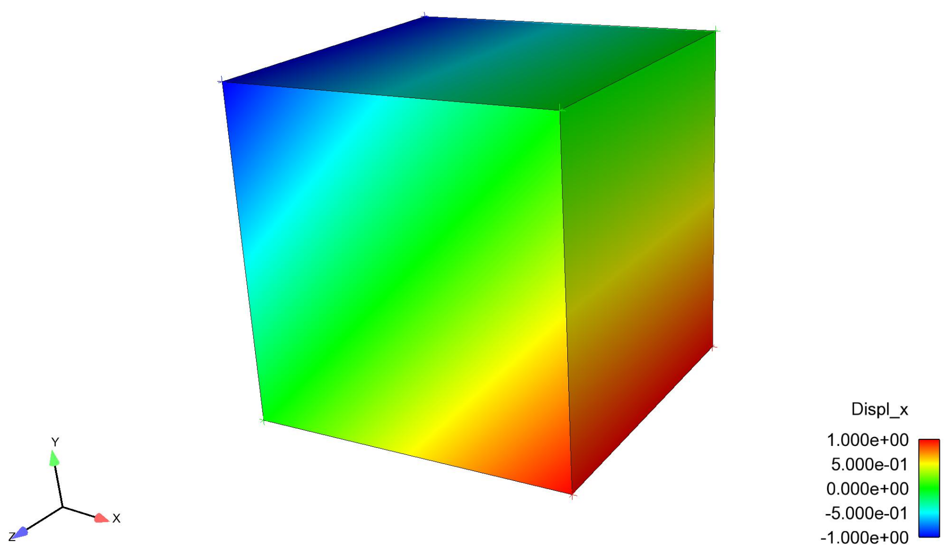


Figure 3.32: Deformed Element after 90° rotation about the z-axis

3.6.3 Material Model

This test contains an elastic-plastic model, and is given material properties similar to steel.

The following material properties were used during analysis:

Metric units are used:
Displacement: meters
Mass: kilograms
Time: seconds
Force: kgm/s^2
Temperature: Kelvin

Table 3.7: Material of Element

Material Properties		
Young's Modulus	E	200×10^9
Poisson's Ratio	ν	0.33
Density	ρ	7.871×10^3
Yield Stress	σ_{yield}	2.76×10^8

3.6.4 Finite Element Model

This test contains a single hex element.

3.6.5 Feature Tested

The primary features tested in this problem are the artificial strain and cylindrical rotation features, while verifying frame indifference.

3.6.6 Results and Discussion

This problem, implementing superimposed rotations applied to the spatial domain, provides consistent results to the the non-rotating uniaxial strain problem. Accordingly, in-plane principle stress components are accounted for during the rotation portion of the analysis. This verification test was adopted from, "Verification tests in solid mechanics: Engineering With Computers Volume:29 Number:4" (K.Kamojjala et al. 2013).

For input deck see Appendix A.13.

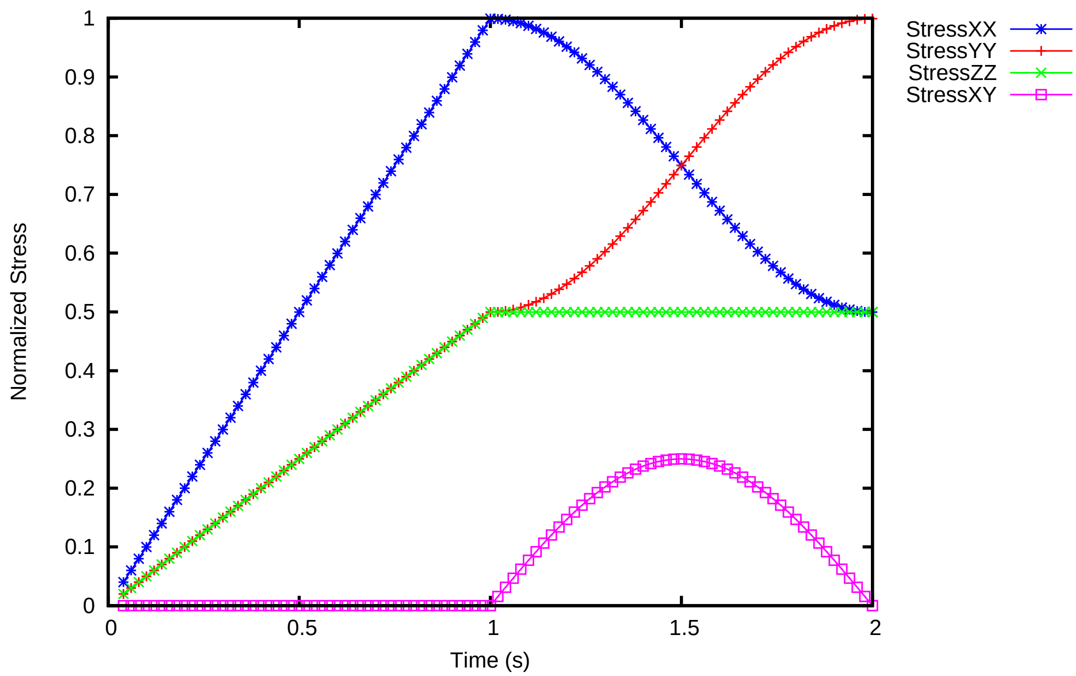


Figure 3.33: Normalized Stress Plot

Appendix A

Input Decks For Example Problems

We provide current versions of input decks that can be used to run the example problems discussed above. In a few cases, parameters have diverged from those used to produce the plots in the main body of this document. However, those input decks do produce results that are illustratively similar. Adjusting parameters to increase similarity is left as an exercise for the user.

A.1 Newton Cradle 1.1

```
begin sierra newtoncradlefinal

  # DEFINE FUNCTIONS,  N/m

  begin function gravity_accel
    type is constant
    begin values
      1.0
    end
  end

# DEFINE DIRECTION

  define direction X with vector 1.0 0.0 0.0
  define direction Y with vector 0.0 1.0 0.0
  define direction Z with vector 0.0 0.0 1.0

# DEFINE MATERIALS:STEEL, E= 200E9 N/m2 = 200E9 Pa , Steel=7840 kg/m3

  begin material balls_outer
    density = 7.48e3
    begin parameters for model elastic
      #youngs modulus = 200.0e9
      youngs modulus = 200e5
      poissons ratio = 0.3
    end parameters for model elastic
  end material balls_outer

  begin material wireANDinnersphere
    density = 7.48e3
    begin parameters for model elastic
      youngs modulus = 100.0e0
      poissons ratio = 0.3
    end parameters for model elastic
  end material wireANDinnersphere

  begin rigid body RB1
    reference location = 0 15 0
    include nodes in nodeset_301
```

```
  end
  begin rigid body RB2
    reference location = 2 15 0
    include nodes in nodeset_302
  end
  begin rigid body RB3
    reference location = 4 15 0
    include nodes in nodeset_303
  end
  begin rigid body RB4
    reference location = 6 15 0
    include nodes in nodeset_304
  end
  begin rigid body RB5
    reference location = 8 15 0
    include nodes in nodeset_305
  end

  begin truss section wiredup1
    area = 0.1
  end truss section wiredup1

# DEFINE FEM MODEL
#Note: Comment out gensesis files for desired test geometery configuration(1, 2, or 3 balls dropped)

  begin finite element model mesh
    Database Name =   newtoncradle_final_1balldrop.g
#   Database Name =   MESH/newtoncradle_final_2balldrop.g
#   Database Name =   MESH/newtoncradle_final_3balldrop.g
    Database Type = exodusII

# DEFINE BLOCKS(block set 1 series are inner volumes, series 100 are outer volumes, and 300 are set
# of curves/truss)

    begin parameters for block block_101 block_102 block_103 block_104 block_105
      material = balls_outer
      model = elastic
    end parameters for block block_101 block_102 block_103 block_104 block_105

    begin parameters for block block_1 block_2 block_3 block_4 block_5
      material = wireANDinnersphere
      model= elastic
    end parameters for block block_1 block_2 block_3 block_4 block_5

    begin parameters for block block_300
      material = wireANDinnersphere
      model = elastic
      section = wiredup1
    end parameters for block block_300

  end finite element model mesh

# DEFINE PROBLEM TIME AND TIME STEP PARAMETERS

  begin presto procedure myProcedure
    begin time control
      begin time stepping block p0
        start time = 0.0
        begin parameters for presto region myRegion
        end parameters for presto region myRegion
      end time stepping block p0
      termination time = 9
    end time control

# DEFINE BOUNDARY CONDITIONS
# ONLY DEFINE RIGID BODY AS FIXED DISPLACEMENT DUE TO REFERENCE LOCATION ON RIGID BODY (do not want
# entire nodeset rigid)
```

```
  begin presto region myRegion

    use finite element model mesh

    begin fixed displacement
      rigid body = RB1
      components = x y z
    end
    begin fixed displacement
      rigid body = RB2
      components = x y z
    end
    begin fixed displacement
      rigid body = RB3
      components= x y z
    end
    begin fixed displacement
      rigid body = RB4
      components= x y z
    end
    begin fixed displacement
      rigid body = RB5
      components = x y z
    end

    begin fixed rotation
      rigid body = RB1
      components= x y
    end
    begin fixed rotation
      rigid body= RB2
      components= x y
    end
    begin fixed rotation
      rigid body = RB3
      components= x y
    end
    begin fixed rotation
      rigid body = RB4
      components= x y
    end
    begin fixed rotation
      rigid body = RB5
      components= x y
    end

    begin gravity
      function = gravity_accel
      direction = y
      gravitational constant = -9.81
    end gravity

# DEFINE PROBLEM REGIONS

    begin contact definition contacts
      skin all blocks = on exclude block_300
      search = dash
      begin interaction defaults
        friction model = sticky
        general contact = on
        self contact  = off
        constraint formulation = node_face
      end
      BEGIN CONSTANT FRICTION MODEL sticky
        FRICTION COEFFICIENT = 0.3
      END
    end contact definition contacts
```

```
Begin Heartbeat Output normalized
    Stream Name = NewtonCradle.csv
    Format = SpyHis
    Start Time = 0.0
    At Time 0 Increment = 0.005
    Termination Time = 9.0
    Global kinetic_energy as KineticEnergy
    Global External_energy as ExternalEnergy
    Global Internal_energy as InternalEnergy
    Global SE as StrainEnergy
    global normIESE as Normalized_Dissipation
    global PEabs_negEEplusMaxEE as PEabs_negEEplusMaxEE
    global AbsTotalE as AbsTotalE
    global PE as PE
    global contact_energy as contact_energy
    global TotalE as TotalE
    global hourglass_energy as hourglass_energy

    global Kinetic_energyKJ as Kinetic_energyKJ
    global External_energyKJ as External_energyKJ
    global Internal_energyKJ as Internal_energyKJ
    global Potential_energyKJ as Potential_energyKJ
    global TotalE_KJ as TotalE_KJ

    global Sum_SE_EE_KE as Sum_SE_EE_KE
    global Sum_IE_EE_KE as Sum_IE_EE_KE
    global Et_Minus_E as Et_Minus_E
    global normEnergy as normEnergy
    global energyNorm as energyNorm
    global DissEnergyOVERKEo as DissEnergyOVERKEo
    global Dissipation_energy as Dissipation_Energy



   # global  max_Sum_KE_IE as max_Sum_KE_IE
   # global  max_Sum_EE_KE_PE as max_Sum_EE_KE_PE
   # Global MaxKineticEnergy as Max_Kinetic_Energy
   # Global MaxExternalEnergy as Max_External_Energy
   # Global MaxInternalEnergy as Max_Internal_Energy
  End Heartbeat Output normalized

 Begin User Output
   compute global SE as sum of element strain_energy
   compute global Sum_SE_EE_KE from expression "kinetic_energy+external_energy+SE"
   compute global Sum_IE_EE_KE from expression "kinetic_energy+external_energy+internal_energy"
   compute global IE_minus_SE from expression "internal_energy-SE"
   compute global normIESE from expression "IE_minus_SE/606990" #606990  is max EE
   # shift and flip EE to show PE
   compute global PEabs_negEEplusMaxEE from expression "abs(-1*external_energy+606990)"
   compute global AbsTotalE from expression "kinetic_energy+internal_energy+PEabs_negEEplusMaxEE"
   compute global PE from expression "(-1*external_energy)+606990"
   compute global TotalE from expression "kinetic_energy+internal_energy+PE"

   compute global Kinetic_energyKJ from expression "kinetic_energy/1000"
   compute global External_energyKJ from expression "external_energy/1000"
   compute global Internal_energyKJ from expression "internal_energy/1000"
   compute global Potential_energyKJ from expression "PE/1000"
   compute global TotalE_KJ from expression "TotalE/1000"



   compute global Et_Minus_E from expression "Sum_IE_EE_KE-Sum_SE_EE_KE"
   compute global normEnergy from expression "Et_Minus_E/1.22634E6"

   compute global energyNorm from expression "max(abs(Kinetic_Energy), abs(Internal_Energy))"
   compute global max_KineticEnergy from expression "max(abs(Kinetic_Energy), abs(0))"
   compute global max_InternalEnergy from expression "max(abs(Internal_Energy), abs(0))"
```

```
  compute global max_ExternalEnergy from expression "max(abs(External_Energy), abs(0))"

  compute global max_Sum_KE_IE from expression "max_KineticEnergy+max_InternalEnergy"
  compute global max_Sum_EE_KE_PE from expression \#
                           "max_KineticEnergy+max_InternalEnergy+max_ExternalEnergy"
  compute global Dissipation_Energy from expression "max_Sum_KE_IE-max_ExternalEnergy"
  compute global DissEnergyOVERKEo from expression "max_Sum_KE_IE/602660" #pulled 602660 as max KE
End User Output

 begin Results Output output_presto
   Database Name =   newtoncradle_final_1balldrop.e
   #Database Name =   newtoncradle_final_2balldrop.e
   #Database Name =   newtoncradle_final_3balldrop.e
   Database Type = exodusII
   At time 0, Increment = .1

   nodal variables   = force_external        as f_ext
   nodal variables   = velocity              as vel
   nodal variables   = displacement          as displ
   nodal variables   = reaction              as reactions
   nodal variables   = force_internal        as f_int
   nodal variables   = force_contact         as f_cont
   nodal variables   = contact_status        as contact_stat

   element variables = stress                as stress
   element variables = von_mises             as vonmises
   element variables = effective_log_strain  as EffLogStrain
   element variables = memb_stress           as stress_memb
   element variables = truss_force           as TrussForce
   element variables = strain_energy         as StrainEnergy

   face variables    = pressure              as prssr
   face variables    = pressure_face         as prssrface

   global variables = timestep               as timestep
   global variables = external_energy        as ExternalEnergy
   global variables = internal_energy        as InternalEnergy
   global variables = kinetic_energy         as KineticEnergy
   global variables = hourglass_energy       as HourglassEnergy #should be 0
   global variables = contact_energy         as Contactenergy
   global variables = momentum               as Momentum

   global variables = energyNorm as energyNorm
   global variables = max_KineticEnergy as max_KineticEnergy
   global variables = max_InternalEnergy as max_InternalEnergy
   global variables = max_ExternalEnergy as max_ExternalEnergy
   global variables = max_Sum_EE_KE_PE as max_Sum_EE_KE_PE

   global variables = max_Sum_KE_IE as max_Sum_KE_IE
   global variables = Dissipation_energy as Dissipation_Energy
   global variables = DissEnergyOVERKEo as DissEnergyOVERKEo

   global variables = normEnergy   as normEnergy
   global variables = Et_Minus_E   as Et_Minus_E
   global variables = Sum_SE_EE_KE as Sum_SE_EE_KE
   global variables = Sum_IE_EE_KE as Sum_IE_EE_KE
   global variables = SE as strain_energy_summedElements
   global variables = PEabs_negEEplusMaxEE as PEabs_negEEplusMaxEE
   global variables = AbsTotalE as AbsTotalE
   global variables = PE as PE
   global variables = TotalE as TotalE
   global variables = normIESE as normIESE
   global variables = IE_minus_SE as IE_minus_SE

   #global variables = ee_strain_block101     as ExtEnergySumblock101
   #global variables = ee_strain_block102     as ExtEnergySumblock102
   #global variables = ee_strain_block103     as ExtEnergySumblock103
   #global variables = ee_strain_block104     as ExtEnergySumblock104
```

```
        #global variables = ee_strain_block105     as ExtEnergySumblock105

        #global variables = ke_block101            as KEsumblock101
        #global variables = ke_block102            as KEsumblock102
        #global variables = ke_block103            as KEsumblock103
        #global variables = ke_block104            as KEsumblock104
        #global variables = ke_block105            as KEsumblock105

        #global variables = ie_block101            as IEsumblock101
        #global variables = ie_block102            as IEsumblock102
        #global variables = ie_block103            as IEsumblock103
        #global variables = ie_block104            as IEsumblock104
        #global variables = ie_block105            as IEsumblock105

      end results output output_presto

    end presto region myRegion

  end presto procedure myProcedure

end sierra newtoncradlefinal
```

A.2 Bullet Collision 1.2

```
Begin Sierra Bullet

#### Title: Bullet collision with concrete block

#### Aprepro variables, which will be used in the user output section of the input file
# Ey: {Ey = 1.999479615E+11}  # Youngs Modulus
# nu: {nu = 0.33333}          # Poissons Ratio
# G: {G = 78.0E+9}            # Shear Modulus
# mu: {mu = 0.3}              # friction coefficient

##############################################################################

#### Define the point for the origin and unit vectors for each axis in both the positive
#### and negative directions.  An axis is also defined using the point 'origin' and the
#### positive x direction, which will be used to define the axis about which the bullet.
#### will rotate
Define point origin with coordinates 0.0 0.0 0.0
Define Direction posX With Vector 1.0 0.0 0.0
Define Direction posY With Vector 0.0 1.0 0.0
Define Direction posZ With Vector 0.0 0.0 1.0
Define Direction negX with Vector -1.0 0.0 0.0
Define Direction negY With Vector 0.0 -1.0 0.0
Define Direction negZ With Vector 0.0 0.0 -1.0
Define Axis Xaxis with Point origin Direction posX

##############################################################################

#### Define Functions

#### The function 'Translation' governs the translation of the bullet, and the function 'rotation'
#### governs its rotation
  Begin Function Translation
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Displacement
    Begin Values
      0.00 0.00
      0.10 0.00
      0.20 0.50
    End Values
  End Function Translation

  Begin Function Rotation
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Velocity
    Begin Values
      0.00 0.10
      0.20 0.10
    End Values
  End Function Rotation

#### The contact radius will be the radius of the cylinder and half sphere representing the bullet
#### This could be hard coded, but the following function allows the user to change his geometries
#### without having to re-change the hard coded radius each time they do so
  Begin Definition for function radius_of_bullet
    type = analytic
    expression variable: r = nodal model_coordinates(y)
    evaluate expression = "r"
  End Definition for function radius_of_bullet

  Begin Definition for function node_torque
    type = analytic
    expression variable: y = nodal coordinates(y)
    expression variable: z = nodal coordinates(z)
```

```
    expression variable: Fy = nodal force_contact(y)
    expression variable: Fz = nodal force_contact(z)
    evaluate expression = "(Fz*y) - (Fy*z)"
    #evaluate expression = "(Fy*z) - (Fz*y)"
  End Definition for function node_torque

################################################################################

#### Define Material Properties

#### Steel
  Begin Property Specification For Material Steel
    density = 10000
    Begin Parameters for Model Elastic_Plastic
      Youngs Modulus = 3.5E+11
      Poissons Ratio = 0.33333
      Yield Stress = 4.5E+8
      Hardening Modulus = 4.5E+8
    end Parameters For Model Elastic_Plastic
  end Property Specification For Material Steel

#### Mat2
  Begin Property Specification For Material Mat2
    density = 2000.00
    Begin Parameters for Model Elastic_Plastic
      Youngs Modulus = 1.7E+8
      Poissons Ratio = 0.15
      Yield Stress = 2.0E6
      Hardening Modulus = 5.0E5
    end Parameters For Model Elastic_Plastic
  end Property Specification For Material Mat2

################################################################################

#### Define Finite Element Model

  Begin Finite Element Model block_spin
    Database Name = Bullet.g
    Database Type = ExodusII


#### Define Blocks

    Begin Parameters For Block block_1
      Material Steel
      Solid Mechanics Use Model Elastic_Plastic
    End Parameters For Block block_1

    Begin Parameters For Block block_2
      Material Mat2
      Solid Mechanics Use Model Elastic_Plastic
    End Parameters For Block block_2

  End Finite Element Model block_spin

################################################################################

  Begin Presto Procedure calculations

#### Define Time and Time Step

    Begin Time Control
      Termination Time = 1.0E-3
      Begin Time Stepping Block Timestep1
        Start Time = 0.0
        Begin Parameters For Presto Region Problem
          Step Interval = 100
        End Parameters For Presto Region Problem
```

```
      End Time Stepping Block Timestep1
    End Time Control

################################################################################

    Begin Presto Region Problem
      Use Finite Element Model block_spin

      Begin Results Output block_spin_output
        Database Name = bullet.e
        database Type = ExodusII
        At Time 0.0 Increment = 5.0E-5
        Nodal Variables = Acceleration As Accel
        Nodal variables = Velocity As Vel
        Nodal Variables = Displacement As Displ
        Nodal Variables = Force_External As Force
        Nodal Variables = Force_Contact As Fc
        Nodal Variables = Reaction As React
        Nodal Variables = Torque as Torque_Node
        Element Variables = Stress As Stress
        Element Variables = Log_Strain As logstra
        Element Variables = Von_Mises As VonMises
        Element Variables = Effective_Log_Strain As ELS
        Global Variables = Ty_top
        Global Variables = T
        Global Variables = P
        Global Variables = Rc
        Global Variables = Rb
      End Results Output block_spin_output

################################################################################

#### Contact Definition

    Begin Contact Definition collide
      Search = dash
      skin all blocks = ON

      Begin Constant Friction Model friction
        Friction Coefficient = 0.3
      End Constant Friction Model friction

      Begin Interaction Defaults
        general contact = ON
        friction model = friction
      End Interaction Defaults

    End Contact Definition collide

################################################################################

      Begin Initial Velocity
        #Surface = sideset_1
        Block = block_1
        Component = X
        #magnitude = 175
        Magnitude = 400
      End Initial Velocity

      Begin Initial Velocity
        #Surface = sideset_2
        Block = block_1
        Cylindrical Axis = Xaxis
        Angular Velocity = 1000
      End Initial Velocity

#### Fixes sides of concrete block so only deformation occurs
      Begin Fixed Displacement
```

```
        Surface = sideset_3
        Components = X Y Z
      End Fixed Displacement

################################################################################

      Begin User Output
        node set = nodeset_2
        compute nodal radius_of_bullet as function radius_of_bullet
        compute global Rb as max of nodal radius_of_bullet
      End User Output

      Begin User Output
        node set = nodeset_1
        compute global P as sum of nodal Force_Contact(x)
        compute global Reac from expression "P/1000"
        compute nodal torque as function node_torque
        compute global Ty_top as sum of nodal torque
        compute global Rc from expression "3^(1/3)\#
                 *(((-1+{nu}^2)*P*(Rb)/{Ey}+0.0001)/(abs((-1+{nu}^2)*P*(Rb)/{Ey}+0.0001)))\#
                 *(abs((-1+{nu}^2)*P*(Rb)/{Ey}))^(1/3)/(2^(2/3))"
        # T is a non-dimensional value of the torque
        compute global T from expression "abs(Ty_top/({mu}*P*Rc+0.0001))"
      End User Output

      Begin history output torque
        database name = bullet.h
        database type = ExodusII
        At Time 0.0 increment = 0.01
        Variable = global Rb
        Variable = global Ty_top
        Variable = global T
        Variable = global P
        Variable = global Rc
      End history output torque

      Begin Heartbeat Output torque
        Stream Name = bullet.csv
        Format = SpyHis
        Start Time = 0.0
        At Time 0 Increment = 0.001
        Termination Time = 0.05
        Global T as Torque
        Global Rc as Contact_Radius
        Global Reac as Reaction_Force
      End Heartbeat Output torque

################################################################################

    End Presto Region Problem
  End Presto Procedure calculations
End Sierra Bullet

################################################################################
```

A.3 Analytic Planes 1.3

```
begin sierra Contact_Planes1

  define direction y_axis with vector 0.0 1.0 0.0
  define direction z_axis with vector 0.0 0.0 1.0

  begin function load
    type is piecewise linear
    begin values
      0.0 0.0
      8.0e-4 1.0
    end values
  end function load

  begin material linear_elastic_steel
    density         = 7.34e-4
    begin parameters for model elastic_plastic
      youngs modulus  = 30e6
      poissons ratio  = 0.3
      yield stress = 34.7e4
      hardening modulus = 34.7e4
      beta = 1.0
    end parameters for model elastic_plastic

  end material linear_elastic_steel

  begin finite element model mesh1
    Database Name = analytic_multiple.g
    Database Type = exodusII

    begin parameters for block block_1
      material = linear_elastic_steel
      model = elastic_plastic
    end parameters for block block_1

  end finite element model mesh1

  begin presto procedure Apst_Procedure

    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for presto region presto
        time step scale factor =  1.0
          step interval = 10
        end parameters for presto region presto
      end time stepping block p1
      termination time = 8.0e-4
    end time control

    begin presto region presto

      use finite element model mesh1

      ### output description ###
      begin Results Output output_presto
        Database Name = analytic_multiple.e
        Database Type = exodusII
        At Time 0.0, Increment = 1.0e-4
        nodal Variables = force_external as f_ext
        nodal Variables = force_internal as f_int
        nodal Variables = velocity as vel
        nodal Variables = acceleration as accl
        nodal Variables = displacement as displ
        nodal variables = contact_status
        nodal variables = force_contact as cforce
```

```
    nodal variables = mass
    element Variables = stress as stress
    global Variables = timestep as timestep
    global variables = external_energy as ExternalEnergy
    global variables = internal_energy as InternalEnergy
    global variables = kinetic_energy as KineticEnergy
    global variables = momentum as Momentum
  end results output output_presto

  ### definition of initial conditions ###

#      begin initial velocity
#        block = block_1
#        direction = y_axis
#        magnitude = -1.0e4
#      end

  begin analytic object obj1
    type = PLANE
    normal direction = 0 1 0
    origin = 0.0 0.0 0.0
    scale factor = 3.0
  end

  begin fixed displacement
    surface = obj1
    component = x y z
  end

  begin analytic object obj2
    type = PLANE
    normal direction = 1 -1 0
    origin = -4.0 0.0 0.0
    scale factor = 1.0
  end

  begin prescribed displacement
    surface = obj2
    component = y
    scale factor = -3.0
    function = load
  end
  begin fixed displacement
    surface = obj2
    component = xz
  end


#      Going forward...

#      begin analytic object obj2
#        type = CYLINDER
#        normal direction = <real> nx <real> ny <real> nz
#        origin = <real> ox <real> oy <real> oz
#        height = <real> height
#        radius = <real> radius
#      end

  ### contact definition ###

  begin contact definition basic1
    contact surface surf_1 contains obj1
    contact surface surf_2 contains obj2
    contact surface block_1 contains block_1
    search = dash
    compute contact variables = on
    begin interaction defaults
      general contact = on
```

```
          self contact = on
        end

        begin interaction
          master = surf_1
          slave = block_1
        end

        begin interaction
          master = surf_2
          slave = block_1
        end

      end contact definition basic1

    end presto region presto
  end presto procedure Apst_Procedure

end sierra Contact_Planes1
```

A.4 Curved Surface Friction Behavior 1.4

```
begin sierra BarrelRoll

define direction z with vector 0 0 1
define direction y with vector 0 1 0

begin function grav
 type is constant
 begin values
 1.0
 end values
end function grav

 begin rigid body roller
  include nodes in nodeset_1
 end rigid body roller

#Define materials
#alluminum

     BEGIN PROPERTY SPECIFICATION FOR MATERIAL MAT1
          DENSITY = 2720
          BEGIN PARAMETERS FOR MODEL ELASTIC_PLASTIC
               YOUNGS MODULUS = 68.9e6
               POISSONS RATIO =0.33
               HARDENING MODULUS = 0.0
               YIELD STRESS = 276e6
               BETA = 1.0
          END PARAMETERS FOR MODEL ELASTIC_PLASTIC
     END PROPERTY SPECIFICATION FOR MATERIAL MAT1

#define FEM model

 begin finite element model ramp
  database name = 40.g
  database type = exodusII
  begin parameters for block block_1 block_2 block_3
   material = MAT1
   SOLID MECHANICS USE MODEL ELASTIC_PLASTIC
  end parameters for block block_1 block_2 block_3
 end finite element model ramp


 begin presto procedure precal
#define problem time parameters
  begin time control
   termination time = 2
   begin time stepping block timestepping
    start time = 0.0
   end time stepping block timestepping
  end time control

#define problem
  begin presto region local
   use finite element model ramp

#define output
   begin results output putout
    database name = %B.e
    database type = exodusII
    at time 0.0, increment = .1
    nodal variables = displacement
    nodal variables = velocity
  #  global variables = angular_momentum
  #  global variables = angular_momentum_block1
    global variables = rotvz_roller as rotational_velocity
```

```
    global variables = velx_roller as vx
    global variables = vely_roller as vy
    global variables = displx_roller
    global variables = disply_roller
 #   nodal variables = contact_incremental_slip_magnitude As slip_magnitude
    nodal variables = contact_status
    global variables = Slip_Ratio
   end results output putout

   begin user output
  #  block = block_1
    node set = nodeset_1
  #   compute global angular_momentum_block1 as max of angular_momentum
    Compute Global tv from expression "abs(rotVz_roller*0.2)"
    compute global displace from expression "sqrt(displx_roller^2+disply_roller^2)"
    Compute Global V from expression "sqrt(velx_roller^2+vely_roller^2)"
    Compute Global Slip_Ratio from expression "V/tv-1"
   end user output

  # begin history output
  #  database name = %B.h
  #  database type = exodusII
  #  overwrite = on
  #  at time 0.0, increment = 0.1
  #  variable = global timestep as t
  #  variable = global Slip_Ratio as Slip_Ratio
  # # Variable = nodal contact_status as contact_status
  #  variable = global rotvz_roller as rotational_velocity
  #  variable = global V as V
  #  variable = global displace as displace
  #  termination time = 2
  # end history output

    Begin Heartbeat Output
      Stream Name = %B.csv
      Format = SpyHis
      Start Time = 0.0
      At Time 0 Increment = 0.04
      Termination Time = 2.0
      global timestep as t
      global Slip_Ratio as Slip_Ratio
      global rotvz_roller as rotational_velocity
      global V as V
      global displace as displace
    End Heartbeat Output

#define boundary conditions
   begin fixed displacement
    block = block_2
    components = x z y
   end fixed displacement
   begin gravity
    function = grav
    direction = y
    gravitational constant = -9.81
   end gravity

#define contact
   begin contact definition contact
    compute contact variables = on
    skin all blocks = on
    search = dash
    begin interaction defaults
     friction model = sticky
     general contact = on
     self contact = off
     #constraint formulation = node_face
    end interaction defaults
```

```
     begin constant friction model sticky
      friction coefficient = 0.40
    end constant friction model sticky
   end contact definition contact
  end presto region local
 end presto procedure precal
end sierra BarrelRoll
```

A.5 Plate Indentation 1.5

```
Begin Sierra Axis_Symmetric_Graded

  #### Title Indentation of a thick plate

  ################################################################################

  #### Initialize Directions
  Define Direction posX With Vector 1.0 0.0 0.0
  Define Direction posY With Vector 0.0 1.0 0.0
  Define Direction posZ With Vector 0.0 0.0 1.0
  Define Direction negX With Vector -1.0 0.0 0.0
  Define Direction negY With Vector 0.0 -1.0 0.0
  Define Direction negZ With Vector 0.0 0.0 -1.0

################################################################################

#### Define Functions

  Begin Function Compression
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Displacement
    Begin Values
      0.00  0.0
      0.01  1.0
    End Values
  End Function Compression

#  Begin Function Compression
#    Type = Analytic
#    Expression Variable: t = Global Time
#    #Evaluate Expression = "250*t"
#    Evaluate Expression = "t"
#  End Function Compression

################################################################################

#### Define Materials

#### Crushable Foam
  Begin Property Specification For Material CF
    Density = 60.0
    Begin Parameters For Model Elastic
      Youngs Modulus = 10E+5
      Poissons Ratio = 0.1
    End Parameters For Model Elastic
  End Property Specification For Material CF

  Begin Property Specification For Material ALUMINIUM
    Density = 7.4E+4
    Begin Parameters For Model Elastic
      Youngs Modulus = 30.0E6
      Poissons Ratio = 0.333
    End Parameters For Model Elastic
  End Property Specification For Material ALUMINIUM

################################################################################

#### Define FEM Model

  Begin Finite Element Model fullgraded
    Database Name = PlateIndentation.g
    Database Type = ExodusII

#### Define Blocks
```

```
    Begin Parameters For Block block_1
      Material CF
      Model = Elastic
    End Parameters For Block block_1

    Begin Parameters For Block block_2
      Material ALUMINIUM
      Model = Elastic
    End Parameters For Block block_2

  End Finite Element Model fullgraded

################################################################################

  Begin presto Procedure calculations

    #### Define Time and Time Step

    Begin Time Control
      Termination Time = 1.0E-2
      Begin Time Stepping Block timestep1
        Start Time = 0.0
        Begin Parameters For presto Region Problem
          Step Interval = 100
        End Parameters For presto Region Problem
      End Time Stepping Block timestep1
    End Time Control

################################################################################

    Begin presto Region Problem
      Use Finite Element Model fullgraded

      Begin Results Output axisymmetric_output
        Database Name = PlateIndentation.e
        Database Type = ExodusII
        At Time 0.0 Increment = 5.0E-4
        Nodal Variables = Acceleration As Accel
        Nodal variables = Velocity As Vel
        Nodal Variables = Displacement As Displ
        Nodal Variables = Reaction As Force
        Element Variables = Stress As Stress
        Element Variables = Log_Strain As logstra
        Element Variables = Von_Mises As VonMises
        Element Variables = Effective_Log_Strain As ELS
#        nodal variables = force_contact as fc
#        nodal variables = contact_status
#        nodal variables = contact_tangential_direction as cdirtan
#        nodal variables = contact_normal_direction as cdirnor
#        nodal variables = contact_normal_traction_magnitude as cfnor
#        nodal variables = contact_tangential_traction_magnitude as cftan
#        nodal variables = contact_area
      End Results Output axisymmetric_output

################################################################################

      Begin Fixed Displacement
        Surface = sideset_1
        Components = X Y Z
      End Fixed Displacement

      Begin Fixed Displacement
        Surface = sideset_6
        Component = X
      End Fixed Displacement

      Begin Fixed Displacement
        Surface = sideset_3
```

```
        Component = X
      End Fixed Displacement

      Begin Fixed Displacement
        Surface = sideset_7
        Component = Z
      End Fixed Displacement

      Begin Fixed Displacement
        Surface = sideset_4
        Component = Z
      End Fixed Displacement

      Begin Prescribed Displacement
        #Surface = sideset_8
        Block = block_2
        Direction = posY
        Function = Compression
        scale factor = -0.2
      End Prescribed Displacement

################################################################################

#### Contact

      Begin Contact Definition fullgraded
        Skin all blocks = ON
        Search = dash
        Enforcement = al
        begin interaction defaults
          friction model = CTF
          general contact = on
          self contact  = off
          constraint formulation = node_face
          al penalty = 1.25
        end
        begin dash options
          subdivision level = 5
          interaction definition scheme = explicit
        end
        Begin Constant Friction Model CTF
          Friction Coefficient = 0.1
        End Constant Friction Model CTF
      End Contact Definition fullgraded

      Begin Solver
        begin loadstep predictor
          type = scale_factor
          scale factor = 0.0
        end
        begin control contact
          target relative residual = 1.0E-3
          Maximum Iterations       = 200
          minimum iterations       = 5
          iteration plot           = 1
        end
        Begin cg
          reference = belytschko
          target relative residual = 1.0E-4
          maximum iterations       = 25
          begin full tangent preconditioner
          end
        End

      End Solver

    End presto Region Problem
  End presto Procedure calculations
```

```
End Sierra Axis_Symmetric_Graded
```

A.6 Angled Crack Cylinder 2.1

```
begin sierra btshell_cylinder

  begin definition for function function_radius_over_time
    type is piecewise linear
    begin values
      0.000  2.5
      0.005  2.5
      0.010  2.5
    end values
  end definition for function function_radius_over_time

  begin definition for function function_2
    type is piecewise linear
    begin values
      0.000  0.0
      0.0025 0.3
      0.005  0.3
    end values
  end definition for function function_2

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin property specification for material linear_elastic
    density         = 2.61e-4
    begin parameters for model elastic_plastic
      youngs modulus  = 1.e9
      poissons ratio  = 0.25
      HARDENING MODULUS = 0.0
      YIELD STRESS = 36000.0
      BETA = 1.0
    end parameters for model elastic_plastic
  end property specification for material linear_elastic

  begin shell section bt_section
    thickness = 2
    formulation = bt_shell
  end

  begin finite element model mesh1
    Database Name = cylinder_shell.g
    Database Type = exodusII

    begin parameters for block block_1
      section = bt_section
      material linear_elastic
      model = elastic_plastic
      hourglass stiffness = 0.05
    end parameters for block block_1

  end finite element model mesh1

  begin presto procedure Presto_Procedure

    begin time control
      begin time stepping block p1
        start time =  0.0

        begin parameters for presto region presto
          time step scale factor = 1.0
        end parameters for presto region presto

      end time stepping block p1
```

```
      termination time = 0.0003
    end time control

    begin presto region presto
      use finite element model mesh1

      begin XFEM xfem
        include all blocks
        add disc = 0.0 -1.0 6.0 0.0 1.0 1.2 function_radius_over_time
        mechanics growth start time = 0.0
        mechanics growth method = mechanics failure
        failure surface evolution = planar
        criterion is element value of max_principal_stress(1) > 2.5e4
        angle change = stress eigenvector
      end

      begin prescribed displacement
        node set = nodelist_2001
        component = y
        function = function_2
        scale factor = 40.0
      end prescribed displacement

      begin prescribed displacement
        node set = nodelist_2002
        component = y
        function = function_2
        scale factor = -40.0
      end prescribed displacement

      ### output description ###
      begin Results Output output_presto
        Database Name = cylinder_shell.e
        Database Type = exodusII
        At time 0.0 increment = 0.000001
        nodal Variables = displacement
        element Variables = memb_stress as mstress
        element Variables = top_stress as topstress
        element Variables = xfem_partial_element_flag
        element Variables = xfem_element_fail_flag
        element Variables = xfem_physical_node_flag
        element Variables = xfem_edge_cut_flag
        element variables = max_principal_stress as my_max_prin
        element Variables = VON_MISES as VonMises
        global Variables = timestep as timestep
        global variables = external_energy
        global variables = internal_energy
        global variables = kinetic_energy
        global variables = momentum
        global variables = facesum
      end results output output_presto

      begin user output
        block = block_1_contact_surface
        compute global faceSum as sum of element skinFaceArea
      end

    end presto region presto
  end presto procedure Presto_Procedure

end sierra btshell_cylinder
```

A.7 Plate with Multiple Holes 2.2

```
begin sierra btshell_multiHoles

  begin definition for function function_2
    type is piecewise linear
    begin values
      0.000  0.0
      0.05    0.1
      0.1    0.2
      0.15    0.3
      0.2    0.4
      0.25    0.5
      0.3     0.6
      0.35    0.7
      0.4    0.8
      0.45    0.9
      0.5    1.0
    end values
  end definition for function function_2

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin material plate_mat
    density         = 0.0078
    begin parameters for model elastic_plastic
      youngs modulus  = 210E3
      poissons ratio  = 0.3
      yield stress = 360
      hardening modulus = 50E3
      beta = 0.75
    end parameters for model elastic_plastic
  end material plate_mat

  begin shell section bt_section
    thickness = 0.8
    formulation = bt_shell
    integration rule = gauss
    number of integration points = 4
  end

  begin cohesive section cohesive_section
    thickness = 1.0
    number of integration points = 4
  end

  begin finite element model mesh1
    Database Name = multiHoles.g
    Database Type = exodusII

    begin parameters for block block_1
      section = bt_section
      material plate_mat
      model = elastic_plastic
      hourglass stiffness = 0.8
      hourglass viscosity = 0.2
    end parameters for block block_1

  end finite element model mesh1

  begin presto procedure Presto_Procedure

    begin time control
      begin time stepping block p1
        start time =  0.0
```

```
      begin parameters for presto region presto
        time step scale factor = 0.1
      end parameters for presto region presto

    end time stepping block p1

    termination time = 0.036
  end time control

  begin presto region presto
    use finite element model mesh1

    begin XFEM xfem1
      include all blocks
      generation by nucleation = element-based
      nucleation criterion is element value of max_principal_stress > 1.0E3
      angle change = stress eigenvector
      mechanics growth start time = 0.0
      mechanics growth method = mechanics failure
      failure surface evolution = piecewise linear
      criterion is element value of max_principal_stress > 8.0E2
      crack branching = allowed
      branching criterion is element value of max_principal_stress > 9.5E2
    end

    begin prescribed force
      node set = nodelist_1
      component = y
      function = function_2
      scale factor = -4000
    end prescribed force

    begin prescribed force
      node set = nodelist_2
      component = y
      function = function_2
      scale factor = 4000
    end prescribed force

    ### output description ###
    begin Results Output output_presto
      Database Name = multiHoles.e
      Database Type = exodusII
      At time 0.0 increment = 0.0001
      nodal Variables = displacement
      nodal Variables = rotational_displacement as rot_disp
      element Variables = memb_stress as ms
      element Variables = VON_MISES as VonMises
      element Variables = xfem_partial_element_flag
      element Variables = xfem_element_fail_flag
      element Variables = xfem_physical_node_flag
      element Variables = xfem_edge_cut_flag
      element variables = max_principal_stress as my_max_prin
      global Variables = timestep as timestep
      global variables = external_energy
      global variables = internal_energy
      global variables = kinetic_energy
      global variables = momentum
    end results output output_presto

  end presto region presto
end presto procedure Presto_Procedure

end sierra btshell_multiHoles
```

A.8 Stress Strain Plate 3.1

```
##
## Mesh | Loading | Command Line
##--------------------------------
## mesh1 | zero z-displacement | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h1 thickness=t05  plane_constraint=zero_displacement"
## mesh1 | zero pressure       | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h1 thickness=t05  plane_constraint=zero_pressure"
## mesh1 | zero z-traction     | sierra adagio -i plateWithHole.i ....
## ... -D "elem=hex section=ug h_refinement=h1 thickness=t05  plane_constraint=zero_traction"
## mesh1 | zero z-force        | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h1 thickness=t05  plane_constraint=zero_force"
## mesh1 | free z-DOF          | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h1 thickness=t05  plane_constraint=free"
## mesh2 | zero z-displacement | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h2 thickness=t025 plane_constraint=zero_displacement"
## mesh2 | zero pressure       | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h2 thickness=t025 plane_constraint=zero_pressure"
## mesh2 | zero z-traction     | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h2 thickness=t025 plane_constraint=zero_traction"
## mesh2 | zero z-force        | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h2 thickness=t025 plane_constraint=zero_force"
## mesh2 | free z-DOF          | sierra adagio -i plateWithHole.i ...
## ... -D "elem=hex section=ug h_refinement=h2 thickness=t025 plane_constraint=free"
##

begin sierra plateWithHole

  title plate with a hole test

  define direction x with vector 1.0 0.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin function line
    type is analytic
    evaluate expression = "x"
  end function line

  begin function zero
    type is analytic
    evaluate expression = "0.0"
  end function zero

  begin material steel
    density = 1.0
    begin parameters for model elastic
      youngs modulus = 200.0e9
      poissons ratio = 0.3
    end parameters for model elastic
  end material steel

  {if(section == "ug")}
  begin solid section {section}
  end solid section {section}
  {endif}

  {if(section == "sd")}
  begin solid section {section}
    formulation = selective_deviatoric
    deviatoric parameter = 1.0
  end solid section {section}
  {endif}

  begin finite element model fem
    database name = {elem}.{h_refinement}.{thickness}.g
    begin parameters for block block_1
```

```
    material = steel
    model = elastic
    section = {section}
  end parameters for block block_1
end finite element model fem

begin adagio procedure agio_procedure

  begin time control
    begin time stepping block p0
      start time = 0.0
      begin parameters for adagio region agio_region
        time increment = 1.0
      end parameters for adagio region agio_region
    end time stepping block p0
    termination time = 1.0
  end time control

  begin adagio region agio_region

    use finite element model fem

    begin fixed displacement left_symmetry_BC
      node set = nodelist_1
      components = x
    end fixed displacement left_symmetry_BC

    begin fixed displacement bottom_symmetry_BC
      node set = nodelist_2
      components = y
    end fixed displacement bottom_symmetry_BC

    begin fixed displacement back_symmetry_BC
      surface = sideset_2
      components = z
    end fixed displacement back_symmetry_BC

    begin traction right_tensile_load
      surface = sideset_3
      direction = x
      function = line
      scale factor = 10000.0
    end traction right_tensile_load

    {if(plane_constraint == "zero_force")}
    begin prescribed force plane_constraint_{plane_constraint}
      surface = sideset_1
      component = z
      function = zero
    end prescribed force plane_constraint_{plane_constraint}
    {endif}

    {if(plane_constraint == "zero_traction")}
    begin traction plane_constraint_{plane_constraint}
      surface = sideset_1
      direction = z
      function = zero
    end traction plane_constraint_{plane_constraint}
    {endif}

    {if(plane_constraint == "zero_pressure")}
    begin pressure plane_constraint_{plane_constraint}
      surface = sideset_1
      function = zero
    end pressure plane_constraint_{plane_constraint}
    {endif}

    {if(plane_constraint == "zero_displacement")}
```

```
      begin prescribed displacement plane_constraint_{plane_constraint}
        surface = sideset_1
        component = z
        function = zero
      end prescribed displacement plane_constraint_{plane_constraint}
      {endif}

      begin results output agio_region_output
        database name = {elem}.{section}.{h_refinement}.{thickness}.{plane_constraint}.e
        at time 1.0 increment = 1.0
        global variables = kinetic_energy
        global variables = internal_energy
        global variables = external_energy
        global variables = stress_zz_max
        global variables = log_strain_zz_max
        nodal variables = displacement
        nodal variables = force_external
        nodal variables = force_internal
        element variables = log_strain
        element variables = stress
        element variables = unrotated_stress
      end results output agio_region_output

      begin user output
        include all blocks
        compute global stress_zz_max     as max absolute value of element stress(zz)
        compute global log_strain_zz_max as max absolute value of element log_strain(zz)
      end user output

      begin solution verification
        {if(plane_constraint == "zero_displacement")}
        verify global log_strain_zz_max <= 1.0e-12
        {else}
        verify global stress_zz_max <= 100.0
        {endif}
        completion file = {elem}.{section}.{h_refinement}.{thickness}.{plane_constraint}.verif
      end solution verification

      begin solver
        begin cg
          reference = external
          target relative residual = 1e-10
          begin full tangent preconditioner
            iteration update = 10
          end full tangent preconditioner
        end cg
      end solver

    end adagio region agio_region
  end adagio procedure agio_procedure
end sierra plateWithHole
```

A.9 Bolt Preload 3.2

A.9.1 Thermal Strain

```
begin Sierra

  # Metric units are used.
  # - displacement: meters
  # - mass: kilograms
  # - time: seconds
  # - force: kg*m/s^2
  # - temperature: Kelvin

  #The bolt is cooled to -10 Kelvin in one step.
  begin definition for function TEMPERATURE
    type is piecewise linear
    ordinate is temperature
    abscissa is time
    begin values
      0.0       0.0
      1.0     -10.0
      10.0    -10.0
    end values
  end definition for function TEMPERATURE

  begin definition for function THERMAL_STRAIN_X
    type is piecewise linear
    ordinate is strain
    abscissa is temperature
    begin values
      0.0        0.0
      1.0        0.0
      1000       0.0
    end values
  end definition for function THERMAL_STRAIN_X

  #Thermal strain is applied to the bolt only in the longitudinal direction of
  #the bolt.
  begin definition for function THERMAL_STRAIN_Y
    type is piecewise linear
    ordinate is strain
    abscissa is temperature
    begin values
      0.0      0.0
      -10       -5.0E-02
      -1000.0   -5.0E-02
    end values
  end definition for function THERMAL_STRAIN_Y

  begin definition for function THERMAL_STRAIN_Z
    type is piecewise linear
    ordinate is strain
    abscissa is temperature
    begin values
      0.0        0.0
      1.0        0.0
      1000       0.0
    end values
  end definition for function THERMAL_STRAIN_Z

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  ### -------------------------- ###
  ###   Materials Specification   ###
  ### -------------------------- ###
```

```
  begin property specification for material steel_bolt
    density          = 7.89e+03
    thermal engineering strain X function = THERMAL_STRAIN_X
    thermal engineering strain Y function = THERMAL_STRAIN_Y
    thermal engineering strain Z function = THERMAL_STRAIN_Z
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
    end parameters for model elastic
  end property specification for material steel_bolt

  begin property specification for material steel_block
    density          = 7.89e+03
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
    end parameters for model elastic
  end property specification for material steel_block

  ### -------------------------------------- ###
  ###   Input Mesh and Material Description   ###
  ### -------------------------------------- ###

  begin finite element model thermalPreLoad
    database name = thermal_final.g
    database type = exodusII

    begin parameters for block block_1
      material steel_block
      model = elastic
    end parameters for block block_1

    begin parameters for block block_2
      material steel_bolt
      model = elastic
    end parameters for block block_2

  end finite element model thermalPreLoad

  ### ---------------------- ###
  ###   Linear Solver        ###
  ### ---------------------- ###

  begin feti equation solver feti
    # Turn on to see feti iterations in the log file.
    #param-string "debugMask" value "solver"
    param-real "damping_coefficient" value 0.0001
    residual norm tolerance = 1e-2
  end

### ---------------- ###
###   Begin Adagio   ###
### ---------------- ###

  begin adagio procedure aProcedure

    ### ---------------- ###
    ###   Time Control   ###
    ### ---------------- ###

    begin time control

      begin time stepping block time1
        start time = 0.0
        begin parameters for adagio region aRegion
          number of time steps = 1
        end parameters for adagio region aRegion
```

```
  end time stepping block time1

  termination time = 1.0

end time control

### ----------------- ###
###   Adagio Region   ###
### ----------------- ###

begin adagio region aRegion
  use finite element model thermalPreLoad


  ### ----------------------- ###
  ###   Boundary Conditions   ###
  ### ----------------------- ###


  # --------------------- #
  #     Bolt Preload      #
  # --------------------- #

  #The temperature changed is applied to the entire bolt.
  begin prescribed temperature
    block = block_2
    function = TEMPERATURE
  end prescribed temperature

  # --------------------- #
  #        Loading        #
  # --------------------- #

  begin fixed displacement
    node set = nodelist_3
    component = Z
  end

  begin fixed displacement
    node set = nodelist_4
    component = X Y
  end

  begin fixed displacement
    node set = nodelist_5
    component = X Y
  end


  # -------------------- #
  #   Contact Surfaces   #
  # -------------------- #

  begin contact definition
    search = acme

    contact surface surf_101 contains surface_101
    contact surface surf_102 contains surface_102
    contact surface surf_201 contains surface_201
    contact surface surf_202 contains surface_202

    begin constant friction model fric
      friction coefficient = 0.5
    end

    begin interaction
      master               = surf_102
      slave                = surf_101
```

```
        friction model       = fric
      end interaction

      begin interaction
        master               = surf_202
        slave                = surf_201
        friction model       = fric
      end interaction

    end contact definition

    ### -------------------- ###
    ###   Output Variables   ###
    ### -------------------- ###

    begin results output aOut
      database name = thermal_final.e
      database type = exodusII
      at step 0 increment = 1
      nodal variables = displacement as displ
      nodal variables = contact_status as celement
      nodal variables = contact_normal_traction_magnitude as cfnor
      nodal variables = contact_tangential_traction_magnitude as cftan
      nodal variables = contact_slip_increment_current as cdtan
      nodal variables = contact_area as carea
      nodal variables = contact_normal_direction as cdirnor
      nodal variables = contact_tangential_direction as cdirtan
      element variables = stress as stress
      element Variables = thermal_strain_3d
      element Variables = temperature as temp
      global variables = total_iter as itotal
      global variables  = timestep as timestep
    end

    begin solver

      begin loadstep predictor
        type = scale_factor
        scale factor = 1.0 0.0
      end

      level 1 predictor = none

      begin control contact
        target relative residual        = 1.0e-3
        maximum iterations              = 100
      end

      begin cg
        target     relative residual    = 1.0e-4
        maximum iterations              = 100

        begin full tangent preconditioner
          linear solver                   = feti
          conditioning                    = no_check
          small number of iterations      = 15
        end
      end
    end

  end adagio region aRegion
 end adagio procedure aProcedure

end Sierra
```

A.9.2 Artificial Strain

```
begin Sierra

  # Metric units are used.
  # - displacement: meters
  # - mass: kilograms
  # - time: seconds
  # - force: kg*m/s^2
  # - temperature: Kelvin


  begin definition for function ARTIFICIAL_STRAIN_X
    type is piecewise linear
    ordinate is strain
    abscissa is time
    begin values
      0.0        0.0
      1.0       0.0005
    end values
  end definition for function ARTIFICIAL_STRAIN_X

  #ARTIFICIAL strain is applied to the bolt only in the longitudinal direction of
  #the bolt.
  begin definition for function ARTIFICIAL_STRAIN_Y
    type is piecewise linear
    ordinate is strain
    abscissa is time
    begin values
      0.0      0.0
      1       -0.05
    end values
  end definition for function ARTIFICIAL_STRAIN_Y

  begin definition for function ARTIFICIAL_STRAIN_Z
    type is piecewise linear
    ordinate is strain
    abscissa is time
    begin values
      0.0        0.0
      1.0        0.0
    end values
  end definition for function ARTIFICIAL_STRAIN_Z

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  ### --------------------------- ###
  ###   Materials Specification   ###
  ### --------------------------- ###

  begin property specification for material steel_bolt
    density          = 7.89e+03
    ARTIFICIAL engineering strain X function = ARTIFICIAL_STRAIN_X
    ARTIFICIAL engineering strain Y function = ARTIFICIAL_STRAIN_Y
    ARTIFICIAL engineering strain Z function = ARTIFICIAL_STRAIN_Z
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
    end parameters for model elastic
  end property specification for material steel_bolt

  begin property specification for material steel_block
    density          = 7.89e+03
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
```

```
    end parameters for model elastic
  end property specification for material steel_block

  ### --------------------------------------- ###
  ###   Input Mesh and Material Description   ###
  ### --------------------------------------- ###

  begin finite element model ARTIFICIALPreLoad
    database name = artificialStrain_final.g
    database type = exodusII

    begin parameters for block block_1
      material steel_block
      model = elastic
    end parameters for block block_1

    begin parameters for block block_2
      material steel_bolt
      model = elastic
    end parameters for block block_2

  end finite element model ARTIFICIALPreLoad

  ### ---------------------- ###
  ###   Linear Solver        ###
  ### ---------------------- ###

  begin feti equation solver feti
    # Turn on to see feti iterations in the log file.
    #param-string "debugMask" value "solver"
    param-real "damping_coefficient" value 0.0001
    residual norm tolerance = 1e-2
  end

### ---------------- ###
###   Begin Adagio   ###
### ---------------- ###

  begin adagio procedure aProcedure

    ### ---------------- ###
    ###   Time Control   ###
    ### ---------------- ###

    begin time control

      begin time stepping block time1
        start time = 0.0
        begin parameters for adagio region aRegion
          number of time steps = 1
        end parameters for adagio region aRegion
      end time stepping block time1

      termination time = 1.0

    end time control

    ### ----------------- ###
    ###   Adagio Region   ###
    ### ----------------- ###

    begin adagio region aRegion
      use finite element model ARTIFICIALPreLoad


      ### ----------------------- ###
      ###   Boundary Conditions   ###
      ### ----------------------- ###
```

```
# --------------------- #
#        Loading        #
# --------------------- #

begin fixed displacement
  node set = nodelist_3
  component = Z
end

begin fixed displacement
  node set = nodelist_4
  component = X Y
end

begin fixed displacement
  node set = nodelist_5
  component = X Y
end


# -------------------- #
#   Contact Surfaces   #
# -------------------- #

begin contact definition
  search = acme

  contact surface surf_101 contains surface_101
  contact surface surf_102 contains surface_102
  contact surface surf_201 contains surface_201
  contact surface surf_202 contains surface_202

  begin constant friction model fric
    friction coefficient = 0.5
  end

  begin interaction
    master              = surf_102
    slave               = surf_101
    friction model      = fric
  end interaction

  begin interaction
    master              = surf_202
    slave               = surf_201
    friction model      = fric
  end interaction

end contact definition

### -------------------- ###
###   Output Variables   ###
### -------------------- ###

begin results output aOut
  database name = artificialStrain_final.e
  database type = exodusII
  at step 0 increment = 1
  nodal variables = displacement as displ
  nodal variables = contact_status as celement
  nodal variables = contact_normal_traction_magnitude as cfnor
  nodal variables = contact_tangential_traction_magnitude as cftan
  nodal variables = contact_slip_increment_current as cdtan
  nodal variables = contact_area as carea
  nodal variables = contact_normal_direction as cdirnor
  nodal variables = contact_tangential_direction as cdirtan
  element variables = stress as stress
```

```
        element Variables = ARTIFICIAL_strain_3d
        global variables = total_iter as itotal
        global variables  = timestep as timestep
      end

      begin solver

        begin loadstep predictor
          type = scale_factor
          scale factor = 0.0 #1.0
        end

        level 1 predictor = none

        begin control contact
          target relative residual        = 1.0e-3
          maximum iterations              = 100
        end

        begin cg
          target     relative residual    = 1.0e-4
          maximum iterations              = 100

          begin full tangent preconditioner
            linear solver                   = feti
            conditioning                    = no_check
            small number of iterations      = 15
          end
        end
      end

    end adagio region aRegion
  end adagio procedure aProcedure

end Sierra
```

A.9.3 Prescribed Displacement

```
begin Sierra

  # Metric units are used.
  # - displacement: meters
  # - mass: kilograms
  # - time: seconds
  # - force: kg*m/s^2
  # - temperature: Kelvin


  begin definition for function Disp
    type is piecewise linear
    ordinate is disp
    abscissa is time
    begin values
      0.0       0.0
      0.1       -0.00005
      0.2       -0.0001
      0.3       -0.00015
      0.4       -0.0002
      0.5       -0.00025
      0.6       -0.0003
      0.7       -0.00035
      0.8       -0.0004
      0.9       -0.00045
      1.0       -0.0005         # Absolute length of initial overlap is 0.0005
      1.1       -0.0005
      1.2       -0.0005
      1.3       -0.0005         # Start of second time step is 1.3 (where contact is turned on)
    end values
  end definition for function Disp


  begin definition for function Disp2
    type is piecewise linear
    ordinate is disp
    abscissa is time
    begin values
      0.0       0.0
      0.1       0.00005
      0.2       0.0001
      0.3       0.00015
      0.4       0.0002
      0.5       0.00025
      0.6       0.0003
      0.7       0.00035
      0.8       0.0004
      0.9       0.00045
      1.0       0.0005        # Absolute length of initial overlap is 0.0005
      1.1       0.0005
      1.2       0.0005
      1.3       0.0005        # Start of second time step is 1.3 (where contact is turned on)
    end values
  end definition for function Disp2


  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  ### -------------------------- ###
  ###   Materials Specification   ###
  ### -------------------------- ###

  begin property specification for material steel_bolt
    density          = 7.89e+03
```

```
  begin parameters for model elastic
    youngs modulus = 200.e+09
    poissons ratio = 0.29
  end parameters for model elastic
end property specification for material steel_bolt

begin property specification for material steel_block
  density          = 7.89e+03
  begin parameters for model elastic
    youngs modulus = 200.e+09
    poissons ratio = 0.29
  end parameters for model elastic
end property specification for material steel_block

### --------------------------------------- ###
###   Input Mesh and Material Description   ###
### --------------------------------------- ###

begin finite element model preDisp
  database name = preDisp_final.g
  database type = exodusII

  begin parameters for block block_1
    material steel_block
    model = elastic
  end parameters for block block_1

  begin parameters for block block_2
    material steel_bolt
    model = elastic
  end parameters for block block_2

end finite element model preDisp

### ---------------------- ###
###   Linear Solver        ###
### ---------------------- ###

begin feti equation solver feti
  residual norm tolerance = 5e-2
end

### ---------------- ###
###   Begin Adagio   ###
### ---------------- ###

begin adagio procedure aProcedure

  ### ---------------- ###
  ###   Time Control   ###
  ### ---------------- ###

  begin time control

    begin time stepping block time1
      start time = 0.0
      begin parameters for adagio region aRegion
        number of time steps = 13
      end parameters for adagio region aRegion
    end time stepping block time1

    begin time stepping block time2
      start time = 1.3
      begin parameters for adagio region aRegion
        number of time steps = 7
      end parameters for adagio region aRegion
    end time stepping block time2
```

```
  termination time = 2.0

end time control

### ----------------- ###
###   Adagio Region   ###
### ----------------- ###

begin adagio region aRegion
  use finite element model preDisp


  ### ----------------------- ###
  ###   Boundary Conditions   ###
  ### ----------------------- ###


  # --------------------- #
  #     Bolt Preload      #
  # --------------------- #

  begin prescribed displacement
    node set = nodelist_2
    function = Disp
    active periods = time1
    component = Y
  end

  begin prescribed displacement
    node set = nodelist_1
    function = Disp2
    active periods = time1
    component = Y
  end

  # --------------------- #
  #        Loading        #
  # --------------------- #

  begin fixed displacement
    node set = nodelist_3
    component = Z
  end

  begin fixed displacement
    node set = nodelist_4
    component = X Y
  end

  begin fixed displacement
    node set = nodelist_5
    component = X Y
  end


  # -------------------- #
  #   Contact Surfaces   #
  # -------------------- #

  begin contact definition
    search = dash

    contact surface surf_101 contains surface_101
    contact surface surf_102 contains surface_102
    contact surface surf_201 contains surface_201
    contact surface surf_202 contains surface_202

    begin constant friction model fric
```

```
    friction coefficient = 0.5
  end

  begin inactive friction model inactive
  end

  begin time variant model tv
    model = inactive during periods time1
    model = fric during periods time2
  end

  begin interaction
    master               = surf_102
    slave                = surf_101
    normal tolerance     = 1.0e-5
    tangential tolerance = 1.0e-5
    capture tolerance    = 1.0e-6
    friction model       = fric
  end interaction

  begin interaction
    master               = surf_202
    slave                = surf_201
    normal tolerance     = 1.0e-5
    tangential tolerance = 1.0e-5
    capture tolerance    = 1.0e-6
    friction model       = tv
  end interaction

end contact definition

### -------------------- ###
###   Output Variables   ###
### -------------------- ###

begin results output aOut
  database name = preDisp_final.e
  database type = exodusII
  at step 0 increment = 1
  nodal variables = displacement as displ
  nodal variables = contact_status as celement
  nodal variables = contact_normal_traction_magnitude as cfnor
  nodal variables = contact_tangential_traction_magnitude as cftan
  nodal variables = contact_slip_increment_current as cdtan
  nodal variables = contact_area as carea
  nodal variables = contact_normal_direction as cdirnor
  nodal variables = contact_tangential_direction as cdirtan
  element variables = stress as stress
  element Variables = thermal_strain_3d
  element Variables = temperature as temp
  global variables = total_iter as itotal
  global variables  = timestep as timestep
end

begin solver

  begin loadstep predictor
    type = scale_factor
    scale factor = 0.0 0.0
  end

  level 1 predictor = none

  begin control contact
    target relative residual        = 1.0e-3
    maximum iterations              = 80
  end
```

```
        begin cg
          target     relative residual    = 1.0e-4
          maximum iterations              = 100
          acceptable residual             = 1.0e+10

          begin full tangent preconditioner
            tangent diagonal scale          = 1.0e-6
            linear solver                   = feti
            conditioning                    = no_check
            small number of iterations      = 10
          end
        end
      end

    end adagio region aRegion
  end adagio procedure aProcedure

end Sierra
```

A.9.4 Spring

```
begin Sierra

  # Metric units are used.
  # - displacement: meters
  # - mass: kilograms
  # - time: seconds
  # - force: kg*m/s^2
  # - temperature: Kelvin

  begin function force_strain ## iterate on this version ## pretty much there with preload 5.0667e5
    type is piecewise linear
    ordinate is force
    abscissa is engineering_strain
    begin values
      -0.05        -7.25e5
      -0.025       -4.25e5
      0.0          0.0
      0.025        4.25e5
      0.05         7.25e5
    end values
  end

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin rigid body TOP_FLANGE
    include nodes in nodelist_101
  end rigid body TOP_FLANGE

  begin rigid body BOTTOM_FLANGE
    include nodes in nodelist_102
  end rigid body BOTTOM_FLANGE

  begin spring section spring_1
    force strain function = force_strain
    default stiffness =5.0265e8
    preload = 5.0667e5 #DEFAULT
    mass per unit length = 0.0
  end


  ### --------------------------- ###
  ###   Materials Specification   ###
  ### --------------------------- ###

  begin property specification for material steel_bolt
    density          = 7.89e+03
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
    end parameters for model elastic
  end property specification for material steel_bolt

  begin property specification for material steel_block
    density          = 7.89e+03
    begin parameters for model elastic
      youngs modulus = 200.e+09
      poissons ratio = 0.29
    end parameters for model elastic
  end property specification for material steel_block

  begin finite element model mesh1

    Database Name = spring_final.g
    Database Type = exodusII
```

```
  begin parameters for block block_1
    material steel_block
    model = elastic
  end parameters for block block_1

  begin parameters for block block_2
    material steel_bolt
    model = elastic
  end parameters for block block_2

  begin parameters for block block_3
    section = spring_1
  end parameters for block block_3


end finite element model mesh1

### ----------------------- ###
###   Linear Solver         ###
### ----------------------- ###

begin feti equation solver feti
  # Turn on to see feti iterations in the log file.
  #param-string "debugMask" value "solver"
  param-real "damping_coefficient" value 0.0001
  residual norm tolerance = 1e-2
end

begin adagio procedure Apst_Procedure

  begin time control
    begin time stepping block p1
      start time = 0.0
      begin parameters for adagio region adagio
        number of time steps = 1
      end parameters for adagio region adagio
    end time stepping block p1

    termination time = 1.0

  end time control

  begin adagio region adagio

    use finite element model mesh1

    begin Results Output output_adagio
      Database Name = spring_final.e
      At step 0, Increment = 1
      nodal variables = force_contact as force_contact
      nodal variables = force_external as force_external
      nodal variables = force_internal as force_internal
      nodal variables = displacement as displ
      nodal variables = contact_status as celement
      nodal variables = contact_normal_traction_magnitude as cfnor
      nodal variables = contact_tangential_traction_magnitude as cftan
      nodal variables = contact_slip_increment_current as cdtan
      nodal variables = contact_area as carea
      nodal variables = contact_normal_direction as cdirnor
      nodal variables = contact_tangential_direction as cdirtan
      element variables = stress as stress
      element variables = spring_force as spring_force
      element variables = spring_engineering_strain as spring_engineering_strain
      global variables  = timestep as timestep
    end

    ### definition of BCs ###
```

```
begin fixed displacement
  node set = nodelist_3
  component = Z
end

begin fixed displacement
  node set = nodelist_1
  component = X Y Z
end

begin fixed rotation
  #node set = nodelist_101
  rigid body= BOTTOM_FLANGE
  component = X
end

begin fixed rotation
  #node set = nodelist_102
  rigid body = TOP_FLANGE
  component = X
end


begin fixed displacement
  node set = nodelist_2
  component = X Y Z
end

# -------------------- #
#   Contact Surfaces   #
# -------------------- #


begin contact definition
  search = acme

  contact surface surf_101 contains surface_101
  contact surface surf_102 contains surface_102
  contact surface surf_201 contains surface_201
  contact surface surf_202 contains surface_202

  begin constant friction model fric
    friction coefficient = 0.5
  end

  begin interaction
    # surfaces = surf_102 surf_101
    master =surf_102
    slave=surf_101
    friction model       = fric
  end interaction

  begin interaction
    # surfaces = surf_202 surf_201
    master =surf_202
    slave=surf_201
    friction model       = fric
  end interaction

end contact definition

begin solver

  begin loadstep predictor
    type = scale_factor
    scale factor = 0.0 0.0
  end
```

```
        level 1 predictor = none

        begin control contact
          target relative residual        = 1.0e-3
          target residual                 = 1.0e-8
          maximum iterations              = 60
          minimum iterations              = 10
        end

        begin cg
          reference                       = internal
          target     relative residual    = 1.0e-4
          target residual                 = 1.0e-9
          maximum iterations              = 1000
          minimum step length             = 0.001
          maximum step length             = 100.0
          begin full tangent preconditioner
            tangent diagonal scale          = 1.0e-6
            linear solver                   = feti
            conditioning                    = no_check
            small number of iterations      = 15
          end
        end
      end

    end adagio region adagio
  end adagio procedure Apst_Procedure

end Sierra
```

A.10 Automated Adaptive Preloading 3.3

A.10.1 Bolt Preload

```
begin sierra bolt_preload

  #
  #  To be used with artificial strain BC
  #
  begin function bolt_preload
    type is analytic
    expression variable: v = element applied_strain
    evaluate expression = "v"
  end

  begin material linear_elastic
    density         = 5.0
    begin parameters for model elastic
      poissons ratio  = 0.34
      youngs modulus  = 7.427E11
    end parameters for model elastic
  end material linear_elastic

  begin finite element model mesh1
    Database Name = bolt_load_test.g
    Database Type = exodusII

    begin parameters for block
      include all blocks
      material = linear_elastic
      model = elastic
    end

  end finite element model mesh1

  begin presto procedure Apst_Procedure

    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for presto region presto
        end
      end time stepping block p1
      termination time = 0.5e-3
    end time control

    begin presto region presto
      use finite element model mesh1
      ### output description ###
      begin Results Output output_presto
        Database Name = bolt_preload.e
        Database Type = exodusII
        At time 0.0, interval = 1.0e-4
        nodal Variables = displacement
        nodal Variables = force_contact
        element variables = applied_strain
      end results output output_presto

      #
      #  Output the force and strain results from the preload solver
      #  for checking.
      #
      begin history output
        Database Name = bolt_preload.h
        Database Type = exodusII
        at time 0.0, interval = 1.0e-6
        variable = global bolt_force_200
```

```
  variable = global applied_strain_200
  variable = global bolt_force_300
  variable = global applied_strain_300
  variable = global bolt_force_400
  variable = global applied_strain_400
end


#
#  Hold bottom of fixture
#
begin fixed displacement
  surface = surface_100
  components = XYZ
end

#
#  Bolt to fixture contacts
#
begin contact definition
  contact surface bolt2_tied contains surface_210
  contact surface bolt4_tied contains surface_410
  skin all blocks = on
  begin constant friction model med_fric
    friction coefficient = 0.3
  end
  begin interaction defaults
    general contact = on
    friction model = med_fric
  end
  begin interaction
    slave = bolt2_tied bolt4_tied
    master = block_100
    friction model = tied
  end
  initial overlap removal = on
end
#
#  Damping coefficent to aid the dynamic relaxation quasistatic solver
#
begin viscous damping
  include all blocks
  velocity damping coefficient = 1.0e-3
end
#
#  State variables for bolt preload solver subroutine, defined once for all subroutines
#
begin user variable applied_strain
  type = element real length = 1
  use with restart
end
begin user variable bolt_preload_state
  type = element real length = 12
  use with restart
end
#
#  Artificial strain driver, need two of these as bolts have two different orientations.  Alternatively
#  could put this in one block with a variable artificial strain direction, but likely more trouble than it
#  worth
#
begin artificial strain
  block = block_201 block_401
  r function = sierra_constant_function_zero
  s function = sierra_constant_function_zero
  t function = bolt_preload
end
begin artificial strain
  block = block_301
```

```
  r function = sierra_constant_function_zero
  s function = bolt_preload
  t function = sierra_constant_function_zero
end
#
#  Solver blocks for preload.  Internal reaction measures the force in the bolts, iterate
#  the preload to acheive that force.
#
begin user output
  block = block_201
  compute global internal_force_200 as internal reaction in direction 0 0 1
  compute global bolt_force_200 as magnitude of global internal_force_200
  subroutine real parameter:   target_value     = 1.0e+7
  subroutine real parameter:   initial_guess    = -3.06e-4
  subroutine real parameter:   iteration_time   = 5.0e-5
  subroutine string parameter: target_variable  = global bolt_force_200
  subroutine string parameter: working_variable = element applied_strain
  subroutine string parameter: state_variable   = element bolt_preload_state
  element block subroutine = aupst_preload_solver
  compute at every step
  compute global applied_strain_200 as average of element applied_strain
end

begin user output
  block = block_301
  compute global internal_force_300 as internal reaction in direction 0 1 0
  compute global bolt_force_300 as magnitude of global internal_force_300
  subroutine real parameter:   target_value     = 4.0e+6
  subroutine real parameter:   initial_guess    = -8.0e-5
  subroutine real parameter:   iteration_time   = 5.0e-5
  subroutine string parameter: target_variable  = global bolt_force_300
  subroutine string parameter: working_variable = element applied_strain
  subroutine string parameter: state_variable   = element bolt_preload_state
  element block subroutine = aupst_preload_solver
  compute at every step
  compute global applied_strain_300 as average of element applied_strain
end

begin user output
  block = block_401
  compute global internal_force_400 as internal reaction in direction 0 0 1
  compute global bolt_force_400 as magnitude of global internal_force_400
  subroutine real parameter:   target_value     = 1.3e+7
  subroutine real parameter:   initial_guess    = -2.95e-4
  subroutine real parameter:   iteration_time   = 5.0e-5
  subroutine string parameter: target_variable  = global bolt_force_400
  subroutine string parameter: working_variable = element applied_strain
  subroutine string parameter: state_variable   = element bolt_preload_state
  element block subroutine = aupst_preload_solver
  compute at every step
  compute global applied_strain_400 as average of element applied_strain
end

#
#  Optional:  Verify that the target bolt forces were actually obtained at the end of the run.
#  This is present mostly to make this a simple verification test rather than a regression test.
#
begin solution verification
  skip times = 0.0 to 0.49e-3
  verify global bolt_force_200 = 1.0e+7
  verify global bolt_force_300 = 4.0e+6
  verify global bolt_force_400 = 1.3e+7
  relative tolerance = 0.025
  completion file = v1
end
```

```
    end presto region presto
  end presto procedure Apst_Procedure

end sierra bolt_preload
```

A.10.2 Wishbone

```
begin sierra wishbone

  #
  #  To be used with distributed force BC
  #
  begin function solved_force
    type is analytic
    expression variable: v = global applied_force
    evaluate expression = "v"
  end

  begin material steelish
    density         = 0.1
    begin parameters for model elastic_plastic
      poissons ratio  = 0.34
      youngs modulus  = 7.427E11
      yield stress = 3.0e+7
      hardening modulus = 1.0e+10
    end parameters for model elastic_plastic
  end

  begin finite element model mesh1
    Database Name = wishbone.g
    Database Type = exodusII

    begin parameters for block
      include all blocks
      material = steelish
      model = elastic_plastic
    end

  end finite element model mesh1

  begin adagio procedure Apst_Procedure

    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for adagio region adagio
          time increment = 1.0e-5
        end
      end time stepping block p1
      termination time = 2.0e-3
    end time control

    begin adagio region adagio
      use finite element model mesh1
      ### output description ###
      begin Results Output output_adagio
        Database Name = wishbone.e
        Database Type = exodusII
        At time 0.0, interval = 1.0e-4
        nodal Variables = displacement
        element variables = eqps
      end results output output_adagio

      #
      #  Output the force and strain results from the preload solver
      #  for checking.
      #
      begin history output
        Database Name = wishbone.h
        Database Type = exodusII
        at time 0.0, interval = 1.0e-6
        variable = global applied_force
        variable = global curDisp
```

```
  variable = global force_left
  variable = global force_right
end


#
#  make plane stress
#
begin fixed displacement
  include all blocks
  components = z
end
#
#  Add symmetry planes to make statically determinate
#
begin fixed displacement
  node set = nodelist_100
  component = x
end
begin fixed displacement
  node set = nodelist_200
  component = y
end


#
#  Damping coefficent to aid the dynamic relaxation quasistatic solver
#
#begin viscous damping
#  include all blocks
#  velocity damping coefficient = 1.0e-3
#end
#
#  State variables for preload solver subroutine
#
begin user variable applied_force
  type = global real length = 1
  initial value = 0.0
  global operator= max
end
begin user variable preload_solver_state
  type = global real length = 12
  global operator= max
end
#
#  Apply a distributed force to the two end holes to mimic load pins, apply force to reach a target deforma
#
begin distributed force
  node set = nodelist_1
  function = solved_force
  scale factor = -1.0
  component = x
end
begin distributed force
  node set = nodelist_2
  function = solved_force
  scale factor = 1.0
  component = x
end

#
#  Solver blocks for preload.  Tune applied force to reach a target displacement
#
begin user output
  node set = nodelist_1
  compute global disp_left as average of nodal displacement(x)
  compute global force_left as sum of nodal force_external(x)
  compute at every step
```

```
      end
      begin user output
        node set = nodelist_2
        compute global disp_right as average of nodal displacement(x)
        compute global force_right as sum of nodal force_external(x)
        compute at every step
      end

      begin user output
        compute global curDisp from expression "disp_right - disp_left"
        subroutine real parameter:   target_value     = 0.2
        subroutine real parameter:   initial_guess    = 3.0e+6
        subroutine real parameter:   iteration_time   = 2.0e-4
        subroutine string parameter: target_variable  = global curDisp
        subroutine string parameter: working_variable = global applied_force
        subroutine string parameter: state_variable   = global preload_solver_state
        element block subroutine = aupst_preload_solver
        compute at every step
      end

      #
      #  Optional:  Verify that the target bolt forces were actually obtained at the end of the run.
      #  This is present mostly to make this a simple verification test rather than a regression test.
      #
      begin solution verification
        skip times = 0.0 to 1.9e-3
        verify global curDisp = 0.2
        relative tolerance = 0.025
        completion file = v2
      end


    begin solver
      begin cg
        begin full tangent preconditioner
        end
      end
    end

    end adagio region adagio
  end adagio procedure Apst_Procedure

end sierra wishbone
```

A.11 Overlap Removal 3.4

```
#{Ym=64e9}
#{Pr=0.2}
#{E11=64e9}
#{E22=64e9}
#{E33=64e9}
#{NU12=.2}
#{NU13=.2}
#{NU23=.2}

begin Sierra

  begin function ramp1
    type is piecewise linear
    begin values
      0.0 1.0
      1.0e-6 0.75
      2.0e-6 1.0
    end
  end

  define point pt with coordinates 0.0 0.0 0.0
  define point xx with coordinates 1.0 0.0 0.0
  define point yy with coordinates 0.0 1.0 0.0
  define point zz with coordinates 0.0 0.0 1.0
  define point pt1 with coordinates 0.75 0.0 0.0
  define point xx1 with coordinates 1.75 0.0 0.0
  define point yy1 with coordinates 0.75 1.0 0.0
  define point zz1 with coordinates 0.75 0.0 1.0
  define point pt2 with coordinates 1.5 0.0 0.0
  define point xx2 with coordinates 2.5 0.0 0.0
  define point yy2 with coordinates 1.5 1.0 0.0
  define point zz2 with coordinates 1.5 0.0 1.0
  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin coordinate system cylind
    type = cylindrical
    origin = pt
    vector = zz
    point  = xx
  end coordinate system cylind

  begin property specification for material mat
    density          = .5

    begin parameters for model elastic_orthotropic
      youngs modulus = {Ym}
      poissons ratio = {Pr}
      E11 = {E11}
      E22 = {E22}
      E33 = {E33}
      NU12 = {NU12}
      NU13 = {NU13}
      NU23 = {NU23}
      G12 = {E11/(2*(1+NU12))}
      G13 = {E11/(2*(1+NU13))}
      G23 = {E11/(2*(1+NU23))}
      coordinate system = cylind
    end
  end

  # section data -------------------------------------------------------
  begin solid section hexes
  end
```

```
# FE model ----------------------------------------------------------

begin finite element model slender_beam
  database name = overlap_removal.g
  database type = exodusII

  begin parameters for block block_1
    material mat
    model = elastic_orthotropic
    section = hexes
  end parameters for block block_1


  begin parameters for block block_2
    material mat
    model = elastic_orthotropic
    section = hexes
  end parameters for block block_2




end finite element model slender_beam

# procedure data ----------------------------------------------------

begin presto procedure beam_procedure

  begin time control

    begin time stepping block
      start time = 0.0
      begin parameters for presto region beam_region

      end parameters for presto region beam_region
    end time stepping block

    termination time = 2e-6

  end time control

  begin presto region beam_region

    use finite element model slender_beam

    # BC data ----------------------------------------------------------

    begin contact definition
      skin all blocks  = on
      begin interaction defaults
        general contact = on
      end interaction defaults
      BEGIN REMOVE INITIAL OVERLAP
        OVERLAP NORMAL TOLERANCE = 0.1
        OVERLAP TANGENTIAL TOLERANCE =0.1
        OVERLAP ITERATIONS = 500
        DEBUG ITERATION PLOT = off
      END REMOVE INITIAL OVERLAP
    end contact definition


    # BC data ----------------------------------------------------------

    begin results output
      database name = overlap_removal.e
      database type = exodusII
      at time 0.0 increment = 1e-10
```

```
        nodal variables = displacement as displ
        nodal variables = velocity as velo
        nodal variables = damage
        nodal variables = removed_overlap
        element variables = stress_degradation
        element variables = max_principal_strain
        element variables = effective_strain
        element variables = effective_log_strain
        element variables = stress as elem_stress
        element variables = max_principal_stress
        element variables = max_principal_stress_direction as mpsdir
        element variables = log_strain
        element variables = temperature as tempE
        element variables = material_direction_1
        element variables = material_direction_2
        element variables = material_direction_3
        global variables = timestep
        global variables = internal_energy
        global variables = kinetic_energy
        global variables = strain_energy
      end

      # RXNDIFF data -------------------------------------------------------------

    end presto region beam_region

  end presto procedure beam_procedure

end sierra
```

A.11.1 Overlap Removal using Artificial Strain and General Contact

```
#{Ym=64e9}
#{Pr=0.2}
#{E11=64e9}
#{E22=64e9}
#{E33=64e9}
#{NU12=.2}
#{NU13=.2}
#{NU23=.2}

begin Sierra

  begin function ramp1
    type is piecewise linear
    begin values
      0.0 1.0
      1.0e-6 0.75
      2.0e-6 1.0
    end
  end

  define point pt with coordinates 0.0 0.0 0.0
  define point xx with coordinates 1.0 0.0 0.0
  define point yy with coordinates 0.0 1.0 0.0
  define point zz with coordinates 0.0 0.0 1.0
  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0

  begin coordinate system cylind
    type = cylindrical
    origin = pt
    vector = zz
    point  = xx
  end coordinate system cylind

  begin property specification for material mat
    density          = .2

    begin parameters for model elastic_orthotropic
      youngs modulus = {Ym}
      poissons ratio = {Pr}
      E11 = {E11}
      E22 = {E22}
      E33 = {E33}
      NU12 = {NU12}
      NU13 = {NU13}
      NU23 = {NU23}
      G12 = {E11/(2*(1+NU12))}
      G13 = {E11/(2*(1+NU13))}
      G23 = {E11/(2*(1+NU23))}
      coordinate system = cylind
    end
end

  # section data ------------------------------------------------------------
  begin solid section hexes

  end

  # FE model ------------------------------------------------------------------

  begin finite element model slender_beam
    database name = overlap_removal_strain.g
    database type = exodusII

    begin parameters for block block_1
```

```
    material mat
    model = elastic_orthotropic
      section = hexes
  end parameters for block block_1


  begin parameters for block block_2
    material mat
    model = elastic_orthotropic
      section = hexes
  end parameters for block block_2

end finite element model slender_beam

# procedure data -----------------------------------------------------

begin presto procedure beam_procedure

  begin time control

    begin time stepping block p1
      start time = 0.0
      begin parameters for presto region beam_region

      end parameters for presto region beam_region
    end time stepping block p1
    begin time stepping block p2
      start time = 1e-6
      begin parameters for presto region beam_region
      end parameters for presto region beam_region
    end time stepping block p2

    termination time = 2e-6

  end time control

  begin presto region beam_region

    use finite element model slender_beam

    # BC data ----------------------------------------------------------

    begin artificial strain b1
      include all blocks
      direction field material_direction_1 Function = ramp1
    end

    begin contact definition
      active periods = p2
      skin all blocks  = on
      begin interaction defaults
        general contact = on
      end interaction defaults
    end contact definition

    # BC data ----------------------------------------------------------

begin results output
  database name = overlap_removal_strain.e
  database type = exodusII
  at time 0.0 increment = 2e-5
  nodal variables = displacement as displ
  nodal variables = velocity as velo
  nodal variables = damage
  nodal variables = removed_overlap
  element variables = stress_degradation
  element variables = max_principal_strain
  element variables = effective_strain
```

```
    element variables = effective_log_strain
    element variables = stress as elem_stress
    element variables = max_principal_stress
    element variables = max_principal_stress_direction as mpsdir
    element variables = log_strain
    element variables = temperature as tempE
    element variables = material_direction_1
    element variables = material_direction_2
    element variables = material_direction_3
    global variables = timestep
    global variables = internal_energy
    global variables = kinetic_energy
    global variables = strain_energy
  end

    end presto region beam_region

  end presto procedure beam_procedure

end sierra
```

A.12 Remeshing 3.5

```
## mesh step      = {step = 4}
## start time     = {start_time = 0.0}
## end time       = {end_time = 10.0} # large value (solution termination ends each analysis)
## load step size = {load_step_size = 0.00125}

begin sierra  weld_specimen

  title  Weld Tensile Specimen Gage Section

{if (step > 1)}
  restart time = {start_time}
{endif}

  begin function applied_velocity
    type = analytic
    evaluate expression = "0.5;"
  end function applied_velocity

  begin definition for function YOUNGS_MODULUS
    type is piecewise linear
    ordinate is value
    abscissa is temperature
    begin values
       273.0   1.0
      5000.0   1.0
    end values
  end definition for function YOUNGS_MODULUS

  begin definition for function POISSONS_RATIO
    type is piecewise linear
    ordinate is value
    abscissa is temperature
    begin values
       273.0   1.0
      5000.0   1.0
    end values
  end definition for function POISSONS_RATIO

  #
  # ONLY HYPO-ELASTIC MATERIALS
  # WORK IN REMESHING/MAPPING RIGHT NOW
  #
  begin property specification for material mat_1
    density = 1.0
    begin parameters for model BCJ_MEM
      YOUNGS MODULUS = 28.0e6
      POISSONS RATIO = 0.27
      RATE INDEPENDENT YIELD CONSTANT = 232060.
      ISOTROPIC DYNAMIC RECOVERY CONSTANT = 1.0e-4
      ISOTROPIC HARDENING CONSTANT = 71358.5
      DAMAGE EXPONENT = 1.0
      IMPLICIT DAMAGE SOLVER NUMBER OF ITERATIONS = 200.
      IMPLICIT DAMAGE SOLVER RESIDUAL TOLERANCE = 1.e-10
      SEMI IMPLICIT PLASTIC STRAIN SOLVER NUMBER OF ITERATIONS = 1000.
      SEMI IMPLICIT PLASTIC STRAIN SOLVER RESIDUAL TOLERANCE = 1.e-8
      INITIAL DAMAGE = 0.
      YOUNGS MODULUS FUNCTION = YOUNGS_MODULUS
      POISSONS RATIO FUNCTION = POISSONS_RATIO
      TEMPERATURE OPTION = 1.0
      PLASTIC DISSIPATION FACTOR = 0.0
      DENSITY FOR PLASTIC DISSIPATION CALCULATIONS = 1.0
      SPECIFIC HEAT FOR PLASTIC DISSIPATION CALCULATIONS = 1.0
      INITIAL TEMPERATURE FOR UNCOUPLED ADIABATIC HEATING = 273.0
    end parameters for model BCJ_MEM
  end property specification for material mat_1
```

```
  #
  # ONLY UPDATED LAGRANGE WITH MIDPOINT STRAIN INCREMENTATION
  # WORKS IN REMESHING/MAPPING RIGHT NOW
  #
  begin solid section solid_1
    formulation = selective_deviatoric
    deviatoric parameter = 1
    strain incrementation = midpoint_increment
  end solid section solid_1

{if (step == 1)}
  begin finite element model send_fem
    Database name = neckingBar.{step-1}.g
    Database type = exodusII
    begin parameters for block block_1
      material = mat_1
      model = bcj_mem
      section = solid_1
    end parameters for block block_1
  end finite element model send_fem
{else}
  begin finite element model send_fem
    Database name = neckingBar.{step-2}.g
    Database type = exodusII
    begin parameters for block block_1
      material = mat_1
      model = bcj_mem
      section = solid_1
    end parameters for block block_1
  end finite element model send_fem

  begin finite element model recv_fem
    Database name = neckingBar.{step-1}.g
    Database type = exodusII
    begin parameters for block block_1
      material = mat_1
      model = bcj_mem
      section = solid_1
    end parameters for block block_1
  end finite element model recv_fem
{endif}

  begin adagio procedure procedure_1

    begin time control
      begin time stepping block p0
        start time = 0.0
        begin parameters for adagio region region_1
          time increment = {load_step_size}
        end parameters for adagio region region_1
      end time stepping block p0
{if (step > 1)}
      termination time = {start_time}
{else}
      termination time = {end_time}
{endif}
    end time control

    begin adagio region region_1
      use finite element model send_fem

      begin restart data
{if (step > 1)}
        input database name = analysis.{step-1}.rsout
{else}
        output database name = analysis.{step}.rsout
        at time 0.0 interval = {load_step_size}
```

```
{endif}
      end restart data

      # symmetry in y
      begin fixed displacement
        node set = nodelist_3
        components = y
      end fixed displacement

      # symmetry in z
      begin fixed displacement
        node set = nodelist_1
        component = z
      end fixed displacement

      # symmetry in x
      begin fixed displacement
        node set = nodelist_2
        component = x
      end fixed displacement

      # applied velocity in y
      begin prescribed velocity
        node set = nodelist_4
        component = y
        function = applied_velocity
      end prescribed velocity

{if (step == 1)}
      begin user output
        surface = surface_4
        compute global end_disp1 as average of nodal displacement(2)
      end user output

      begin user output
        surface = surface_4
        compute global load as sum of nodal force_internal(2)
      end user output

      begin user output
        compute global eqps_max as max of element eqps
        compute at every step
      end

      #
      # this controls the remeshing interval
      #
      begin solution termination
        terminate global eqps_max >= {step * 0.3} # remesh every time eqps increases by 0.3
        tolerance = 1.0e-6
        terminate type = entire_run
      end solution termination

      begin heartbeat output load_disp_out
        stream name = neckingBar.{step}.dat
        at time 0.0 increment = {load_step_size}
        format = original
        global time
        global end_disp1
        global load
        labels = off
        timestamp format ""
      end heartbeat output load_disp_out

      begin results output adagio_output
        database name = analysis.{step}.e
        database type = exodusii
        at time 0.0 increment = {load_step_size}
```

```
        nodal variables = displacement
        nodal variables = reaction
        nodal variables = force_internal
        nodal variables = force_external
        element variables = stress
        element variables = hydrostatic_stress
        element variables = left_stretch
        element variables = rotation
        element variables = unrotated_stress
        element variables = eqps
        element variables = element_shape
        element variables = nodal_jacobian_ratio
      end results output adagio_output
{endif}

      begin solver
        Begin cg
          reference = external
          target    relative residual = 1.0E-10
          target    residual          = 1.0E-9
          Maximum Iterations          = 2000
          Minimum Iterations          = 3
          begin full tangent preconditioner
            linear solver = feti
            iteration update = 10
          end
        end
      end solver

    end adagio region region_1

  end adagio procedure procedure_1

{if (step > 1)}
  begin adagio procedure procedure_2

    begin procedural transfer migration1
      begin l2_projection transfer fred
        send blocks = block_1
        receive blocks = block_1
        transformation type = element2element
        send coordinates = current
        receive coordinates = original
        linear solver = feti_parallel_direct
      end l2_projection transfer fred
    end procedural transfer migration1

    begin time control
      begin time stepping block p0
        start time = {start_time}
        begin parameters for adagio region region_2
          time increment  = {load_step_size}
        end parameters for adagio region region_2
      end time stepping block p0
      termination time = {end_time}
    end time control

    begin adagio region region_2
      use finite element model recv_fem

      begin restart data
        output database name = analysis.{step}.rsout
        at time 0.0 interval = {load_step_size}
      end restart data

      # symmetry in y
      begin fixed displacement
        node set = nodelist_3
```

```
  components = y
end fixed displacement

# symmetry in z
begin fixed displacement
  node set = nodelist_1
  component = z
end fixed displacement

# symmetry in x
begin fixed displacement
  node set = nodelist_2
  component = x
end fixed displacement

# applied velocity in y
begin prescribed velocity
  node set = nodelist_4
  component = y
  function = applied_velocity
end prescribed velocity

begin user output
  surface = surface_4
  compute global end_disp1 as average of nodal displacement(2)
end user output

begin user output
  surface = surface_4
  compute global load as sum of nodal force_internal(2)
end user output

begin user output
  compute global eqps_max as max of element eqps
  compute at every step
end

#
# this controls the remeshing interval
#
begin solution termination
  terminate global eqps_max >= {step * 0.3} # remesh every time eqps increases by 0.3
  tolerance = 1.0e-6
  terminate type = entire_run
end solution termination

begin heartbeat output load_disp_out
  stream name = neckingBar.{step}.dat
  at time 0.0 increment = {load_step_size}
  format = original
  global time
  global end_disp1
  global load
  labels = off
  timestamp format ""
end heartbeat output load_disp_out

begin results output adagio_output
  database name = analysis.{step}.e
  database type = exodusii
  at time 0.0 increment = {load_step_size}
  nodal variables = displacement
  nodal variables = reaction
  nodal variables = force_internal
  nodal variables = force_external
  element variables = stress
  element variables = hydrostatic_stress
  element variables = left_stretch
```

```
        element variables = rotation
        element variables = unrotated_stress
        element variables = eqps
        element variables = element_shape
        element variables = nodal_jacobian_ratio
      end results output adagio_output

      begin solver
        Begin cg
          reference = external
          target    relative residual = 1.0E-10
          target    residual          = 1.0E-9
          Maximum Iterations          = 2000
          Minimum Iterations          = 3
          begin full tangent preconditioner
            linear solver = feti
            iteration update = 10
          end
        end
      end solver

    end adagio region region_2

  end adagio procedure procedure_2
{endif}

  begin feti equation solver feti
  end

  begin feti equation solver feti_iterative
    param-string "debugMask" value "solver"
    #param-string "local_rbm_tol" value 1.0e-32
    #param-string "global_rbm_tol" value 1.0e-32
    #residual norm tolerance = 1e-13
  end

  begin feti equation solver feti_parallel_direct # for projection
    param-string "debugMask" value "solver"
    corner algorithm = 9
  end

end sierra  weld_specimen
```

A.13 Frame Indifference 3.6

```
Begin Sierra Frame

  #### Title Frame-Indifference Verification Test

  ################################################################################

  define direction x with vector 1 0.0 0.0
  define direction y with vector 0.0 1 0.0
  define direction z with vector 0.0 0.0 1

  DEFINE POINT origin WITH COORDINATES  0.0 0.0 0.0
  DEFINE POINT along_z WITH COORDINATES 0.0 0.0 1.0
  DEFINE POINT along_x WITH COORDINATES 1.0 0.0 0.0
  DEFINE AXIS z_axis WITH POINT origin DIRECTION z

#### Define Functions


  Begin Function Rotate
    Type = Analytic
    Evaluate Expression = "1.57079632679"
  End Function Rotate

  Begin Function art_strain_X
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Strain
    Begin Values
      0.00  0.00
      1.00  1E-3
    End Values
  End Function art_strain_X

  Begin Function art_strain_Y
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Strain
    Begin Values
      0.00  0.00
      1.00  0.00
    End Values
  End Function art_strain_Y

  Begin Function art_strain_Z
    Type = Piecewise Linear
    Abscissa = Time
    Ordinate = Strain
    Begin Values
      0.00  0.00
      1.00  0.00
    End Values
  End Function art_strain_Z

################################################################################

#### Define Material Properties

#### Steel
  Begin Property Specification For Material Steel
    density = 7871.966988
    Begin Parameters for Model Elastic_Plastic
      Youngs Modulus = 1.999479615E+11
      Poissons Ratio = 0.33333
      Yield Stress = 275790291.7
      Hardening Modulus = 275790291.7
```

```
    end Parameters For Model Elastic_Plastic
    Artificial Engineering Strain X Function = art_strain_X
    Artificial Engineering Strain Y Function = art_strain_Y
    Artificial Engineering Strain Z Function = art_strain_Z
  end Property Specification For Material Steel


################################################################################

#### Define Finite Element Model

  Begin Finite Element Model block_rotate
    Database Name = Frame_Ind.g
    Database Type = ExodusII

#### Define Blocks

    Begin Parameters For Block block_1
      Material Steel
      Model = Elastic_Plastic
    End Parameters For Block block_1
  End Finite Element Model block_rotate

################################################################################

  Begin Adagio Procedure calculations

    #### Define Time and Time Step

    Begin Time Control
      Begin Time Stepping Block Timestep1
        Start Time = 0.0
        Begin Parameters For Adagio Region Problem
          #Step Interval = 100
          Number of time steps = 50
        End Parameters For Adagio Region Problem
      End Time Stepping Block Timestep1
      Begin Time Stepping Block Timestep2
        Start Time = 1.0
        Begin Parameters For Adagio Region Problem
          #Step Interval = 100
          Number of time steps = 50
        End Parameters For Adagio Region Problem
      End Time Stepping Block Timestep2
      Termination Time = 2.0
    End Time Control

################################################################################

    Begin Adagio Region Problem
      Use Finite Element Model block_rotate

#      Begin restart data restart
#        Restart Time = 1.0
#        INPUT DATABASE NAME = %B.rst
#        OUTPUT DATABASE NAME = %B_1.rst
#        At Time 0.0 Increment = 0.1
#      End restart data restart

################################################################################

#### BCs

      Begin Fixed Displacement
        node set = nodeset_6 nodeset_4
        #Block = block_1
        Components = Y X Z
        Active Periods = Timestep1 Timestep2
```

```
    End Fixed Displacement

    Begin Prescribed Velocity
      Surface = sideset_1
      #Block = block_1
      #Rigid Body = rigidbody_1
      Cylindrical Axis = z_axis
      Function = Rotate
      Scale Factor = 1
      Active Periods = Timestep2
    End Prescribed Velocity


###############################################################################

    Begin User Output
      compute global max_abs_stress_xx as max absolute value of element stress(xx)
      compute global max_abs_stress_yy as max absolute value of element stress(yy)
      compute global max_abs_stress_zz as max absolute value of element stress(zz)
      compute global max_abs_stress_xy as max absolute value of element stress(xy)

      compute global stressxxnorm from expression "max_abs_stress_xx/3E8"
      compute global stressyynorm from expression "max_abs_stress_yy/3E8"
      compute global stresszznorm from expression "max_abs_stress_zz/3E8" #0.1088
      compute global stressxynorm from expression "max_abs_stress_xy/3E8" #1.023E6

      #compute global stressxxnorm from expression "(stress_xx)/abs(max_stress_xx)"
      #compute global max_stress_xx as max of element stress(xx)
      #compute global max_stress_yy as max of element stress(yy)
      #compute global max_stress_zz as max of element stress(zz)
      #compute global max_stress_xy as max of element stress(xy)
    End User Output



    Begin Heartbeat Output normalized
      Stream Name = FrameInd.csv
      Format = SpyHis
      Start Time = 0.0
      At Time 0 Increment = 0.005
      Termination Time = 2.0
      Global stressxxnorm as stressxxnorm
      Global stressyynorm as stressyynorm
      Global stresszznorm as stresszznorm
      Global stressxynorm as stressxynorm
    End Heartbeat Output normalized

    Begin Results Output block_spin_output
      Database Name = Frame_Ind.e
      Database Type = ExodusII
      #At Time 0.0 Increment = 0.1
      At Step 0 Increment = 2
      Nodal Variables = Acceleration As Accel
      Nodal variables = Velocity As Vel
      Nodal Variables = Displacement As Displ
      Nodal Variables = Force_External As Force
      Element Variables = Stress As Stress
      Element Variables = Log_Strain As logstra
      Element Variables = Von_Mises As VonMises
      Element Variables = Effective_Log_Strain As ELS
      Global Variables = stressxxnorm as stressxxnorm
      Global Variables = stressyynorm as stressyynorm
      Global Variables = stresszznorm as stresszznorm
      Global Variables = stressxynorm as stressxynorm

      #Global Variables = max_abs_stress_xx as max_stress_xx
      #Global Variables = max_abs_stress_yy as max_stress_yy
      #Global Variables = max_abs_stress_zz as max_stress_zz
```

```
        #Global Variables = max_abs_stress_xy as max_stress_xy

      End Results Output block_spin_output

################################################################################

      begin solver
        begin cg
          target relative residual  = 1.0E-5
          maximum iterations  = 1000
          acceptable residual = 1.0e+10
          begin full tangent preconditioner
          end full tangent preconditioner
        end
      end

################################################################################

    End Adagio Region Problem
  End Adagio Procedure calculations
End Sierra Frame
```

Distribution

1	0899	Technical Library, 9536 (1 electronic)

Sandia National Laboratories