SANDIA REPORT

Analysis of Modeling Parameters on Threaded Screws

Miquela S. Vigil, Matthew R. W. Brake, and Douglas J. VanGoethem

![Sandia National Laboratories logo] Sandia National Laboratories

Analysis of Modeling Parameters on Threaded Screws

Miquela S. Vigil,[1,2] Matthew R. W. Brake,[1] and Douglas J. VanGoethem[1]

[1] Component Science and Mechanics
Sandia National Laboratories
MS 1070; P.O. Box 5800
Albuquerque, NM 87185-1070

[2] Department of Mechanical Engineering
New Mexico Institute of Mining and Technology
801 Leroy Place
Socorro, NM 87801

Abstract

Assembled mechanical systems often contain a large number of bolted connections. These bolted connections (joints) are integral aspects of the load path for structural dynamics, and, consequently, are paramount for calculating a structure's stiffness and energy dissipation properties. However, analysts have not found the optimal method to model appropriately these bolted joints. The complexity of the screw geometry cause issues when generating a mesh of the model. This paper will explore different approaches to model a screw-substrate connection. Model parameters such as mesh continuity, node alignment, wedge angles, and thread to body element size ratios are examined. The results of this study will give analysts a better understanding of the influences of these parameters and will aide in finding the optimal method to model bolted connections.

Acknowledgment

Contents

Figures

Tables

1   Introduction

Numerical modeling through the use of finite element, finite difference, or finite volume codes has become very prevalent in the analysis of structures and components. Although many numerical models produce valid results and ultimately are an effective tool in the area of solid mechanics, many analysts still encounter issues that result from limitations of the codes. Code developers continuously research these issues and work towards eliminating these limitations. Meanwhile, it is important for analysts to be knowledgeable of the limitations of the codes used and how those limitations effect the numerical results of the simulation. The lack of information available to analysts about code limitations is an issue many analysts face. Often, when a developer or an analyst explore these issues further they do not have time to conduct a thorough study or write a formal report on their findings. This becomes an issue for many analysts because when they encounter similar issues they do not have references to gain a better understanding of the issue and as a result do not know how to proceed without conducting a study. This report aims to address some of these issues by documenting best practices and lessons learned from developing models of fasteners.

The failure of fasteners is a critical area to examine in the solid mechanics analysis of a system. Some of the components modeled by the analyst can consist of hundreds of small fasteners. In a model of the whole system it is unrealistic to include every small fastener. However, the stiffness and support that the fastener provides to the design is needed for the system model. The question that arises from this issue is, how can a fastener be incorporated into a model of a component that consists of many fasteners? Before this question can be answered, a more complete understanding of the failure mechanisms of a fastener is needed.

In this study, a threaded screw is modeled in Sierra/Solid Mechanics, also known as Adagio, an implicit nonlinear finite element code developed by Sandia National Laboratories [1]. Currently, most finite element models of screws within larger systems do not include the threads of the screws, which adds an overly burdensome level of complexity when creating a finite element model. Several modeling methods exists for generating a finite element model of a detailed screw. Although these methods would simplify the overall modeling of a screw, the limitations of these approaches and the effects they have on the numerical results are unclear. To gain a better understanding of these effects, a parametric study is conducted on a simplified finite element model of a screw in what follows.

This parametric study explores the differences between a model containing a tied contact to combine two solids and a model of a single solid. In this section a short explanation of the contact algorithm created in Adagio is given. The idea of the contact algorithm is to control the interaction between two solids using specified physics parameters [2]. These parameters, which are based on physics models, make sure that the two solids do not interpenetrate. The type of interaction between surfaces, such as being tied together or sliding, is specified as part of this algorithm [2].

Constraints are defined depending on the interaction of the two solids. There are three types of contact constraints:

1. Node/Face - prevents nodes of one solid from penetrating the faces of the other solid

2. Face/Face - prevents faces of one solid from penetrating the faces of the other solid

3. Node/Analytic Surface - prevents nodes of one solid from penetrating the predefined surfaces of the other solid

Once the contact surfaces are defined then the constraints need to be enforced. The constraints are enforced by defining master and slave surfaces. To eliminate gaps as well as interpenetration of the two solids, all the nodes on the surface of the slave surface are forced to lie on the surface of the master surface. Figure 1 illustrates how the contact algorithm defines the classification of the contact nodes. There are three different classes of nodes for the two solids that are in contact. The active contact nodes are identified by the contact search algorithm. These nodes are subsets of the potential contact nodes which are those associated with the master-slave interaction [1]. The remaining nodes are defined as non-active nodes. The contact algorithm follows the steps shown in Figure 2. The diagram gives a general understanding of the process that the algorithm undergoes to identify the nodes, enforce the contact constraints, and implement the type of contact.
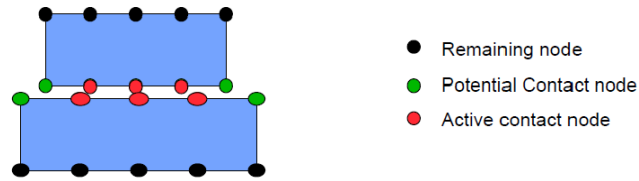
Figure 1. The classification of contact nodes [1].
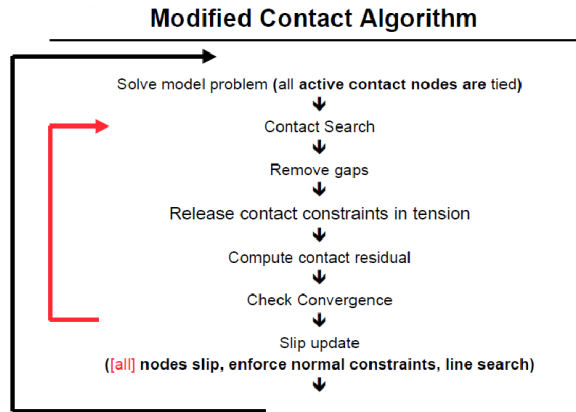
Figure 2. The diagram outlines the steps and problem solving method of the modified contact algorithm [1].

2 Modeling Approach

2.1 Geometry and Materials

This parametric study is based on a finite element solid mechanics model of a detailed #4-40 screw with simplified geometry. A symmetrical modeling technique is utilized allowing a fraction of the screw and substrate to be modeled. Hence the overall geometry is in the shape of a wedge. Instead of modeling the full length of the screw, only four threads in a substrate are analyzed. Another modification made to this model to eliminate complexity is the elimination of the thread angle. Unlike a realistic screw, the thread angle is not included in this model. The geometric model, which is produced in Cubit, can be seen in Fig. 3. An example of the journal file for this model can be found in Appendix A.
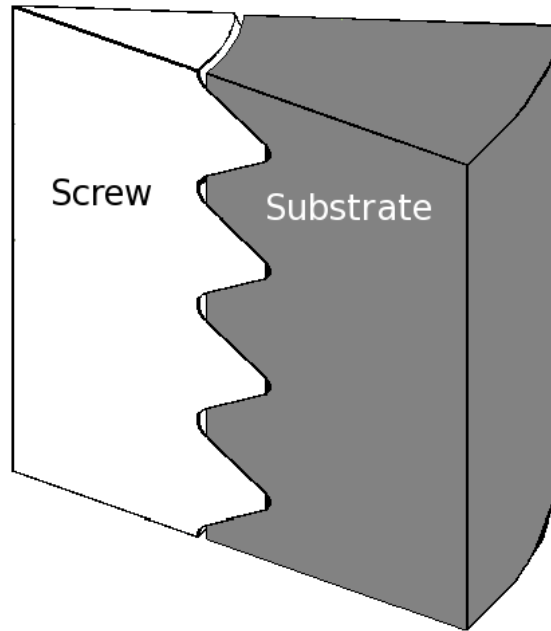
Figure 3. Geometric model of the screw and substrate.

The materials chosen for the screw and substrate are based on common materials used in many components: an alloy A-286 screw and a 304L stainless steel substrate. Material properties can be found in Table 1 The elastic-plastic solid mechanics model is applied to both the screw and substrate; and a frictionless contact interaction is applied between the threads of the screw and substrate even though the interaction between the threads of a screw and substrate is not frictionless.

Table 1. Material Properties

	Screw - Alloy A-286	Substrate - 304L Stainless Steel
Young's Modulus (psi)	29E6	28E6
Poisson's Ratio	0.28	0.28
Yield Stress (psi)	120E3	32.7E3
Ultimate Tensile Strength (psi)	160E3	78E3

2.2 Boundary Conditions

Kinematic boundary conditions are applied to sets of nodes in the screw-substrate model. Fixed displacement boundary conditions are applied to node sets 1 and 3. Table 2 and Fig. 4 show the type of boundary condition applied to each node set.

Table 2. Boundary Conditions Applied at each Node Set

Node Set	Type of Boundary Condition	Component
1	Fixed Displacement	Z
2	Fixed Displacement	Tangential
3	Fixed Displacement	X
3	Fixed Displacement	Y
3	Fixed Displacement	Z

The fixed displacement boundary condition sets the specified nodes to have a zero displacement in the specified directions. Therefore the outer diameter of the substrate (node set 3) is fixed in the X, Y, and Z axes. Node set 1 is fixed in the z direction. A prescribed displacement of 0 with respect to the center axis (y-axis) is applied to node set 2. This allows the node set 2 to be fixed in the tangential direction while allowing the nodes to move in the radial direction.

2.3 Loading

In many scenarios, fasteners are exposed to a range of mechanical loading conditions, that can be reduced to components of tensile and shear loading. For simplicity, only pure tension is considered in these simulations. To incorporate the loading on the screw, a prescribed displacement that is incrementally increased from 0 to 0.025 inches is applied to node set 10 in the Y-axis. The total displacement is the same length as the pitch of the thread of the screw. The load is applied to the top surface of the screw causing the screw to be pulled out of the substrate. Fig. 4 highlights the surface of node set 10. The screw at a displacement of 0.025 inches can be seen in Fig. 5.
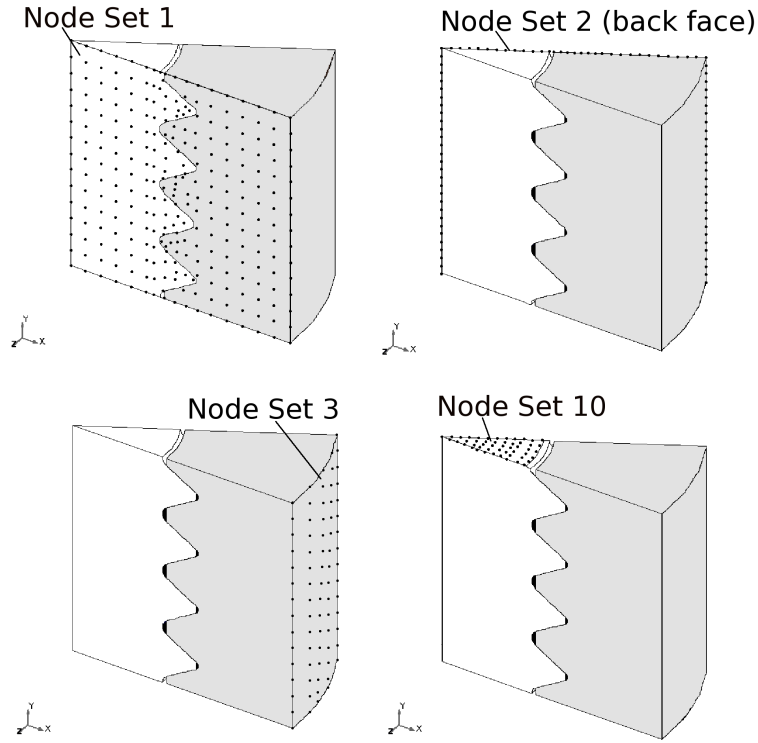
Figure 4. The respective node sets are labeled on the geometric model. Node sets 1 and 2 are the symmetrical surfaces of the wedge, node set 3 is the surface on the outer diameter of the substrate, and node set 10 is the surface on the top of the screw.

2.4 Mesh Convergence

Before analyzing the outcomes from the various modeling methods of interest, the model needs to be meshed correctly. A convergence study is used to determine the optimal mesh for the model. During this study the simulation is run with increasingly fine meshes until the numerical results converged. This is achieved by analyzing the force versus displacement data (the system's compliance curve). The convergence study is conducted for both the model with a uniform mesh and the model with a mesh that is connected with a tied contact. Table 3 lists the element sizes used in this study. The convergence of the force-displacement curves for both the model with a unified mesh and the model with tied contacts are illustrated in Figs. 6 and 7.
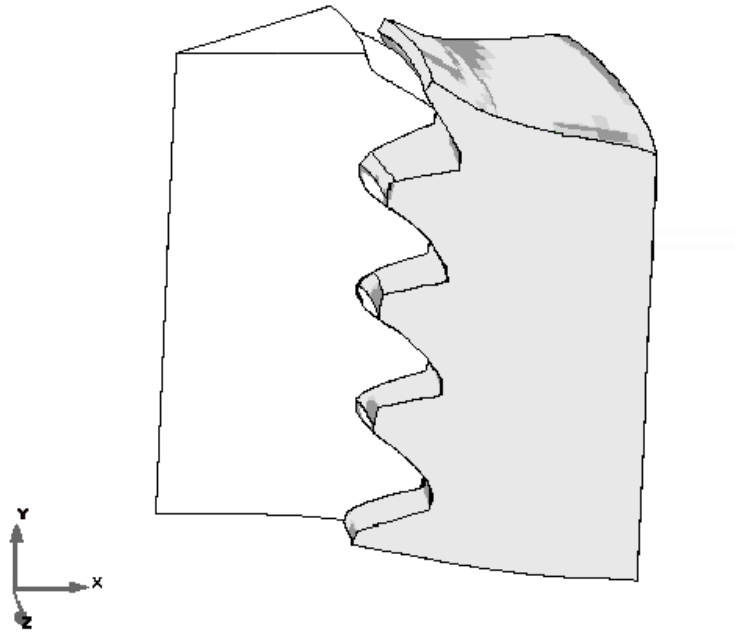
Figure 5. Screw displaced at the maximum displacement of 0.025 inches.

Table 3. Element Sizes in Convergence Study

Refinement Level	Element Size	Number of Elements
1	0.0035	8224
2	0.0025	14,948
3	0.0015	24,344
4	0.0005	905,216
5	0.00025	17,198,808
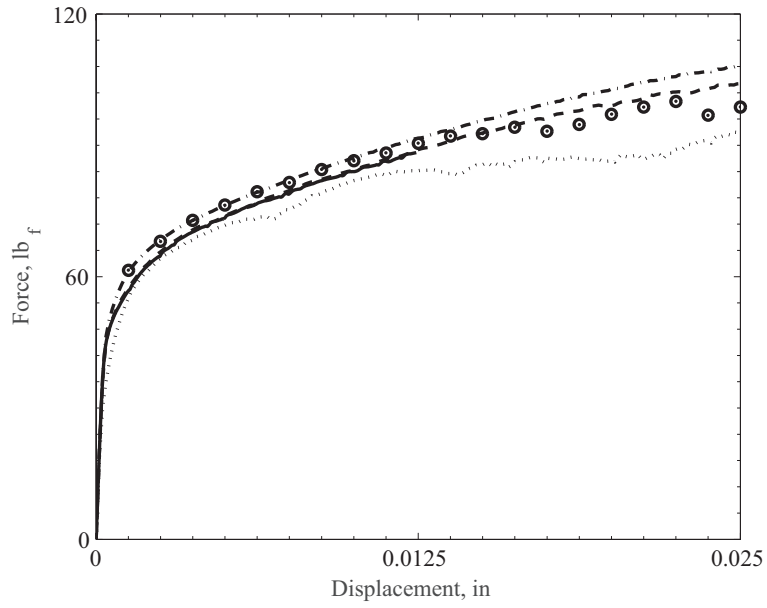
Figure 6. Mesh convergence for the model with a uniform mesh for Refinement Level 1 (○), Refinement Level 2 ($- \cdot -$), Refinement Level 3 ($\cdots$), Refinement Level 4 ($--$), Refinement Level 5 ($-$).
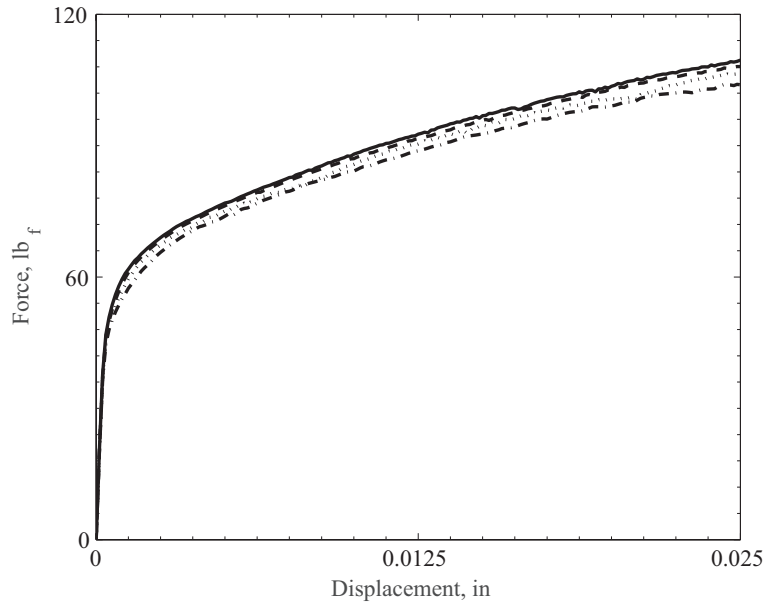
Figure 7. Mesh convergence for the model with tied meshes for Refinement Level 1 ($-$), Refinement Level 2 ($--$), Refinement Level 3 ($\cdots$), Refinement Level 4 ($- \cdot -$).

3 Modeling Parameters

Multiple modeling approaches are of interest in the finite element analysis of a screw because the complexity of the geometry of the threads adds difficulty in obtaining a quality mesh. As a result, various approaches are investigated to find the optimal modeling method. For the approaches considered, there are several parameters in these methods that may or may not influence the numerical results. To gain a better understanding of the effects of these parameters, a series of numerical tests are performed. This section discusses the following modeling parameters that are investigated:

1. Mesh Continuity

2. Node Alignment

3. Symmetrical Modeling

4. Ratio of Meshes

3.1 Mesh Continuity

When creating a finite element model of a detailed screw several meshing issues emerge. The helical shape of the threads cause the nodes of the shank and the nodes of the threads to be misaligned. As a result, the issues occur when attempting to mesh the volume. One approach to overcome this problem would be to create two separate solids, one being the shank and the other being the threads, and then connect the solids together using a tied contact. This will allow the threads and the shank to be meshed individually, eliminating the issues that arise from the node misalignment. However, since the screw is not meshed as one solid, the accuracy of this model needs to be assessed.

Two models are used to analyze the effects of the tied contact and to determine the differences between a model with individual meshes that are tied together and a model with a contiguous mesh. Both models are similar to that discussed in Section 2. The major difference between the two models is the solid modeling of the geometries in Cubit. The model that consists of the uniform mesh is made up of two solids; the screw and the substrate. The other model with the tied meshes consists of four solids; the screw shank, the threads of the screw, the body of the substrate, and the threads of the substrate. These differences are shown in Fig. 8.

Several contact interactions are applied to the model to connect the threads to the respective bodies. The threads of the screw and substrate are defined as the masters in each interaction. Therefore, the shank of the screw and the body of the substrate are the slaves. The tied friction model is used to connect the screw body with the threads of the screw. Similarly, the body of the substrate is connected to the threads of the substrate. The contact algorithm and constraint formulation is dependent on the simulation type. The use of the contact algorithm ties the solids of the threads and bodies together while guaranteeing that they will not interpenetrate.
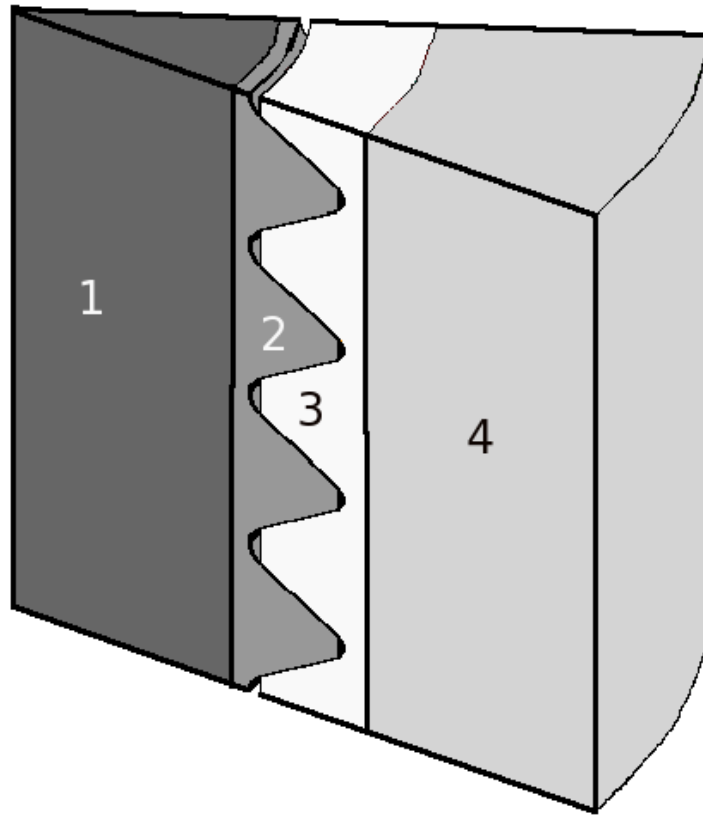
Figure 8. The four bodies defined in the model with tied contacts;
(1) screw body, (2) threads of screw, (3) threads of substrate, and
(4) substrate body.

3.2 Node Alignment

As discussed in Section 2, the thread angle causes the nodes between the threads and the shank to be misaligned. To ensure that the misalignment of the nodes does not influence the numerical results, several simulations with models of various node alignments are analyzed. The model of the screw and substrate that consists of the tied contacts is modified so that the nodes between the threads and the bodies are misaligned. Because the node misalignment can be in multiple directions three different models are used.

1. Axial Misalignment

2. Circumferential Misalignment

3. Both Axial and Circumferential Misalignment
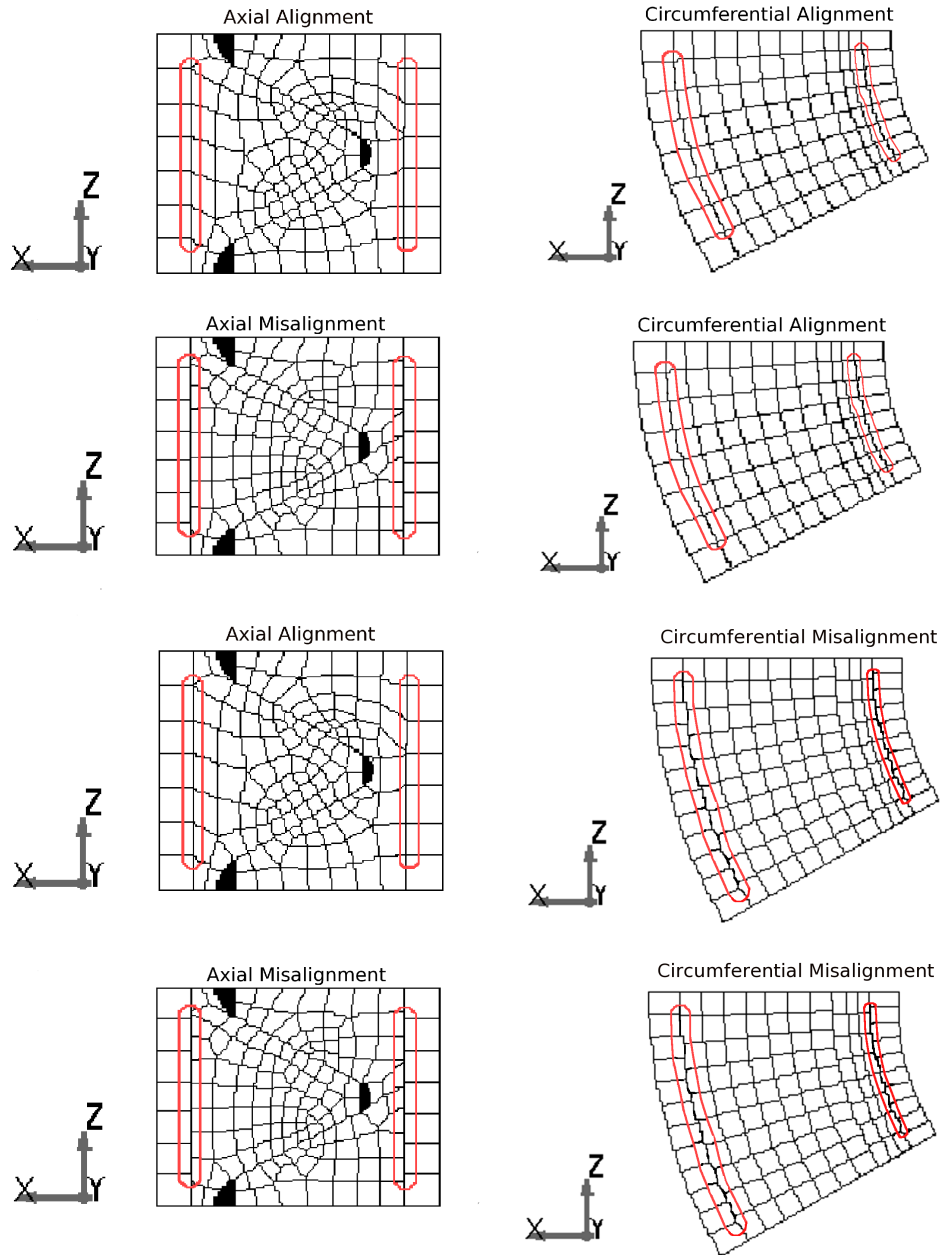
Figure 9. The four different meshes with the respective node misalignment

Figure 9 demonstrates the three different meshes for each node alignment case compared to the mesh that has axial and circumferential node alignment.

3.3 Symmetrical Modeling

A common approach used in FEA to reduce the number of degrees of freedom in a problem is to use symmetry relationships. In this case, since a screw is cylindrical, a fraction of the geometry can be modeled instead. As a result, a wedge that is symmetrical about the Y-axis is modeled. This modeling approach requires the user to apply boundary conditions at the symmetrical surfaces. Although this modeling approach eliminates computational time and cost, other effects can influence the results. One major concern with this approach is the influence of edge effects on the model. To find the optimal wedge angle five different angles (listed in Table 4) are compared. The element size for all of these models is identical with a size of 0.0015 inches determined from the previous convergence studies.

Table 4. Wedge Angles

Wedge Angle (°)
15
30
45
90
360

3.4 Ratio of Meshes

The previously listed studies use a consistent thread-to-body mesh ratio of one. This means that the element size between the threads and the bodies are the same. One question an analyst may ask is would the mesh be more optimal if the mesh became coarser towards the center of the screw or the body of the substrate (i.e. if the mesh of the threads are refined)? To answer this question, four different thread-to-body element size ratios are analyzed to find the optimal mesh ratio (reported in Table 5 and Figure 10).

Table 5. Ratio of Meshes

Thread to Body Ratio	Element Size - Thread	Element Size - Body
1	0.0005	0.0005
1/2	0.0005	0.001
1/4	0.0005	0.002
1/6	0.0005	0.003
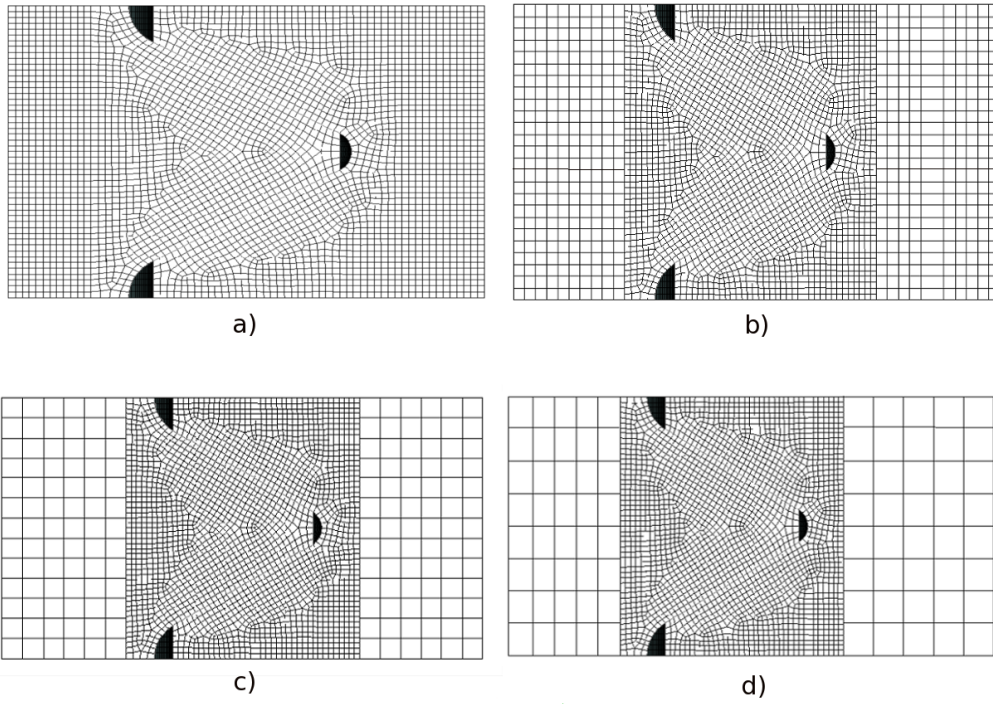
Figure 10. The difference in thread to body meshes for Case 1 - 1/1 (a), Case 2 - 1/2 (b), Case 3 - 1/4 (c), Case 4 - 1/6 (d).

4   Results and Discussion

4.1   Mesh Continuity

Figure 11 compares the global force with respect to the screw displacement for the model with a uniform mesh and the model with tied contacts between the threads and the body. As seen in Fig. 11, the two curves are coincident. For the finely meshed models there is no difference between the uniform mesh and the tied mesh. The maximum Von Mises stresses for both models can also be observed in Fig. 12. Likewise, the Von Mises stress in each model is very similar (illustrated in Fig. 13).
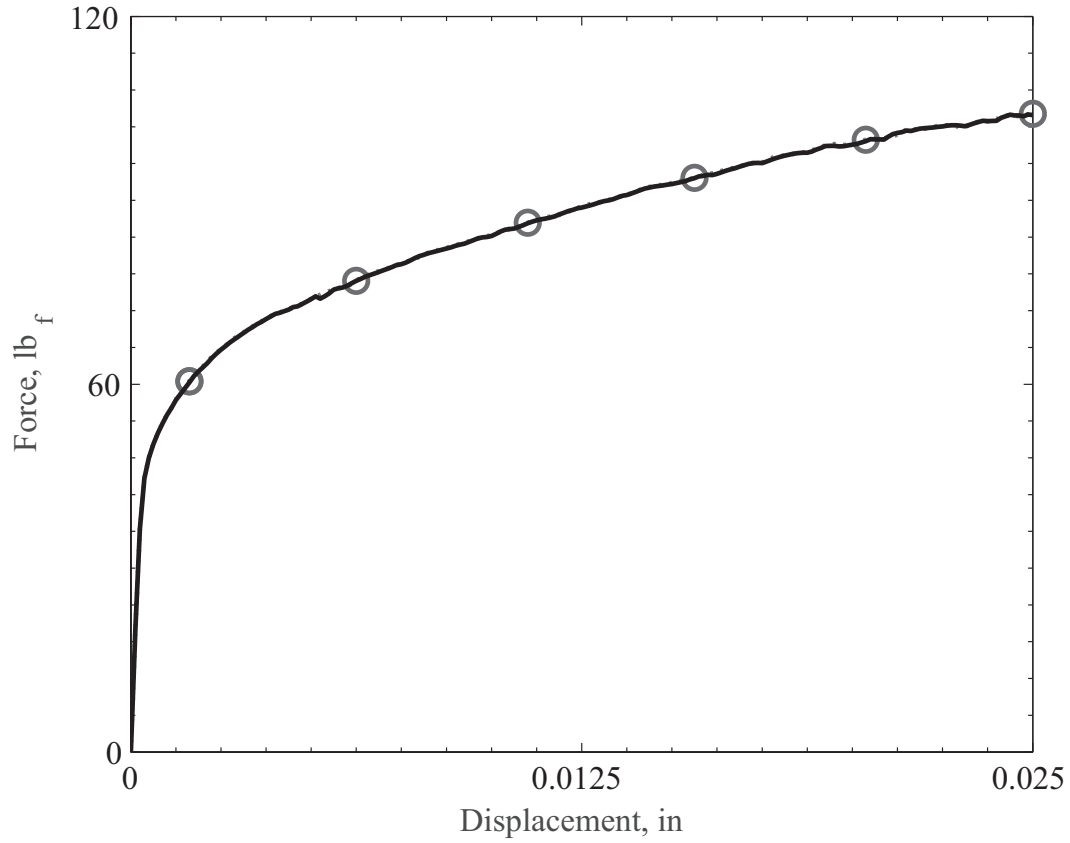
Figure 11. Force with respect to displacement for the model with the uniform mesh ($\circ$) compared to the model with the tied meshes ($-$).
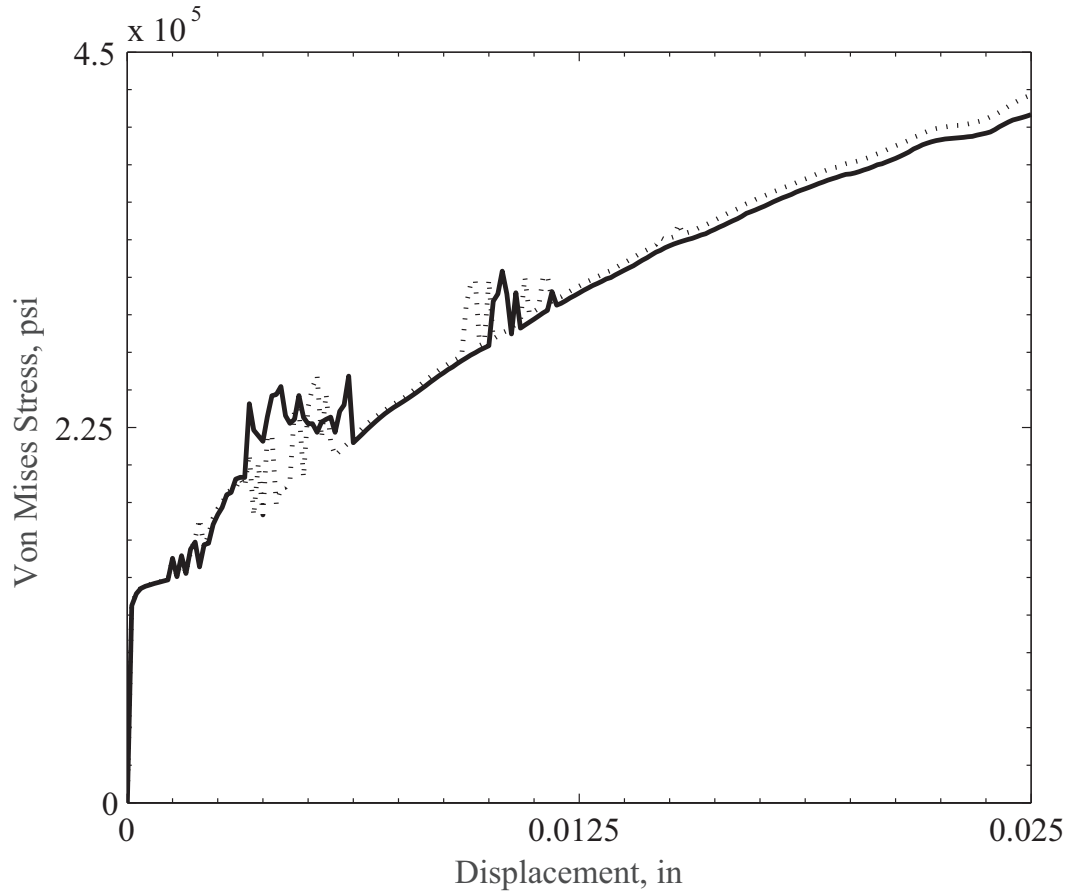
Figure 12. Maximum Von Mises stress for model with the uniform mesh $(--)$ compared to the model with the tied meshes $(\cdots)$.

4.2 Node Alignment

The global force for the three different cases of node misalignment are compared with that of the model with node alignment. Figure 14 shows that node alignment does not influence the global force-displacement response of the finite element analysis for the cases considered. Thus, using meshes with misaligned nodes does not present an issue for the accuracy of the analysis.

4.3 Symmetrical Modeling

Five models with different wedge angles (angle about Y-axis) are compared in Fig. 15. The models with wedge angles of 15° and 360° have the largest differences. Whereas the models with wedge angles 30°, 45°, and 90° are very similar. The results of the 360° case are somewhat surprising as the other cases appear to converge to a different value. As the actual screw consists of a helix angle, it is not immediately clear which wedge angle is more appropriate, and some further
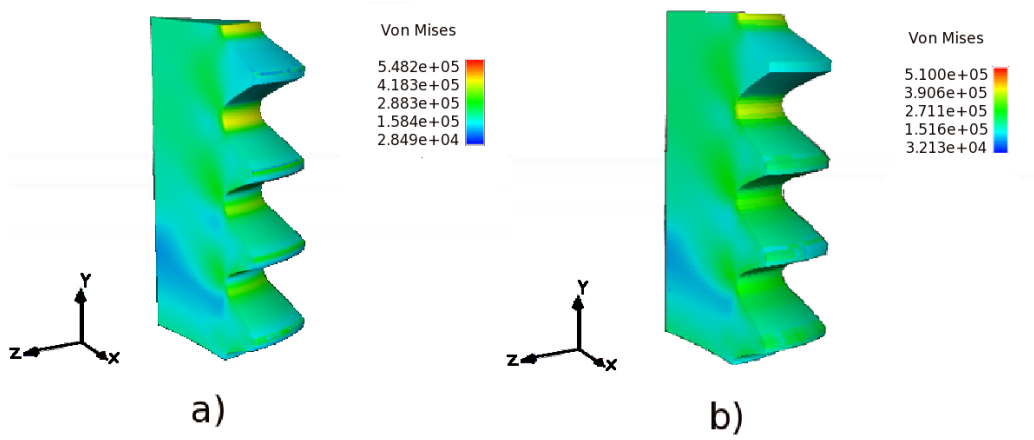
Figure 13. Countour plot of Von Mises Stress in the screw for contiguous mesh (a) and tied mesh (b).

study is warranted once validation data is obtained. However, one conclusion from the present analysis is that if symmetry is used, a wedge angle between 30° and 90° is the most suitable.

4.4 Ratio of Meshes

The thread-to-body element size ratio on a converged mesh does not appear to have an appreciable influence on the numerical solution. Fig. 16 shows that the force with respect to displacement is the same for thread-to-body element size ratios of 1/1, 1/2, and 1/4. The numerical solution begins to diverge at a thread-to-body ratio of 1/6. Therefore, mesh refining can be used to improve computational performance without a loss of accuracy (at least for the refinement ratios 1/1, 1/2, and 1/4 considered in the present analysis).

21
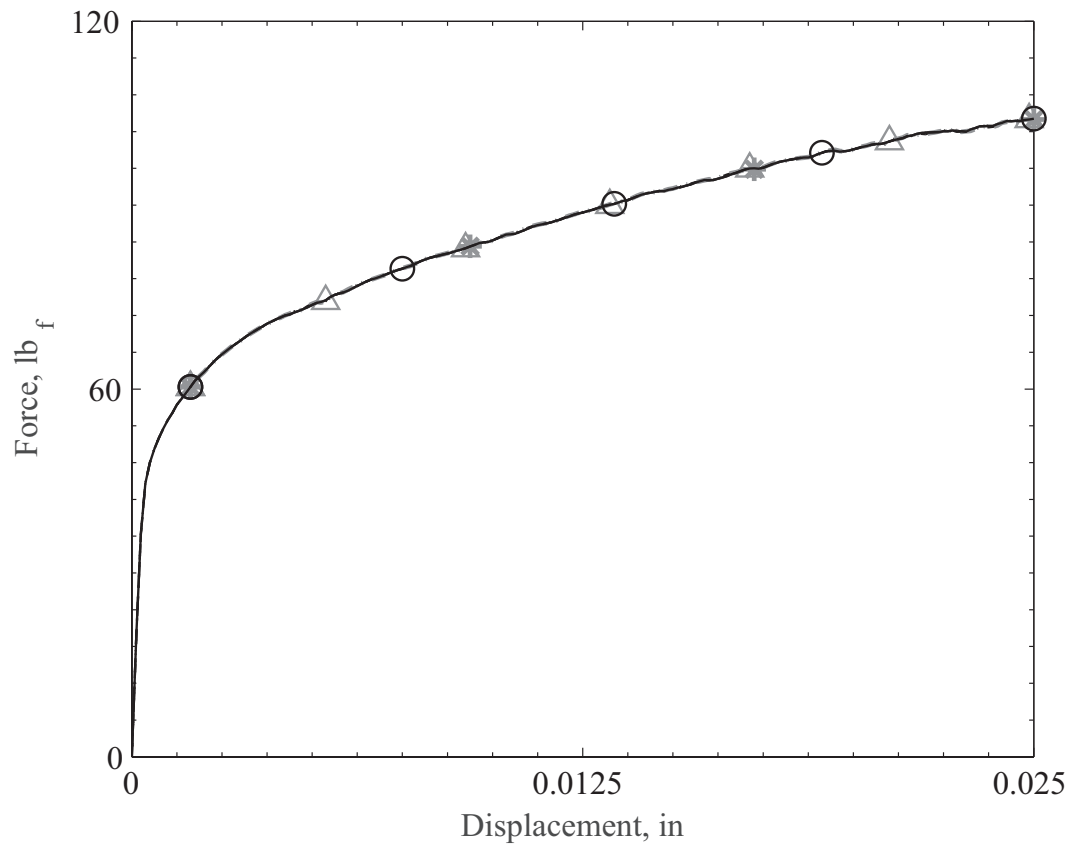
Figure 14. Force with respect to the screw displacement for Case 1- both axial and circumferential alignment ($\triangle$), Case 2 - axial misalignment ($*$), Case 3 - circumferential misalignment ($-$), and Case 4 - both axial and circumferential misalignment ($\circ$).
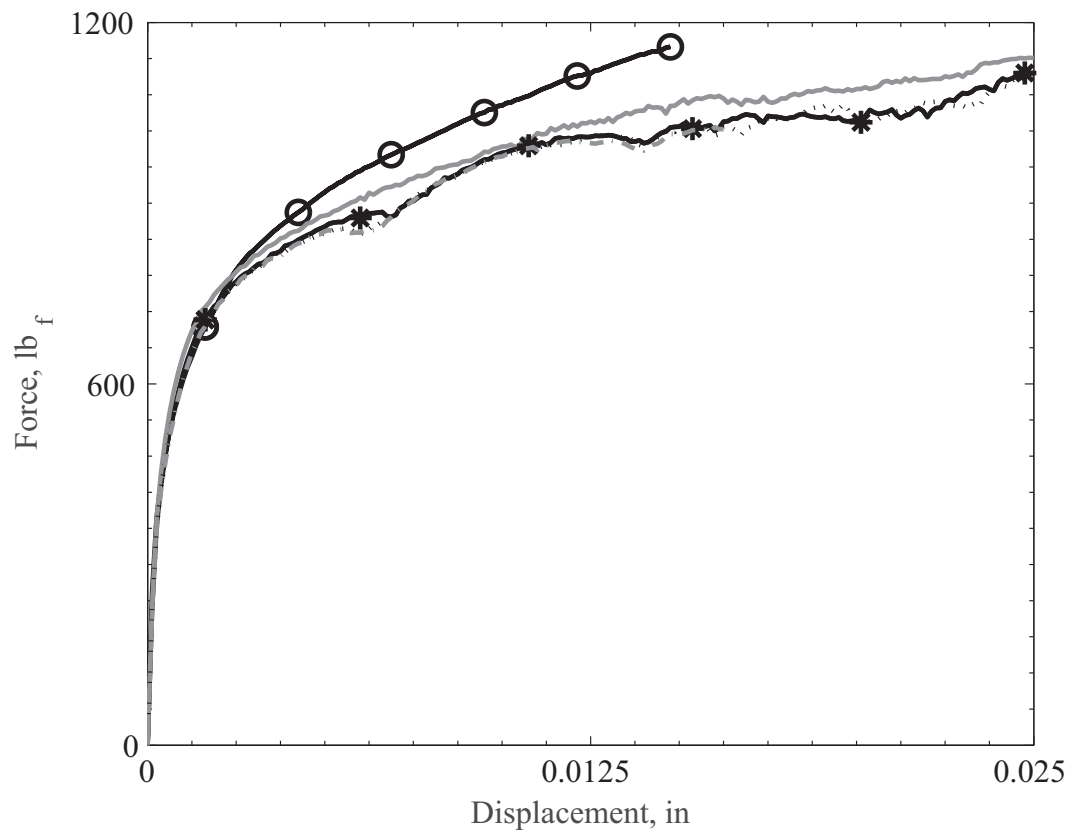
Figure 15. Force with respect to displacement for wedge angles:
15° (−), 30° (∗), 45° (⋯), 90° (− · −), 360° (∘).

23
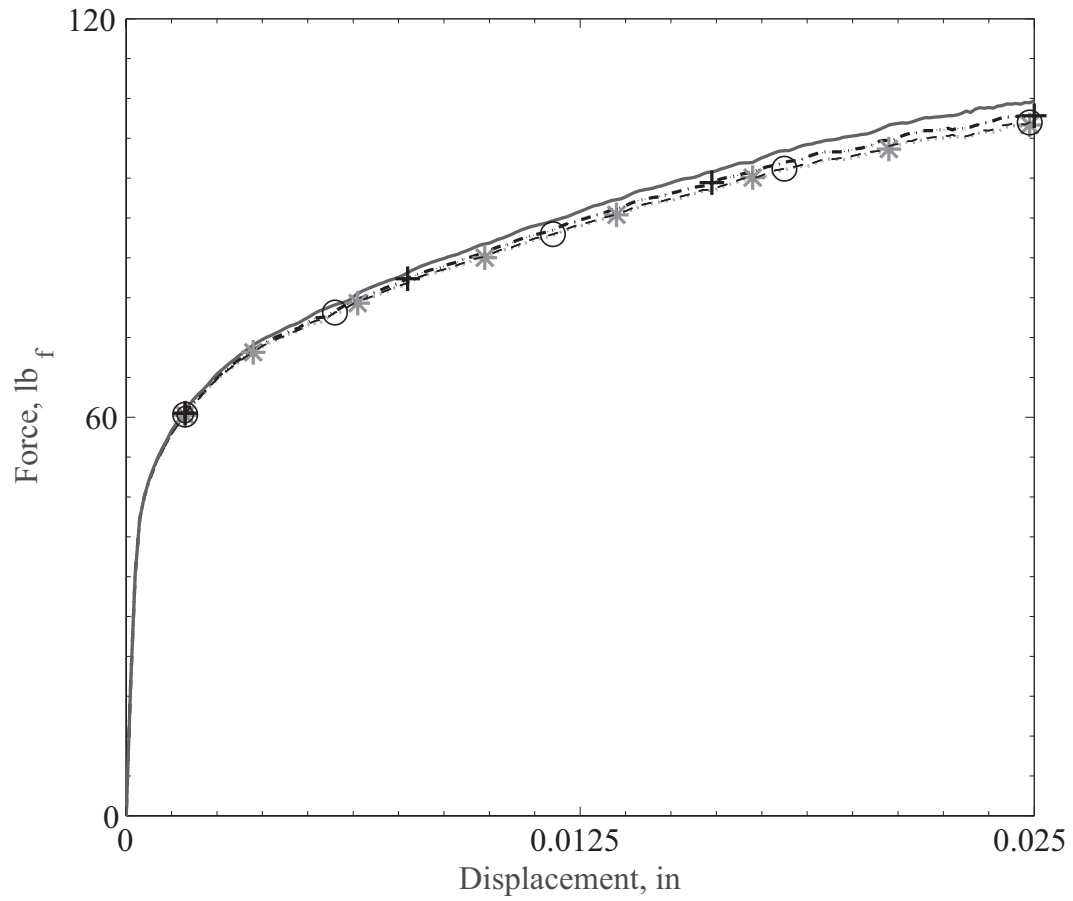
Figure 16. Force with respect to the screw displacement for threadtobody element size ratios: 1/6 (−), 1/4 (+), 1/2 (○), 1 (∗).

5 Conclusions

This report presents an in-depth analysis of the phenomenologically different methods in which a screw can be modeled using SIERRA. The approaches considered include:

- uniform meshing versus tied contact,

- aligned versus misaligned nodes,

- reduced models via symmetry versus complete models,

- biased versus unbiased meshes.

From these analyses, several major conclusions are drawn, which are anticipated to be able to be applicable for other models developed in SIERRA:

1. Meshes with tied contact exhibit better convergence properties than continuous (uniform) meshes.

2. The alignment of nodes for tied contact does not appreciably affect the accuracy of the model.

3. In modeling an axial-symmetric system, wedge angles between $30°$ and $90°$ yield congruent results. Smaller wedge angles are not recommended without further convergence analysis/refinement.

4. Refined ratios up to a factor of four are found to have no appreciable affect on the results.

References

[1] Sierra/SolidMechanics 4.34 User's Guide (2014). Sierra/SM. Sandia National Laboratories, Albuquerque, NM.

[2] K.H. Pierson, B.W. Spencer, and T.M. Hensley. Adagio Contact Algorithm Enhancements. SAND2007-3003C.

Appendix A: Example Cubit Journal Code

```
# This program generates a meshed model of a fraction of a screw in a substrate. Note that the thread an-
gle is not incorporated.This model is similar to those used in the parametric study discussed in the pa-
per.
# Create mesh for thread having Diameter number 4, Length number 5,
# TPI 40, and substrate inclusion ratio 0.3333337
# Axis-symmetry Wedge Geometry of Threaded Screw
# Created by DJV & MSV June 2014


### parameters
# wedge˙angle = 30
# nthread = 4
# esize1 = .0005
## esize2 = 2*esize1   # used for coarsening away from thread

# nut˙router = .1   outer radius for material added onto thread for ''nut'' substrate
#
reset
# INPUT PARAMETERS
#----------------------------------------------------------------------------------------------------------------
-----------------------------

#diameterNumber = 4
#lengthNumber = 5
#tpi = 40

# OPTIONAL PARAMETERS OF THIS FUNCTION
#----------------------------------------------------------------------------------------------------------------
------------------------------
#ScrewLength =    lengthNumber*0.0625
#capRadius =   (1/2)*(2.1625e-2*diameterNumber+9.275e-2)
#capHeight =  1.1719e-2*diameterNumber+6.5104e-2
# p =1/tpi
# dRatio = 1/3
# Dmaj = 0.060 + diameterNumber*0.013
# Rmaj = Dmaj/2
# Dmin = Dmaj-5*sqrt(3)*p/8
# Rmin = Dmin/2
# d = dRatio*(Rmaj-Rmin)
# Rs = p*sqrt(3)/12

##SCREW THREADS
#·······························································································
#
```

generate points for threads

r1 = Rmin - d
r2 = Rmin - Rs/2
r3 = Rmin + Rs/2
r4 = Rmin
r5 = Rmaj
r6 = Rmaj
r7 = Rmin
r8 = Rmin + p*sqrt(3)/24
r9 = Rmin - p*sqrt(3)/24
r10 = Rmin - d

y1 = 0
y2 = 0
y3 = 0
y4 = p/8
y5 = 7*p/16
y6 = 9*p/16
y7 = 7*p/8
y8 = p
y9 = p
y10 = p

Create vertices
create vertex r1   y1   0
vertex Id("vertex") name 'v1'
create vertex r2   y2   0
vertex Id("vertex") name 'v2'
create vertex r3   y3   0
vertex Id("vertex") name 'v3'
create vertex r4   y4   0
vertex Id("vertex") name 'v4'
create vertex r5   y5   0
vertex Id("vertex") name 'v5'
create vertex r6   y6   0
vertex Id("vertex") name 'v6'
create vertex r7   y7   0
vertex Id("vertex") name 'v7'
create vertex r8   y8   0
vertex Id("vertex") name 'v8'
create vertex r9   y9   0
vertex Id("vertex") name 'v9'
create vertex r10   y10   0
vertex Id("vertex") name 'v10'

#Create curves
create curve Vertex v1 Vertex v2
curve Id("curve") name 'c1'
create curve arc center v3 v2 v4
curve Id("curve") name 'c2'
create curve Vertex v4 Vertex v5
curve Id("curve") name 'c3'
create curve Vertex v5 Vertex v6
curve Id("curve") name 'c4'
create curve Vertex v6 Vertex v7
curve Id("curve") name 'c5'
create curve arc center v8 v7 v9
curve Id("curve") name 'c6'
create curve Vertex v9 Vertex v10
curve Id("curve") name 'c7'
create curve Vertex v10 Vertex v1
curve Id("curve") name 'c8'
create surface c1 c2 c3 c4 c5 c6 c7 c8
surface Id("surface") name 's1'
#
sweep surface with name 's1' yaxis angle wedge˙angle
surface Id("surface") name 's1otherside'
volume Id('volume') name 'vscrewthread'

 # want to try creating curve and sweeping surface along curve in order to have named curve for circumferential interval setting

SHANK
#······························
add interior volume of screw here
create vertex   0   0   0
vertex Id("vertex") name 'v201'
create vertex   0   p   0
vertex Id("vertex") name 'v210'
create surface vertex v1 v10 v210 v201
surface Id("surface") name 's2'
sweep surface with name 's2' yaxis angle wedge˙angle
surface Id("surface") name 's2otherside'
volume Id('volume') name 'vscrewbody'

NUT THREADS
#······························
Verticies for nut thread

 # Vertices needed for inner thread
r11 = Rmin
y11 = 0
r12 = Rmin
y12 = p
r13 = Rmaj + d
y13 = 0
r14 = Rmaj + d
y14 = p
r15 = Rmaj-p*sqrt(3)/48
y15 = p/2

create vertex   r11   y11   0
vertex Id("vertex") name 'v11'
create vertex   r4   y4   0
vertex Id("vertex") name 'v4'
create vertex   r5 y5   0
vertex Id("vertex") name 'v5'
create vertex   r6 y6   0
vertex Id("vertex") name 'v6'
create vertex   r15   y15   0
vertex Id("vertex") name 'v15'
create vertex   r7   y7   0
vertex Id("vertex") name 'v7'
create vertex   r12   y12   0
vertex Id("vertex") name 'v12'
create vertex   r14   y14   0
vertex Id("vertex") name 'v14'
create vertex   r13   y13   0
vertex Id("vertex") name 'v13'

#
create curve Vertex v11 Vertex v4
curve Id("curve") name 'c101'
create curve Vertex v4 Vertex v5
curve Id("curve") name 'c102'
create curve arc center v15 v5 v6
curve Id("curve") name 'c103'
create curve Vertex v6 Vertex v7
curve Id("curve") name 'c104'
create curve Vertex v7 Vertex v12
curve Id("curve") name 'c105'
create curve Vertex v12 Vertex v14
curve Id("curve") name 'c106'

```
create curve Vertex v14 Vertex v13
curve Id("curve") name 'c107'
create curve Vertex v13 Vertex v11
curve Id("curve") name 'c108'
create surface c101 c102 c103 c104 c105 c106 c107 c108
surface Id("surface") name 's201'
#
sweep surface with name 's201' yaxis angle wedge˙angle
surface Id("surface") name 's201otherside'
volume Id('volume') name 'vnutthread'
```

##SUBSTRATE
```
#············································
```
add interior volume
```
create vertex  nut˙router      0       0
vertex Id("vertex") name 'v313'
create vertex  nut˙router      p       0
vertex Id("vertex") name 'v314'
create surface vertex v13 v14 v314 v313
surface Id("surface") name 's202'
sweep surface with name 's202' yaxis angle wedge˙angle
surface Id("surface") name 's202otherside'
volume Id('volume') name 'vnutbody'
del free vertex all
```

#MESHING DETAILS
```
#········································
```
Set meshing details prior to copying multiple threads
Set parameters for paving the cross section
```
curve c3 c5  interval  31
curve c104 c102 interval 31
curve c1 c7  interval  4
#curve c8     interval 36
#curve c107 interval 36
curve c4     interval  6
curve c25 c58 interval 36
```

#set circumferential intervals
--¿ may want to figure out way to create curve and sweep surf along curve to control intervals using named curve, this should work for now
```
# circ˙sthread˙int = 32
# circ˙nthread˙int =32
curve 21 interval circ˙sthread˙int    #screw thread circ intervals
curve 30 interval circ˙sthread˙int    # screw body circ intervals
curve 46 interval circ˙nthread˙int    #nut thread circ intervals
```

curve 68 interval circ˙nthread˙int   *#nut body circ intervals*

 # do manual pick here and use name for BC to apply to arbitrary number of threads
Surface 11 1 16 26 name 'swedgeBC'
Surface 28 name 'snutOD˙BC'

vol with name 'vscrewthread*' size esize1
vol with name 'vnutthread*' size esize1
vol with name 'vscrewbody*' size esize1
vol with name 'vnutbody*' size esize1

mesh vol all

copy volumes to make screw mesh
Volume all copy move y .025 repeat nthread - 1
imprint vol with name 'vscrewbody*'
merge vol with name 'vscrewbody*'
imprint vol with name 'vscrewthread*'
merge vol with name 'vscrewthread*'
imprint vol with name 'vnutbody*'
merge vol with name 'vnutbody*'
imprint vol with name 'vnutthread*'
merge vol with name 'vnutthread*'

blocks and sets
block 1 vol with name 'vscrewbody*'
block 2 vol with name 'vscrewthread*'
block 3 vol with name 'vnutbody*'
block 4 vol with name 'vnutthread*'

nodeset 1 surf with z˙coord ¿ 0.0 - 1e-6 and z˙coord ¡ 0.0 + 1e-6   *# Z plane symmetry BC*
 # No easy way to make symmetry BC nodeset on angled face for arbitrary number of threads
 # Use names manually defined prior to copy
nodeset 2 surf with name 'swedgeBC*'
nodeset 3 surf with name 'snutOD˙BC*'

 # need to update .025 to calculated pitch parameter
nodeset 10 surf in vol with name 'vscrew*' with y˙coord ¿ .025*nthread - 1e-6
                                and y˙coord ¡ .025*nthread + 1e-6
 # top of screw for loading
sideset 10 surf in vol with name 'vscrew*' with y˙coord ¿ .025*nthread - 1e-6

and y˙coord ¡ .025*nthread + 1e-6
top of screw for loading

#export mesh using aprepro parameters in file name
line = "export genesis 'TM4meshed˙screw˙wedge"//tostring(wedge˙angle)//"˙nthread"//tostring(nthread)//".g'
 block all overwrite"
rescan(line)

Appendix B: Example Adagio Code

```
# Simple screw thread pullout simulation, quasistatic, wedge symmetry
# Created by DJV & MSV June 2014
# Units are in inches
# quasistatic, prescribed displacement with contact
# adapt contact and solver settings
from ../adagio˙rtest/VerificationTestManual/ContactVerificationTests/HertzSphereContact/

# Timing
# t0 = 0   start of qs loading
# t1 = 1   maximum load reached
# t2 = 2   return load to zero


# disp = 0.025


### QS solution parameters
#nstep˙up = 200
#nstep˙down = 200


# set flags to determine run type
# kinematic = 0   faceFace = 1


## variables from regression test
# numThetaCollections = numberThetaCollections = 5  # Number of Collection Points Around Cir-
cumference of Contact Patch


## Variables


# degreeInterval      = degreeInterval = 90.0 / (numberThetaCollections - 1)
# degreeTolerance     = degreeTolerance = degreeInterval / 2.0
ifdef(kinematic)
# constraint          = constraint˙formulation = "node˙face"
# contactAlgorithm    = contact˙alg = "kinematic node-Face"
else
ifdef(faceFace)
# constraint          = constraint˙formulation = "face˙face"
# contactAlgorithm    = contact˙alg = "augmented Lagrange face-Face"
else
# constraint          = constraint˙formulation = "node˙face"
# contactAlgorithm    = contact˙alg = "augmented Lagrange node-Face"
endif
endif
```

BEGIN SIERRA

TITLE screw thread pullout, wedge symmetry, contact˙alg

```
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 point origin with coordinates 0.0 0.0 0.0
define axis center˙axis with point origin direction y

begin definition for function const1
  type     = constant
  begin values
    1.0
  end values
end
begin definition for function const0
  type     = constant
  begin values
    0.0
  end values
end

BEGIN DEFINITION FOR FUNCTION ramp
  TYPE = PIECEWISE LINEAR
  BEGIN VALUES
    t0     0.0
    t1     1.0
    2.0     1.0
  END
END

BEGIN DEFINITION FOR FUNCTION rampupdown
  TYPE = PIECEWISE LINEAR
  BEGIN VALUES
    t0     0.0
    t1     1.0
    t2     0.0
    10.0     0.0
  END
END
```

#-------------- materials-------------#

BEGIN PROPERTY SPECIFICATION FOR MATERIAL A286˙screw
  DENSITY = 0.286/(32.174*12)
   begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 29e6
     POISSONS RATIO = 0.28
     YIELD STRESS = 120e3
    HARDENING MODULUS = 90.698e4 # ult = 160ksi, elong = 12%, eu˙frac = .5, epUTS = 5.4%
    END
 END

BEGIN PROPERTY SPECIFICATION FOR MATERIAL SS303˙screw
  DENSITY = 0.289/(32.174*12)
   begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 28e6
     POISSONS RATIO = 0.25
     YIELD STRESS = 34.8e3
    HARDENING MODULUS = 34.94e4   # ult = 89.9ksi, elong = 50%, eu˙frac = .5, epUTS = 22.2%
    END
 END

BEGIN PROPERTY SPECIFICATION FOR MATERIAL PH13-8˙H1100
  DENSITY = 0.279/(32.174*12)
   begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 28.8e6
     POISSONS RATIO = 0.28
     YIELD STRESS = 135e3
    HARDENING MODULUS = 39.483e4   # ult = 150ksi, elong = 14%, eu˙frac = .5, epUTS = 6.3%
    END
 END

BEGIN PROPERTY SPECIFICATION FOR MATERIAL hiperco50
  DENSITY = 0.293/(32.174*12)
   begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 30e6
     POISSONS RATIO = 0.3
     YIELD STRESS = 30e3
    HARDENING MODULUS = 65.644e4 # ult = 53ksi, elong = 4%, eu˙frac = 1, epUTS = 3.8%
    END
 END

BEGIN PROPERTY SPECIFICATION FOR MATERIAL SS304L˙VAR

```
    DENSITY = 0.289/(32.174*12)
  begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 28e6
     POISSONS RATIO = 0.28
     YIELD STRESS = 32701.85
    HARDENING MODULUS = 23.205e4   # ult = 78ksi, elong = 80%, eu˙frac = .5, epUTS = 33.5%
    END
  END

  BEGIN PROPERTY SPECIFICATION FOR MATERIAL titanium6Al4V
    DENSITY = 0.16/(32.174*12)
   begin parameters for model elastic˙plastic
     YOUNGS MODULUS = 16e6
     POISSONS RATIO = 0.31
     YIELD STRESS = 125e3
    HARDENING MODULUS = 38.464e4   # ult = 135ksi, elong = 10%, eu˙frac = .5, epUTS = 4.1%
    END
  END




#---- sections -------------------------------------------------------------

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

BEGIN FINITE ELEMENT MODEL femodel
 DATABASE NAME =TM4meshed˙screw˙wedge30˙nthread4.g

  BEGIN PARAMETERS FOR BLOCK block˙1    #screw˙body
    MATERIAL A286˙screw
    SOLID MECHANICS USE MODEL elastic˙plastic
  END
  BEGIN PARAMETERS FOR BLOCK block˙2    #screw˙threads
    MATERIAL A286˙screw
    SOLID MECHANICS USE MODEL elastic˙plastic
  END
  BEGIN PARAMETERS FOR BLOCK block˙3    #nut˙body
    MATERIAL SS304L˙VAR
    SOLID MECHANICS USE MODEL elastic˙plastic
  END
  BEGIN PARAMETERS FOR BLOCK block˙4    #nut˙threads
    MATERIAL SS304L˙VAR
    SOLID MECHANICS USE MODEL elastic˙plastic
  END
END
```

```
#========== QS Disp Procedure ==============
  BEGIN ADAGIO PROCEDURE PULL

   BEGIN TIME CONTROL
    BEGIN TIME STEPPING BLOCK tb1
     START TIME = t0
     BEGIN PARAMETERS FOR ADAGIO REGION rPULL
       TIME INCREMENT = 1/nstep˙up
     END
    END

    BEGIN TIME STEPPING BLOCK tb2
     START TIME = t1
     BEGIN PARAMETERS FOR ADAGIO REGION rPULL
       TIME INCREMENT = 1/nstep˙down
     END
    END

    TERMINATION TIME = t2
   END

   ### ----------------- ###
   ###    Region         ###
   ### ----------------- ###

   BEGIN ADAGIO REGION rPULL
    USE FINITE ELEMENT MODEL femodel

    ### ---------------------- ###
    ###   BCs and Loading  ###
    ### ---------------------- ###

    BEGIN FIXED DISPLACEMENT
     node set = nodelist˙3  # OD of nut
     COMPONENTS = X
    END
    BEGIN FIXED DISPLACEMENT
     node set = nodelist˙3  # OD of nut
     COMPONENTS = Y
    END
    BEGIN FIXED DISPLACEMENT
     node set = nodelist˙3  # OD of nut
     COMPONENTS = Z
    END
```

```
BEGIN FIXED DISPLACEMENT
  node set = nodelist˙1   #symetry face
  COMPONENTS = Z
END
BEGIN prescribed displacement
  node set = nodelist˙2
  cylindrical axis = center˙axis
  function = const0
END

# loading
BEGIN prescribed displacement
  node set = nodelist˙10
  COMPONENT = Y
  function = rampupdown
  scale factor = disp
END


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

BEGIN RESULTS OUTPUT 3Doutput
 DATABASE NAME = %B.e
  AT STEP 0 INCREMENT = 1
  #AT time t0 INCREMENT =
  nodal variables = displacement as displ
  nodal variables = force˙contact
  element variables = von˙mises
  element variables = eqps
END

BEGIN heartbeat OUTPUT heartOut
  stream name = %B.hrt
  AT STEP 0 INCREMENT = 1
  precision = 8
  labels = OFF
  legend = ON
  VARIABLE = GLOBAL time
  VARIABLE = GLOBAL Dinp
  VARIABLE = GLOBAL Fr
  VARIABLE = GLOBAL maxVM˙screw˙thread
  VARIABLE = GLOBAL maxVM˙screw˙body
  VARIABLE = GLOBAL maxVM˙nut˙thread
```

```
  VARIABLE = GLOBAL maxVM˙nut˙body
  VARIABLE = GLOBAL maxEQPS˙screw˙thread
  VARIABLE = GLOBAL maxEQPS˙screw˙body
  VARIABLE = GLOBAL maxEQPS˙nut˙thread
  VARIABLE = GLOBAL maxEQPS˙nut˙body



END

BEGIN HISTORY OUTPUT hisoutput
  DATABASE NAME = %B.h
  AT STEP 0 INCREMENT = 1
  VARIABLE = GLOBAL TIMESTEP AS tstep
  VARIABLE = GLOBAL wall˙clock˙time as twall
  VARIABLE = GLOBAL wall˙clock˙time˙per˙step as twallps
  VARIABLE = GLOBAL KINETIC˙ENERGY AS EK
  VARIABLE = GLOBAL HOURGLASS˙ENERGY AS EHG
  VARIABLE = GLOBAL INTERNAL˙ENERGY AS EI
  VARIABLE = GLOBAL EXTERNAL˙ENERGY AS EE
  VARIABLE = GLOBAL contact˙energy AS EC
  VARIABLE = GLOBAL Dinp
  VARIABLE = GLOBAL Finp
  VARIABLE = GLOBAL Fr
  VARIABLE = GLOBAL maxVM˙screw˙thread
  VARIABLE = GLOBAL maxVM˙screw˙body
  VARIABLE = GLOBAL maxVM˙nut˙thread
  VARIABLE = GLOBAL maxVM˙nut˙body
  VARIABLE = GLOBAL maxSzz˙screw˙thread
  VARIABLE = GLOBAL maxSzz˙screw˙body
  VARIABLE = GLOBAL maxSzz˙nut˙thread
  VARIABLE = GLOBAL maxSzz˙nut˙body
  VARIABLE = GLOBAL maxEQPS˙screw˙thread
  VARIABLE = GLOBAL maxEQPS˙screw˙body
  VARIABLE = GLOBAL maxEQPS˙nut˙thread
  VARIABLE = GLOBAL maxEQPS˙nut˙body


END

BEGIN USER OUTPUT
  node set = nodelist˙10
  COMPUTE GLOBAL Dinp AS average OF nodal displacement(2)
  COMPUTE GLOBAL Finp AS sum OF nodal reaction(2)
END
```

```
BEGIN USER OUTPUT
  node set = nodelist˙3
  COMPUTE GLOBAL Fr AS sum OF nodal reaction(2)
END

BEGIN USER OUTPUT
  block = block˙1
  COMPUTE GLOBAL maxVM˙screw˙body AS max OF element von˙mises
  COMPUTE GLOBAL maxSzz˙screw˙body AS max OF element stress(zz)
  COMPUTE GLOBAL maxEQPS˙screw˙body AS max OF element eqps
END
BEGIN USER OUTPUT
  block = block˙2
  COMPUTE GLOBAL maxVM˙screw˙thread AS max OF element von˙mises
  COMPUTE GLOBAL maxSzz˙screw˙thread AS max OF element stress(zz)
  COMPUTE GLOBAL maxEQPS˙screw˙thread AS max OF element eqps
END
BEGIN USER OUTPUT
  block = block˙3
  COMPUTE GLOBAL maxVM˙nut˙body AS max OF element von˙mises
  COMPUTE GLOBAL maxSzz˙nut˙body AS max OF element stress(zz)
  COMPUTE GLOBAL maxEQPS˙nut˙body AS max OF element eqps
END
BEGIN USER OUTPUT
  block = block˙4
  COMPUTE GLOBAL maxVM˙nut˙thread AS max OF element von˙mises
  COMPUTE GLOBAL maxSzz˙nut˙thread AS max OF element stress(zz)
  COMPUTE GLOBAL maxEQPS˙nut˙thread AS max OF element eqps
END

### ---------------------- ###
### Contact Definition ###
### ---------------------- ###

begin contact definition screw
search = dash
    enforcement = al
  contact surface screw˙thread contains block˙2
  contact surface screw˙body contains block˙1
contact surface nut˙body contains block˙3
  contact surface nut˙thread contains block˙4

  begin tied model tie˙1
  end tied model tie˙1
```

```
      begin interaction int˙0
        friction model = frictionless
        master = nut˙thread
        slave = screw˙thread
      end interaction int˙0


begin interaction int˙1
master = screw˙thread
slave = screw˙body
friction model = tie˙1
end interaction int˙1


begin interaction int˙2
master = nut˙thread
slave = nut˙body
friction model = tie˙1
end interaction int˙2


    end contact definition screw


#     begin contact definition cont1
#       search = dash
#       contact surface screw˙thread contains block˙2
#       contact surface nut˙thread contains block˙4
#       enforcement = al
#       begin interaction int˙0
#         friction model = frictionless
#         master = nut˙thread
#         slave = screw˙thread
#         constraint formulation = constraint˙formulation
#       end interaction int˙0
#       compute contact variables = on
#     end contact definition cont1


    ### ---------------------- ###
    ###   Solver Definition    ###
    ### ---------------------- ###


    begin solver
     ifdef(kinematic)

     begin loadstep predictor
       type = scale˙factor
       scale factor = 1.0 0.0
     end
```

```
      begin control contact
        target relative residual = 1.0e-4
        target relative contact residual = 1.0e-4
        maximum iterations = 100
        minimum iterations = 2
      end
      begin cg
        target relative residual = 1.0e-6
        acceptable relative residual = 1.0e10
        maximum iterations = 1000
        minimum iterations = 10
        iteration print = 10
        preconditioner = elastic
      end cg

      else

      begin control contact
        target relative residual = 1.0e-3
        target relative contact residual = 1.0e-3
        maximum iterations = 100
      end
      begin cg
        target relative residual = 1.0e-5
        acceptable relative residual = 1.0e10
        maximum iterations = 100
        iteration print = 10
      end cg

      endif

    end solver



#     begin solver
#       level 1 predictor = none
#       begin loadstep predictor
#         type = scale˙factor
#         scale factor = 0.0
#       end
#       begin control contact
#         target relative residual =  1e-4
#         maximum iterations      = 500
#         minimum iterations      = 3
```

```
#       end control contact
#       begin cg
#          target relative residual = 1e-5
#          acceptable relative residual = 1000
#          minimum residual improvement = 0.99
#          maximum iterations = 250
#          iteration print = 1
#          begin full tangent preconditioner
#             #number of smoothing iterations  = 10
#             #iteration update = 100
#             #small number of iterations = 20
#          end full tangent preconditioner
#       end cg
#    end solver

#    begin adaptive time stepping
#       cutback factor = 0.5
#       growth factor  =  1.2   #1.1
#       maximum failure cutbacks = 8
#       maximum multiplier =
#       minimum multiplier = 1.e-9
#       target iterations = 50
#       iteration window = 10
#    end
#
```

  END ADAGIO region rPULL

  END ADAGIO procedure PULL

  BEGIN FETI EQUATION SOLVER FETI
     #use defaults
  END FETI EQUATION SOLVER FETI

  END

DISTRIBUTION:

1 MS 0346   Organization 1526, 1526 (electronic)
1 MS 0557   C. Dennis Croessmann, 1520 (electronic)
1 MS 0840   Organization 1554, 1554 (electronic)
1 MS 9042   Organization 8259, 8259 (electronic)
1 MS 0899   Technical Library, 9536 (electronic copy)