SAND2024-01732

SANDIA REPORT

Printed February 15, 2024

Sandia National Laboratories

Sierra/SD – Verification Test Manual – 5.18

Sierra Structural Dynamics Development Team

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

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & 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@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

Verification and validation (V&V) of scientific computing programs are important at *Sandia National Labs* due to the expanding role of computational simulation in managing the United States nuclear stockpile. The complexities of structural response calculations used to analyze physical problems, the varieties of codes applied to the calculations, and the importance of accurate predictions when assessing field conditions demand confidence in the consistency and accuracy of computer codes. Confidence in the accuracy of the predictions arising from computer simulations must ultimately be gained through verification and validation.

The Sierra salinas structural dynamics analysis code, **Sierra/SD**, is used at the DOE Laboratories, and in several DOD projects. The roles of **Sierra/SD** in the qualification of weapon systems and components for normal and hostile environments throughout the Stockpile-to-Target Sequence include to,

- Redesign weapon components.
- Certify weapon components and systems for target environments such as hypersonic vehicles.
- Certify that components will survive the thermal mechanical shock loads associated with hostile environments.
- Evaluate current stockpile issues, including issues associated with uncertainty quantification.
- Address many other problems that are encountered in stockpile management.

The **Sierra/SD** verification plan is described, and an evolving set of key verification tests are described in detail. The verification tests ensure the correctness of the mathematics and numerical algorithms associated with functionality describing engineering phenomena. Development is in accordance with a set of tailored Software Quality Engineering (SQE) practices.[54] SQE practices guide the overall verification and validation effort.

This page left blank

CONTENTS

LIST OF FIGURES

This page intentionally left blank.

LIST OF TABLES

Acknowledgments

The **Sierra/SD** software package is the collective effort of many individuals and teams. A core Sandia National Laboratories based **Sierra/SD** development team is responsible for maintenance of documentation, testing, and support of code capabilities. This team includes Dagny Beale, Gregory Bunting, David Day, Clark Dohrmann, Payton Lindsay, Justin Pepe, and Julia Plews.

The **Sierra/SD** team also works closely with the Sierra Inverse and Plato teams to jointly enhance and maintain several capabilities. This includes contributions from Ryan Alberdi, Wilkins Aquino, Brett Clark, Sean Hardesty, Clay Sanders, Chandler Smith, Adam Sokolow, Benjamin Treweek, Timothy Walsh, and Ray Wildman.

The **Sierra/SD** team works closely with other Sierra teams on core libraries and shared tools. This includes the DevOps, Sierra Toolkit, Solid Mechanics, Fluid Thermal Teams. Additionally, analysts regularly provide code capabilities as well as help review and verify code capabilities, testing, and documentation. Other individuals not already mentioned directly contributing to the **Sierra/SD** documentation, testing, and code base during the last year include Simon Bignold, Leah Brinkman, Samuel Browne, Michael Buche, Mark Chen, Nathan Crane, Jared Crean, David Glaze, Mark Hamilton, Sidharth Joshi, Andrew Kimler, Dong Lee, Mario LoPrinzi, Kevin Manktelow, Matthew Mosby, Devin OConnor, Tolu Okusanya, Heather Pacella, Krishen Parmar, Kendall Pierson, Tim Shelton, Greg Sjaardema, Matt Staten, Johnathan Vo, Tyler Voskuilen, Alan Williams, and Riley Wilson.

Historically dozens of other Sandia staff, students, and external collaborators have also contributed to the **Sierra/SD** product and its documentation.

Many other individuals groups have contributed either directly or indirectly to the success of the **Sierra/SD** product. These include but are not limited to;

- Garth Reese implemented the original **Sierra/SD** code base. He served as principal investigator and product owner for **Sierra/SD** for over twenty years. His efforts and contributions led to much of the current success of **Sierra/SD**.
- The ASC program at the DOE which funded the initial **Sierra/SD** (Salinas) development as well as the ASC program which still provides the bulk of ongoing development support.
- Line managers at Sandia Labs who supported this effort. Special recognition is extended to David Martinez who helped establish the effort.
- Charbel Farhat and the University of Colorado at Boulder. They have provided incredible support in the area of finite elements, and especially in development of linear solvers.
- Carlos Felippa of U. Colorado at Boulder. His consultation has been invaluable, and includes the summer of 2001 where he visited at Sandia and developed the Hexshell element for us.
- Danny Sorensen, Rich Lehoucq and other developers of ARPACK, which is used for eigenvalue problems.

- Esmond Ng who wrote *sparspak* for us. This sparse solver package is responsible for much of the performance in **Sierra/SD** linear solvers.
- The *metis* team at the University of Minnesota. *Metis* is an important part of the graph partitioning schemes used by several of our linear solvers. These are copyright 1997 from the University of Minnesota.
- Padma Raghaven for development of a parallel direct solver that is a part of the linear solvers.
- The developers of the SuperLU Dist parallel sparse direct linear solver. It is used through GDSW for a variety of problems.
- Leszek Demkowicz at the University of Texas at Austin who provided the HP3D[24] library and has worked with the **Sierra/SD** team on several initiatives. The HP3D library is used to calculate shape functions for higher order elements.

This work was supported by the Laboratory Directed Research and Development (LDRD) program.

1. INTRODUCTION

This document covers the verification of **Sierra/SD**.

Sierra/SD provides a massively parallel implementation of structural dynamics finite element analysis. This capability is required for high fidelity, validated models used in modal, vibration, static and shock analysis of weapons systems. General capabilities for modal, statics and transient dynamics are provided.

Sierra/SD is similar to commercial codes like Nastran or Abaqus. It has some nonlinear capability, but excels in linear computation. It is different than the above commercial codes in that it is designed to operate efficiently in a massively parallel environment.

This page intentionally left blank.

2. SOLUTION CASES

2.1. Craig Bampton Reduction

The goal of the test is to verify both generation of a Craig-Bampton (CB) reduction and associated matrix sensitivities as well as the Taylor series expansion of the resulting matrices to generate a point evaluation of a parameter.

In the two hexahedron element structure in Figure 2-1. the yellow right-hand side element is reduced to a superelement. The red left-hand side element is the "residual structure." It is clamped on the far left surface, and will use the superelement.

After the CB reduction, sensitivity matrices dKr/dp and dMr/dp are generated. The reduction is performed in two ways: by constant vector, and by finite difference approaches. The last simulation step is a system analysis (of the two element structure). Finally dKr/dp and dMr/dp are used in a Taylor series expansion for comparison purposes.

For this analysis, we use the material density as the sensitivity parameter. The model is selected so there are no repeated frequencies.

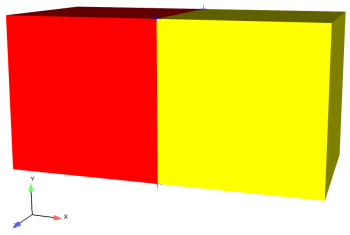


Figure 2-1. – One hexahedron superelement model.

In the constant vector sensitivity analysis uses the reduced order matrices, Identical relations exist for the mass matrix.

$$\tilde{k}_o = T_o^T K(p_o) T_o, \quad \tilde{k}_1 = T_o^T K(p_o + \Delta p) T_o, quad \frac{d\tilde{k}}{dp} \sim \frac{\tilde{k}_1 - \tilde{k}_o}{\Delta p}$$

In our example, the density of a single element is the only sensitivity parameter. The density has no impact on the stiffness matrix, so $\tilde{k}_1 = \tilde{k}_o$, and $d\tilde{k}/dp = 0$. There is a change in the mass matrix, which will affect the system eigen frequencies.

T_o	transformation matrix evaluated at p_o
p_o	sensitivity parameter value
Δp	change in p_o
$\tilde{k}$	reduced stiffness matrix
$K()$	unreduced stiffness matrix

A **finite difference** sensitivity analysis uses the transformation matrix T_1 evaluated at $p_o + \Delta p$,

$$\tilde{k}_o = T_o^T K(p_o) T_o, \quad \hat{k}_1 = T_1^T K(p_o + \Delta p) T_1, \quad \frac{d\tilde{k}}{dp} \sim \frac{\hat{k}_1 - \tilde{k}_o}{\Delta p}$$

Because T_1 depends on the density, the reduced stiffness matrix is affected by the transformation. Interestingly enough, the reduced mass matrix is impacted less because of normalization of the fixed interface nodes, which counter the effect of increased mass. The 1,2 and 2,2 sections of the matrix do change.

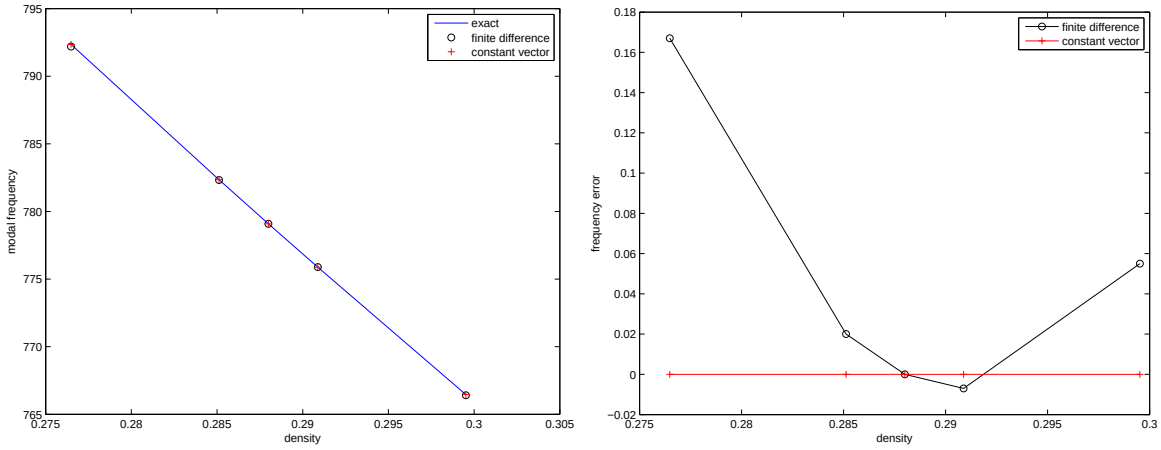


Figure 2-2. – Frequency variation with Density (left) and errors (right).

The right plot in Figure 2-2 of system level solutions as a function of density actually shows three curves. The exact solution shows results obtained by rebuilding the superelement using the parameter, and without sensitivities. The other two curves evaluate dk/dp at the nominal value, and estimate the superelement contribution using a Taylor series expansion. Results are shown for mode 3. The left plot shows the errors.

These analyses compare results for application of sensitivity matrices to superelement analysis. In this extremely simple example, the constant vector method is exact, while finite difference methods introduce a slight error. That is not a general case. For input deck see Appendix 9.1.

2.2. Beam-Beam with Craig-Bampton Reduction

This model exercises CBR reductions on a beam. The full model consists of 200 beam elements, each of length 0.01, for a total length of 2 units. The beam is free floating in the X direction, but constrained in all other directions. It is driven by a simple force on the left ($x = 0$) end. The load is a saw tooth force with a period and duration of 1.5 ms. The system is integrated with a fixed time step of 0.1 ms.

An "equivalent" model is generated by separating the model into two equal sections of 100 elements each. The right-hand side segment is converted into a superelement, and then attached to the left-hand structure. The superelement includes the single fixed dof on the left end, and 90 internal generalized dofs representing most of the modes of the system. The loading and integration are identical to the full structure.

Figure 2-3 compares the X component of displacement on node 101 of both models. Node 101 is located at the junction of the superelement. Clearly the superelement and residual structure represent the solution very well. Figure 2-4 shows the difference of the solutions.

For comparison, Figure 2-5 compares results with a CBR model that includes no generalized dofs. As anticipated, the results are not nearly as good.

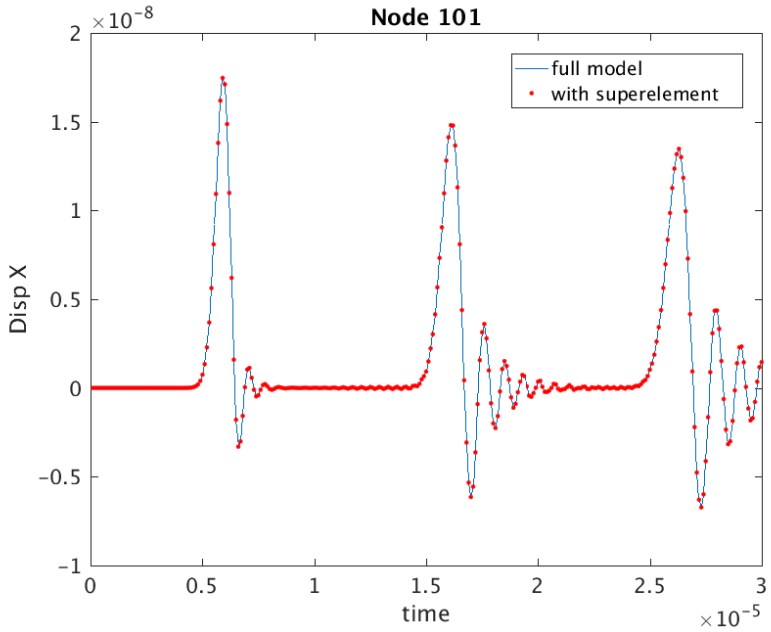


Figure 2-3. – Comparison of Full Model with CBR Reduction.

For input deck see Appendix 9.2.

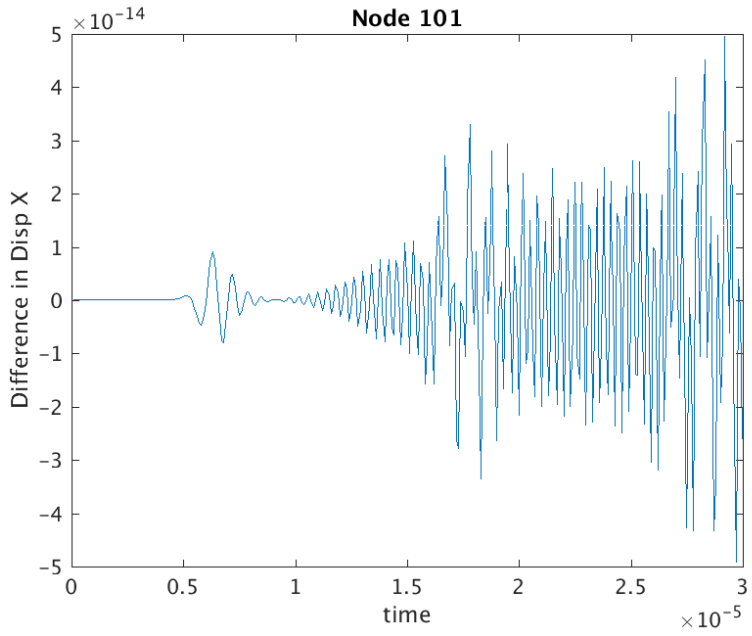


Figure 2-4. – CBR Reduction Error.

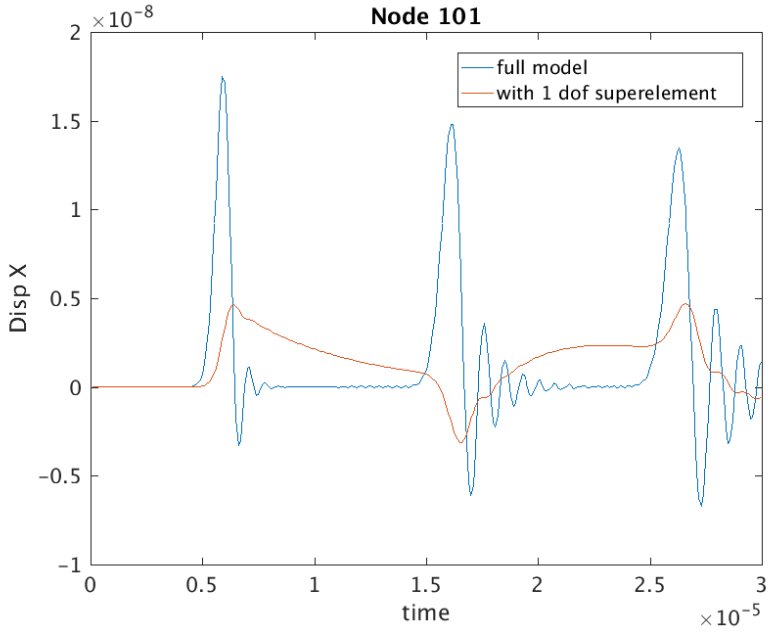


Figure 2-5. – Comparison of Full Model with Guyan Reduction. Without the generalized DOFs, the comparison is poor.

2.3. Craig-Bampton Output Transfer Matrix (OTM)

CBR including the Output Transfer Matrix (OTM) are discussed in[43] and.[42] The following steps are to be used for verification. The model used is the multi-element/olio_cbr_test.

1. ensure eigenvalues are consistent between models (reduced versus full)

   This portion of the test that is evaluated as part of the automated test.

2. check OTM for displacement in serial.

   a) Is data consistent with ϕ and ψ?

   This is checked in the debugger.

   b) does the product make sense,

$$x_k = [OTM][x_1]$$

$$\bar{x}_k = K^{-1}x_1$$

   and,

$$x_k \approx \bar{x}_k$$

This is done as follows.

   a) The model is clamped away from the interface to eliminate the confusion caused by redundant modes and zero energy modes. The system response is computed for mode 1 (a flexible mode). This is done by pulling in Kssr and Mssr and computing the eigenvalues, **E**, and eigenvectors, **V**.

   b) The reduced model is also computed for mode 1. We do this by computing the eigenvalues and eigenvectors of **Kr** and **Mr**, `[vr, er]=eig(Kr,Mr);`

   We ensure that the eigenvalues are approximately the same. See figure 2-6.

   c) The first eigenvectors is expanded to the system from both systems. The reduced eigenvectors contain both a physical coordinate and a modal coordinate component. MATLAB code to do this expansion is shown in Figure 2-8. A comparison of the two vectors is shown in Figure 2-7. Note that there is a scale factor difference of -1 in the two vectors. This is acceptable as eigenvector scaling is arbitrary to that factor.

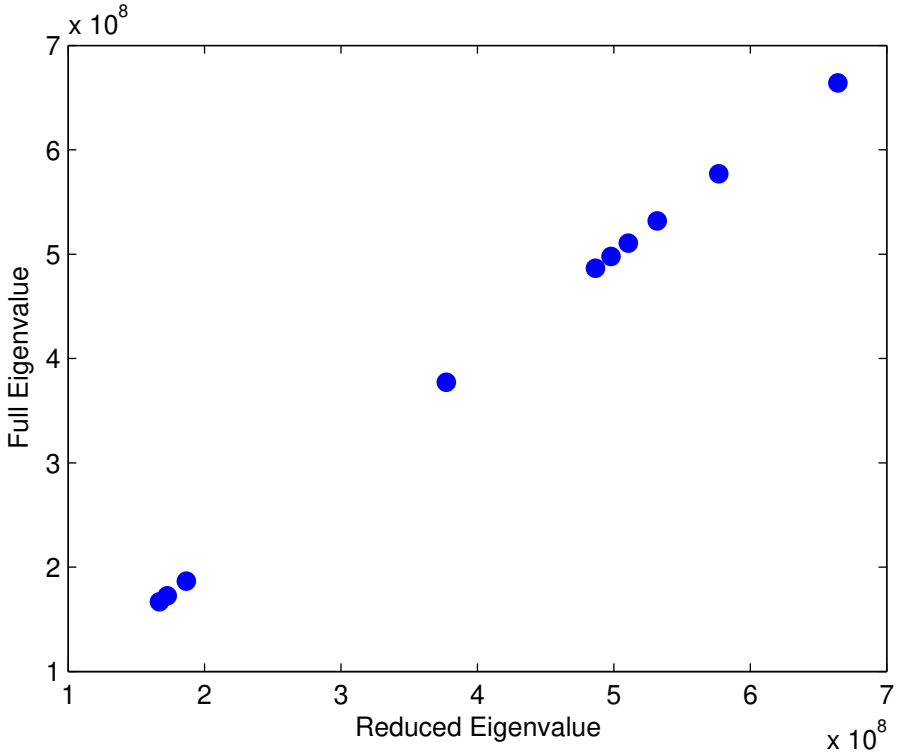


Figure 2-6. – Comparison of reduced and full eigenvalues.

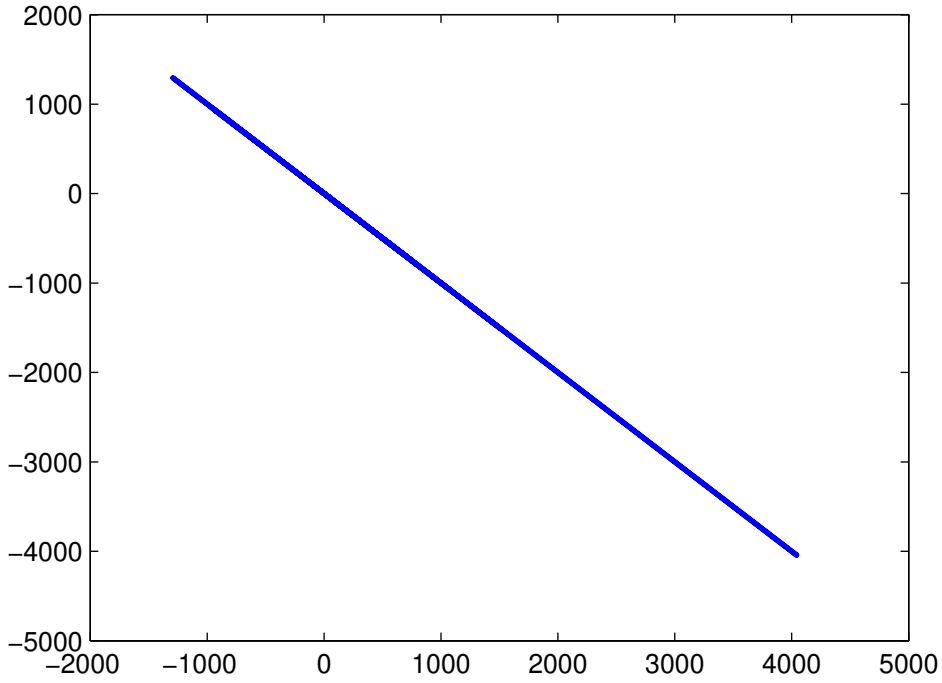


Figure 2-7. – Comparison of reduced and full eigenvectors.

```
function [dispgr,nodes]=expandRmodel( cbmap, OTM, OutMap, vr )
% expands a vector in the reduced, Craig Bampton space into the
% full physical space.
% cbmap - map to interface DOFs. Output into cbr.m
% OTM - Output transfer matrix. also in cbr.m
% OutMap - map to interior (and interface) nodes in output.
% vr - the reduced space vector.
%         vr(1:numeig) is the amplitude of the fixed interface modes
%         vr(numeig:end) is the amplitude of the constraint modes (physical
%                         degrees of freedom).
% results are output sorted by node number. 6 DOFs per node are output.

nodes=[cbmap(:,1)' OutMap];
nodes=unique(nodes);
nout=size(nodes,2);
nr=max(size(vr));
nc=size(cbmap,1);
nmodes=nr-nc;

dispgr=zeros(nout*6,1);
ur=OTM*vr;   % compute vector on OTM space, ur

% store components from OTM space.
for i=1:size(OutMap,2)
  n=OutMap(i);
  k=find(nodes==n);
  for cid=1:6
    k2=(k-1)*6+cid;
    k1=(i-1)*6+cid;
    dispgr(k2)=ur(k1);
  end
end

% transfer interface DOFs directly
for i=1:nc
  n=cbmap(i,1);
  cid=cbmap(i,2);
  k=find(nodes==n);
  k2=(k-1)*6+cid;
  dispgr(k2)=vr(i+nmodes);
end
```

Figure 2-8. – MATLAB code to convert from reduced space.

2.4. Residual Vectors

As a small problem to test the residual vector computations in **Sierra/SD**, two beams are connected to each other to simulate a longer beam. To keep the overall number of DOFs as small as possible, the finite element mesh of the beam cross-section is limited to two elements in each direction. This is the bare minimum required to model bending vibrations. The physical parameters for the beams are listed in Table 2-1.

Table 2-1. – Physical parameters for the beams.

Parameter	Beam 1	Beam 2
Density	7860 Kg/m3	7860 Kg/m3
Poisson Rs Ratio	0.29	0.29
Modulus of Elasticity	200 GPa	200 GPa
Width (Y-direction)	0.01 m	0.01 m
Height (Z-direction)	0.005 m	0.005 m
Length	0.25 m	0.225 m

When the two beams are combined the overall length is 0.475 m. Analytical solutions for the resonance frequencies are available in the book by Weaver, Timoshenko and Young[52] for a variety of boundary conditions.

The analysis strategy is standard. Component modes synthesis (CMS) has been in use for a long time and many variations on the general analysis procedure are available. The basic idea of all CMS computations is to divide the structure into Scomponents T whose displacements are represented as a summation of Snormal modes T with the mode sets truncated above an upper limiting frequency. This representation is adequate to accurately compute displacements, but not nodal forces or stresses (which represent spatial derivatives of the displacement field). Thus, some method must be used in a CMS analysis to account for truncated modes, especially at locations where the forces must be computed accurately. One simple method is to add Sresidual T or Smodal truncation augmentation T vectors to the analysis for specified nodal locations and DOFs. An excellent derivation of modal truncation augmentation vectors is given in.[16] The vectors are orthogonal to the normal modes, have same normalization, and may be added to the basis.

In the most general form of CMS analysis, interfaces are defined between each of the components and Sinterface modes T are used to represent the connections themselves. Here, a simplified form of CMS is used where the connections between components is at discrete nodal locations instead of interfaces. This eliminates the need to compute Sinterface modes T. It applies to problems (and frequency ranges) where the interfaces can be considered to vibrate as rigid bodies. For the current example of two connected beams, rigid elements are used to make all the nodes at the ends of the beams dependent on nodes at the beam center line. Figure 2-9 illustrates the implementation of one of the rigid elements in NASTRAN.

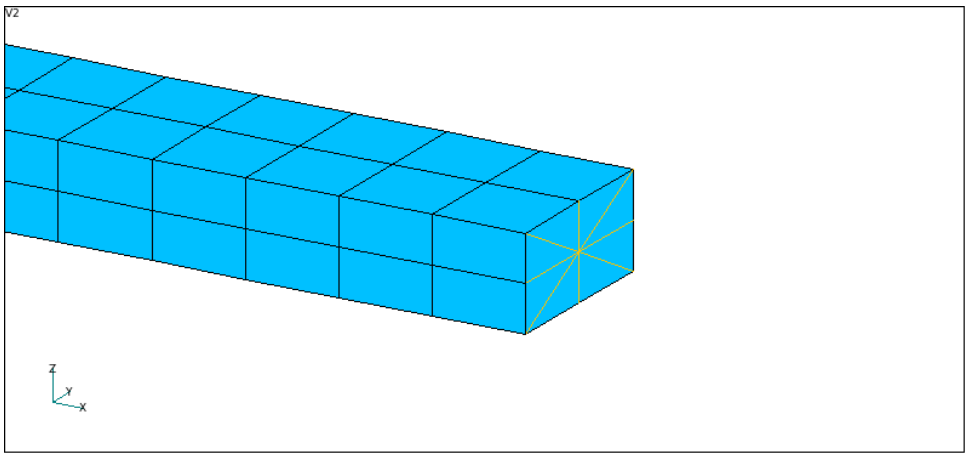


Figure 2-9. – Illustration of a rigid element making all the nodes at the end of the beam dependent on a single node.

This is a reasonable assumption for the beams under consideration because modes with significant variations across the cross-section occur above the frequency range of interest.

As mentioned previously, the user must specify the nodes for the residual vectors calculations. The connection forces between the components must be computed accurately in a CMS solution, and thus residual vectors are included in the basis set for all 6 DOFs at any location where two components are connected to each other. It is often useful to also include residual vectors for nodal locations where boundary conditions are to be applied instead of explicitly including the boundary conditions as nodal constraints in the finite element analysis. The the normal modes and residual vectors are extracted only once, and a variety of boundary conditions can be applied subsequently. Since forces also have to be computed accurately at the locations where boundary conditions are to be applied, residual vectors are also included for all the DOFs at these nodes. For the present case, one end of each beam connects to the other beam and the other end may possibly be used to apply boundary conditions. Residual vectors are not extracted for all the nodes at the ends of the beams. Instead rigid elements are used to make all the nodes dependent on a single node at the beam centerline. Ultimately, this means that residual vectors are extracted for nodes at both ends of each beam, thus adding 12 residual vectors to the basis set for each beam.

The computations for the single beam were performed in a variety of ways and validated in NASTRAN first before proceeding with the component modes synthesis (CMS) analysis. The goal is to allow 6 DOF for each beam at the connection location and at the ends, RBar elements are used at the ends of the beams to force all the nodes to move together as rigid entities. This representation does not allow the cross-section at the beam ends to deform. It is first compared to a contiguous model without RBars to verify that it does not significantly change the resonance frequencies for the bending modes. Table 2-2 lists the analytical solution for the resonance frequencies assuming free boundary conditions along with the two NASTRAN computations.

The integer N in the table lists the number of nodal lines along the beam's length. The table does

Table 2-2. – Analytical solution for the resonance frequencies of a free-free beam along with solutions from NASTRAN.

N	Primary Direction	Analytical	Contiguous	RBar at Connection
2	Y	114.9 Hz	114.8 Hz	115.0 Hz
2	Z	229.8 Hz	229.3 Hz	229.6 Hz
3	Y	316.7 Hz	316.2 Hz	316.3 Hz
4	Y	621.0 Hz	619.3 Hz	620.0 Hz
3	Z	633.5 Hz	630.0 Hz	630.0 Hz
5	Y	1026.4 Hz	1022.4 HZ	1022.6 Hz
4	Z	1242.0 Hz	1229.5 Hz	1230.7 Hz
6	Y	1533.4 Hz	1525.0 Hz	1526.5 Hz

not include N = 0 and N = 1 modes because they represent rigid body vibrations (and are at 0 Hz). The beam's width was chosen to be twice its height, and thus the resonance frequencies in the Z-direction are double those for the Y-direction. The results show that the mesh is refined enough to give accurate results, although it is not clear why the resonance frequencies from NASTRAN are lower than those for the analytical solution.

The next step is to perform the calculations as a CMS analysis with the resonance frequencies, mode shapes and residual vectors computed separately for each beam. For both beams in both CMS analyses, 10 normal modes are retained and residual vectors are included for all 6 DOFs for a single node at both ends of the beams. For reference purposes, Table 2-3 lists the resonance frequencies for both the normal modes (excluding rigid body modes) and residual vectors for the two shorter beams.

For the CMS analyses, a separate computer program is used to combine the mode sets and apply the connections between the components and the boundary conditions. The calculations are performed in "modal space" similar to that discussed in the NASTRAN Basic Dynamics User's Guide.[8] The connections and boundary conditions are applied with user-specified stiffnesses between two nodes or between a single node and ground. Specifying large stiffnesses (1x1012 N/m for the current analysis) has the effect of rigidly constraining two nodes to each other or constraining specific DOFs to zero displacement at a single node.

Once the CMS analysis is set-up, it is possible to rapidly perform the computations for the beam with a variety of specified boundary conditions. The NASTRAN solution with the two beams connected to each other with a rigid RBar element is used as the reference since the CMS analysis should produce identical results. Table 2-4 Table 2-5 Table 2-6 Table 2-7 list the beam resonance frequencies for various boundary conditions using the NASTRAN solution with an RBar connection and for the two CMS analyses.

The results in the tables show good agreement between the NASTRAN model and the CMS analyses that include residual vectors. Without residual vectors, the resonance frequencies are considerably too high. While the CMS analyses require some extra effort to set-up, it is possible to perform all the computations with a single model by changing the stiffnesses applied at the ends

Table 2-3. – Resonance frequencies for the normal modes and residual vectors in NASTRAN and **Sierra/SD**.

Type	Beam 1 NASTRAN	Beam 1 **Sierra/SD**	Beam 2 NASTRAN	Beam 2 **Sierra/SD**
Normal Mode	414.5 Hz	414.5 Hz	511.7 Hz	511.7 Hz
	825.7 Hz	825.7 Hz	1018.3 Hz	1018.3 Hz
	1142.1 Hz	1142.1 Hz	1409.6 Hz	1409.6 Hz
	2237.7 Hz	2237.9 Hz	2761.4 Hz	2761.7 Hz
Residual Vector	2335.3 Hz	2335.2 Hz	2877.5 Hz	2877.4 Hz
	4030.5 Hz	4030.8 Hz	4976.4 Hz	4976.9 Hz
	4684.9 Hz	4684.7 Hz	5767.6 Hz	5767.3 Hz
	5521.6 Hz	5520.6 Hz	6133.1 Hz	6131.8 Hz
	6181.5 Hz	6182.3 Hz	7634.8 Hz	7636.1 Hz
	11174.2 Hz	11164.8 Hz	12422.1 Hz	12410.5 Hz
	12270.5 Hz	12265.1 Hz	13622.0 Hz	13615.9 Hz
	16403.7 Hz	16399.7 Hz	20131.7 Hz	20126.7 Hz
	22639.3 Hz	22627.8 Hz	27801.1 Hz	27789.8 Hz
	25214.8 Hz	25151.2 Hz	28060.7 Hz	27981.9 Hz
	28419.4 Hz	28412.3 Hz	34774.4 Hz	34766.1 Hz
	32990.6 Hz	32980.5 Hz	40458.6 Hz	40453.8 Hz

of the beams. The NASTRAN computations for the model required a separate mode extraction analysis for each boundary condition.

Table 2-4. – Comparison of the NASTRAN solution with an RBar connecting the beams to the CMS solutions using NASTRAN and **Sierra/SD** for free-free boundary conditions.

N	Primary Direction	RBar at Connection	CMS, NASTRAN	CMS, **Sierra/SD**	CMS, NASTRAN w/o Residual Vectors
2	Z	115.0 Hz	115.1 Hz	115.1 Hz	132.4 Hz
2	Y	229.6 Hz	229.8 Hz	229.8 Hz	319.3 Hz
3	Z	316.3 Hz	316.7 Hz	316.7 Hz	319.2 Hz
4	Z	620.0 Hz	621.3 Hz	621.4 Hz	706.1 Hz
3	Y	630.0 Hz	631.3 Hz	631.3 Hz	654.6 Hz
5	Z	1022.6 Hz	1025.9 Hz	1026.0 Hz	1053.9 Hz
4	Y	1230.7 Hz	1235.5 Hz	1235.6 Hz	> 2000 Hz
6	Z	1526.5 Hz	1533.7 Hz	1533.9 Hz	1769.0 Hz

Table 2-5. – Comparison of the NASTRAN solution with an RBar connecting the beams to the CMS solutions using NASTRAN and **Sierra/SD** for clamped-clamped boundary conditions.

N	Primary Direction	RBar at Connection	CMS, NASTRAN	CMS, **Sierra/SD**	CMS, NASTRAN w/o Residual Vectors
2	Z	115.2 Hz	115.3 Hz	115.3 Hz	167.3 Hz
2	Y	229.9 Hz	230.0 Hz	230.0 Hz	> 2000 Hz
3	Z	317.2 Hz	317.4 Hz	317.4 Hz	411.3 Hz
4	Z	622.0 Hz	622.7 Hz	622.9 Hz	877.8 Hz
3	Y	631.2 Hz	631.8 Hz	631.8 Hz	> 2000 Hz
5	Z	1026.1 Hz	1028.2 Hz	1028.4 Hz	1346.5 Hz
4	Y	1232.8 Hz	1235.4 Hz	1235.6 Hz	> 2000 Hz
6	Z	1532.0 Hz	1537.0 Hz	1537.4 Hz	> 2000 Hz

Table 2-6. – Comparison of the NASTRAN solution with an RBar connecting the beams to the CMS solutions using NASTRAN and **Sierra/SD** for simply supported boundary conditions.

N	Primary Direction	RBar at Connection	CMS, NASTRAN	CMS, **Sierra/SD**	CMS, NASTRAN w/o Residual Vectors
2	Z	50.7 Hz	50.7 Hz	50.8 Hz	56.5 Hz
2	Y	101.4 Hz	101.4 Hz	101.4 Hz	126.8 Hz
3	Z	202.6 Hz	202.7 Hz	202.8 Hz	203.9 Hz
3	Y	404.4 Hz	404.7 Hz	404.7 Hz	412.8 Hz
4	Z	456.2 Hz	456.7 Hz	456.7 Hz	527 6 Hz
5	Z	809.5 Hz	811.0 Hz	811.1 Hz	839.5 Hz
4	Y	907.7 Hz	909.4 Hz	909.5 Hz	> 2000 Hz
6	Z	1264.6 Hz	1268.3 Hz	1268.4 Hz	1444.3 Hz

Table 2-7. – Comparison of the NASTRAN solution with an RBar connecting the beams to the CMS solutions using NASTRAN and **Sierra/SD** for clamped-free boundary conditions.

N	Primary Direction	RBar at Connection	CMS, NASTRAN	CMS, **Sierra/SD**	CMS, NASTRAN w/o Residual Vectors
1	Z	18.1 Hz	18.1 Hz	18.1 Hz	20.4 Hz
1	Y	36.1 Hz	36.2 Hz	36.2 Hz	46.1 Hz
2	Z	113.4 Hz	113.4 Hz	113.4 Hz	148.1 Hz
2	Y	226.3 Hz	226.4 Hz	226.4 Hz	458.6 Hz
3	Z	316.9 Hz	317.2 Hz	317.2 Hz	362.1 Hz
4	Z	621.0 Hz	622.0 Hz	622.1 Hz	798.1 Hz
3	Y	630.9 Hz	631.8 Hz	631.8 Hz	> 2000 Hz
5	Z	1024.3 Hz	1027.0 Hz	1027.2 Hz	1172.5 Hz

2.5. Waterline of a ship

A code to code comparison was performed between Sierra-SD and the Navy's finite element code Float. This is a ship model, that utilizes the waterline solution case in Sierra-SD. An image of the model is shown in Figure 2-10. Three key parameters were analyzed between the two codes the

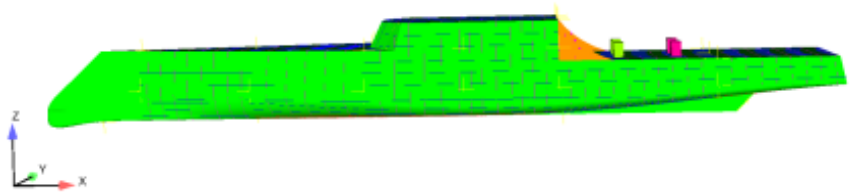


Figure 2-10. – uhwmGeometry.

draft which is the distance from the bottom of the ship to the waterline, the pitch which is the rotation about the y-axis, and the roll which is the rotation about the x-axis. The results can be seen in Table 2-8. For input deck see Appendix 9.3.

Table 2-8. – Sierra-SD solution vs. Float (Navy code).

	Sierra-SD	Float
Draft	187.0580	187.0579
Pitch (about y-axis)	0.0503	0.0497
Roll (about x-axis)	-0.0001	0.0000

2.6. Transient Convergence

A verification test was created for temporal convergence of the transient solution. A vertical load was applied at the end of a cantilever beam, and the vertical displacement at the end of the beam after 4.5 seconds was calculated, and plotted in Figure 2-11.

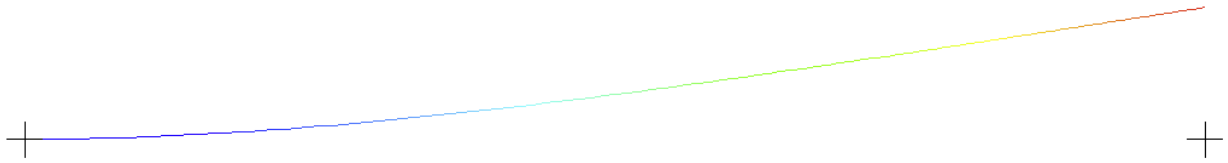


Figure 2-11. – Beam.

Figure 2-12 shows on the left the time history result for the problem, solved at three different time-steps. The Richardson Convergence of the problem, shown on the right in Figure 2-12,

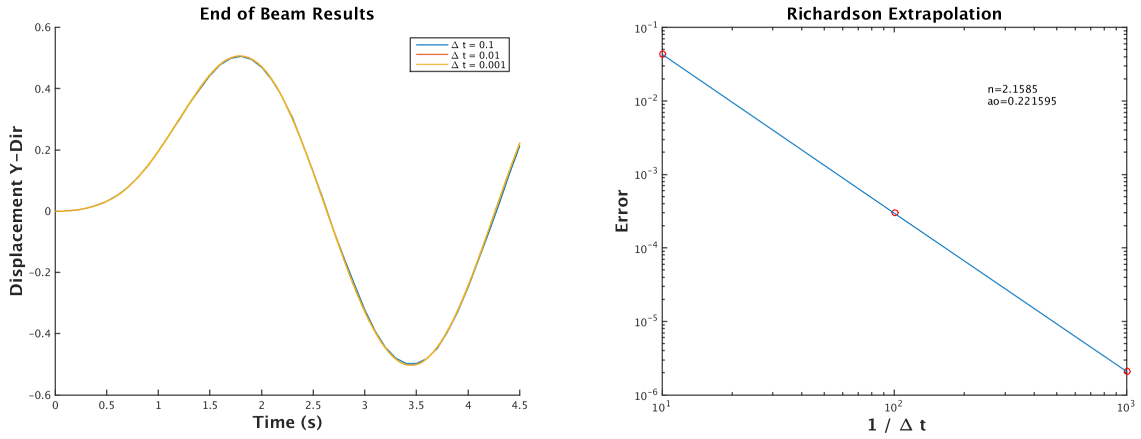


Figure 2-12. – Transient Time History (left) and Richardson Extrapolation (right).

demonstrates second order convergence. For input deck see Appendix 9.4.

2.7. Modal Transient Temporal Convergence

The modal transient temporal convergence The test consists of a 100 element cantilevered beam that is loaded using a triangle pulse function. The modal transient test was run using 3 different time steps, and the results of these tests are compared to the results obtained from the same tests run using the direct transient method.

Figure 2-13 shows the plot of the deformed beam. The loading for the three tests is the same and it consists of a ramp load applied at the free end of the beam. The load has a duration of 2 seconds and a max value of 1 at 1 second.

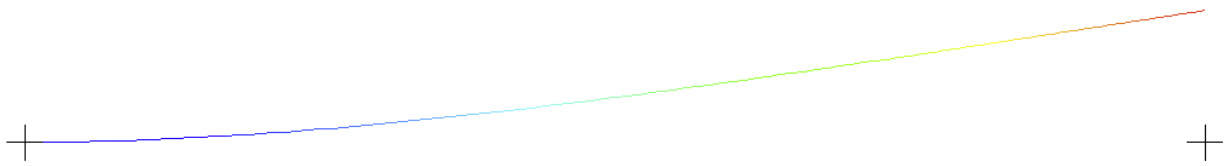


Figure 2-13. – Verification Problem - Beam.

Figure 2-14 shows the time history of the beam end point for the problem for three time steps.

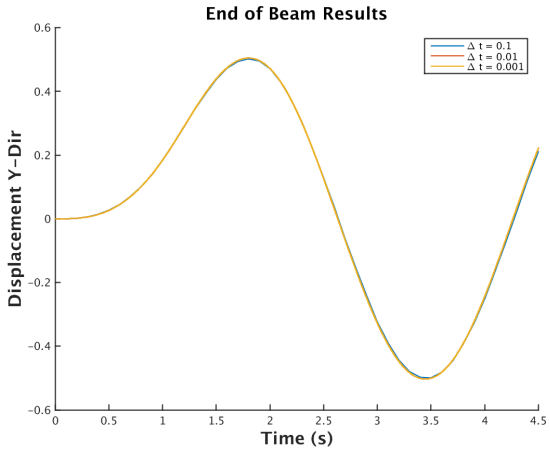


Figure 2-14. – Time History of Modal Transient Verification Problem.

Richardson convergence was used by determining the order of convergence for the modal transient method. Figure 2-15 shows the Richardson Convergence of the problem. Convergence values $n = 2$ implies second order convergence. This result is similar to the Richardson convergence obtained from the direct transient method.

The modal transient tests were run using only 3 modes for verification purposes. Figure 2-16 shows the difference in displacement at the end of the beam between the direct transient method and the modal transient method for $\Delta t = 0.001$.

This difference decreases as the number of modes used in the modal transient method are increased. The three modes retained are sufficient to approximate most of the solution for this low

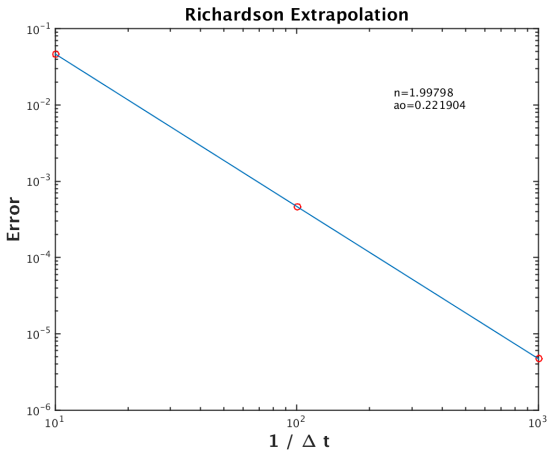


Figure 2-15. – Richardson Extrapolation of Modal Transient Verification Problem.

frequency loading. This corresponds well to the analysis use case where modal transient is used to represent the lower frequency response of complex systems. Temporal convergence depends on adequate modal basis. A similar study with high frequency input could not be expected to converge without a much larger modal basis.

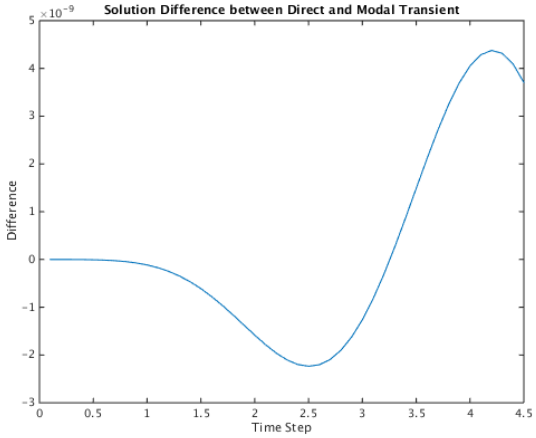


Figure 2-16. – Displacement Difference for Modal and Direct Transient Solutions.

For input deck see Appendix 9.5.

2.8. Transient Restart

Analysts should be able to restart from any of the transient analysis capabilities into any of the others. Of course, there are differences in the solutions with respect to accuracy, and output quantities. For example, the nonlinear transient integrator outputs the number of nonlinear steps as a global output variable. This is unavailable for modal transient. In addition, there are internal variables associated with nonlinear elements and viscoelastic materials which may not be propagated across the restart boundary.

Verification of this use case involves the following steps.

1. Computation of 40 normal modes.
2. Computation and output of 30 ms of time history with the first integrator.
3. Exit **Sierra/SD**, and start a new **Sierra/SD** analysis.
4. Restart read the previous normal modes.
5. Restart read the previous time history data, and computation of the next 10 ms of data.
6. Check of the .rslt to ensure that the time history data was restarted (as opposed to recomputed from scratch).
7. Check the history file for accuracy. Note that the tolerances are loose on this check. Each integrator provides a somewhat different solution (as expected).
8. Visual comparison of the results.

Table 2-9 indicates the tests that have been performed. Nonlinear transient as the first integrator is not currently tested. Figure 2-17 provides the data for the second row of Table 2-9, which includes all cases where the direct transient was the first integrator. Likewise, figure 2-18 shows data for modaltransient as the initial integrator.

Integrator	**NLtransient**	**transient**	**modaltrans**	**Explicit**	**modaltrans**
NLtransient	Untested	Untested	Untested	Untested	NA
transient	TESTED	TESTED	TESTED	TESTED	NA
modaltransient	TESTED	TESTED	TESTED	TESTED	NA
Explicit	TESTED	TESTED	Untested	TESTED	NA
modaltrans	NA	NA	NA	NA	NA

Table 2-9. – Tested restart capabilities for transient integrators in **Sierra/SD**.

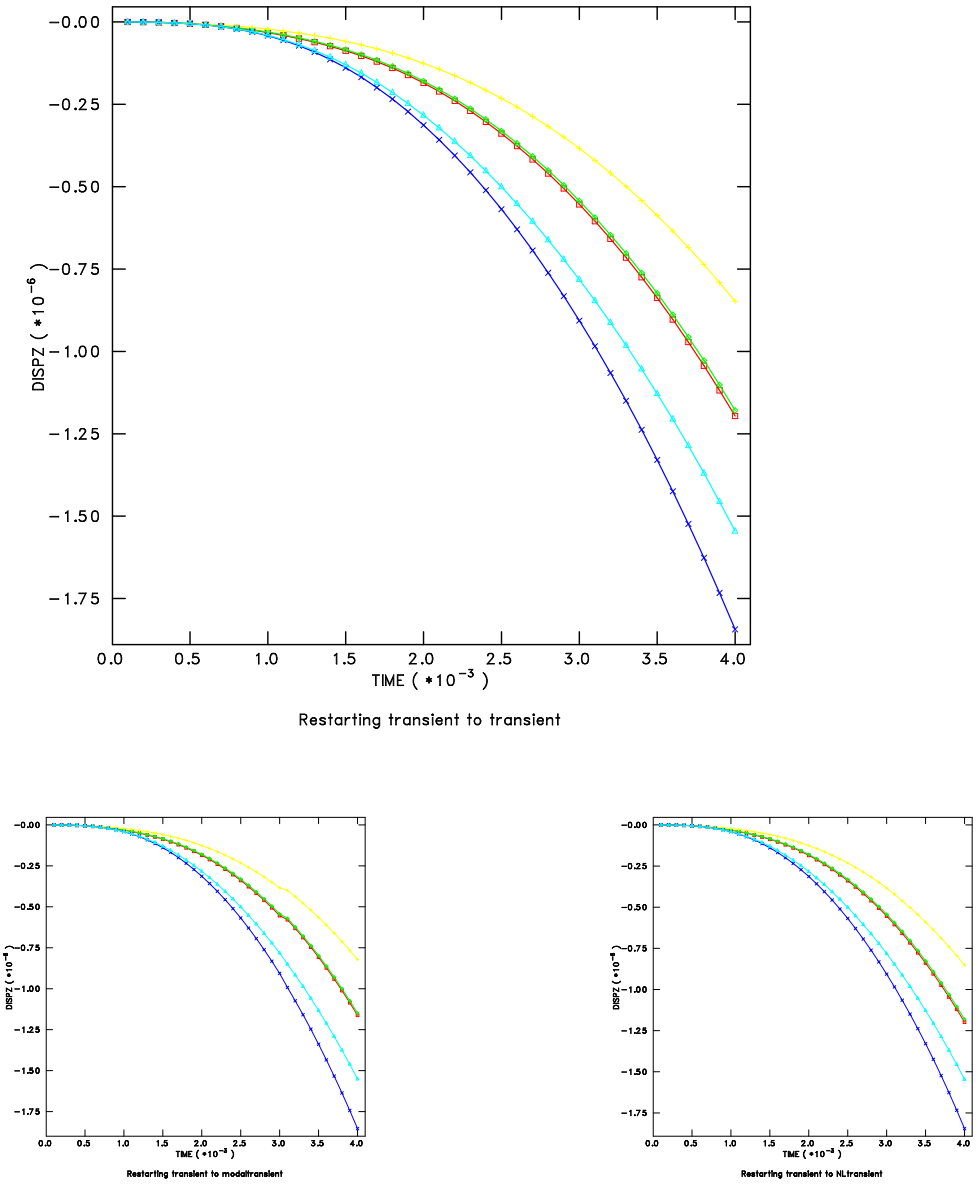


Figure 2-17. – Restart from Direct Transient Analysis. In each case, 30 ms of analysis is completed using a direct transient run, and is followed by a restart.

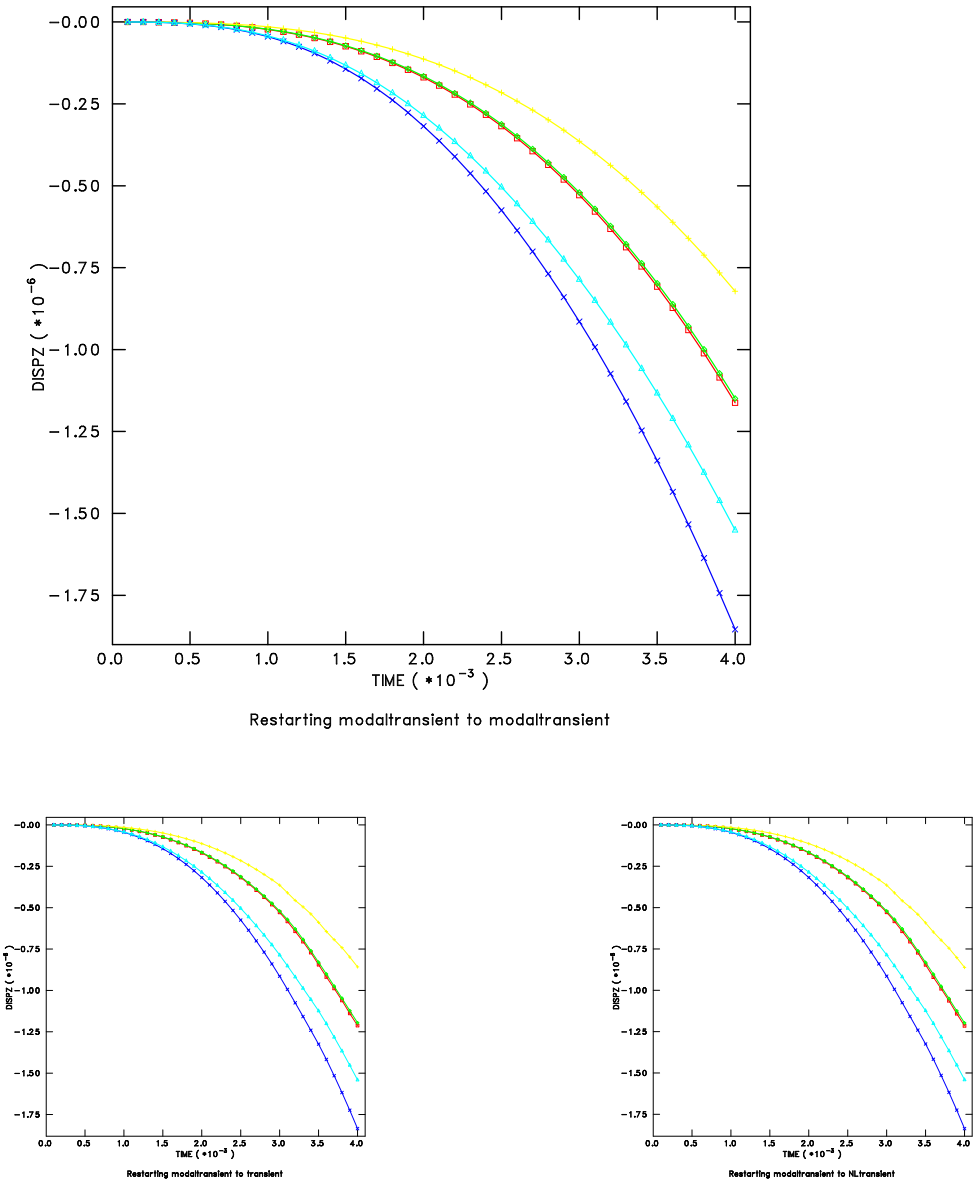


Figure 2-18. – Restart from Modal Transient Analysis. In each case, 30 ms of analysis is completed using a Modal transient run, and is followed by a restart.

2.9. Generalized Alpha Time Integration

Though it is not always done in finite element code verification, it was deemed appropriate to verify that the generalized alpha time integrator[13] was implemented correctly. To isolate that feature, a single degree of freedom simple harmonic oscillator problem was solved. In this problem, the mass and stiffness were each set to unity. The period of free vibration would be 2. A unit load was imposed for a half a period and the resulting free vibration was calculated. The exact solution to this problem is

$$u(t) = 2\cos t$$

The **Sierra/SD** results for time steps $2\pi/200$, $2\pi/400$, $2\pi/800$, and $2\pi/1600$ were computed. The resulting displacements for all four cases are almost identical and are shown in Figure 2-19.

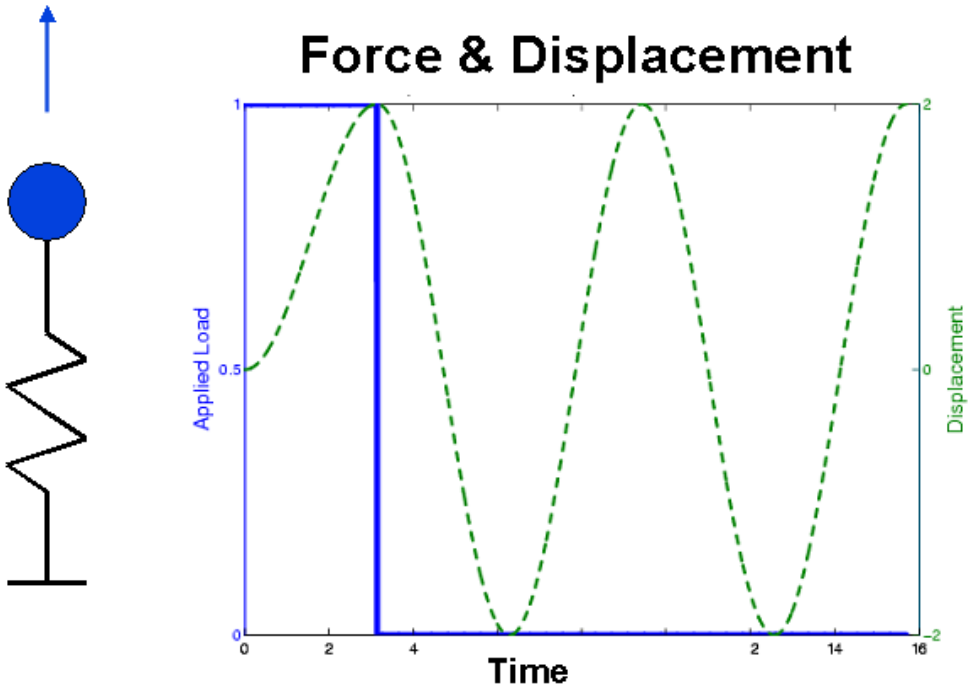


Figure 2-19. – The time integrator is tested against a simple harmonic oscillator. Values of displacement at time 8π are compared and tested for convergence.

Values at time 8π were compared and the resulting convergence plot is shown in Figure 2-20. We see that the convergence rate is almost exactly two – the theoretical value.

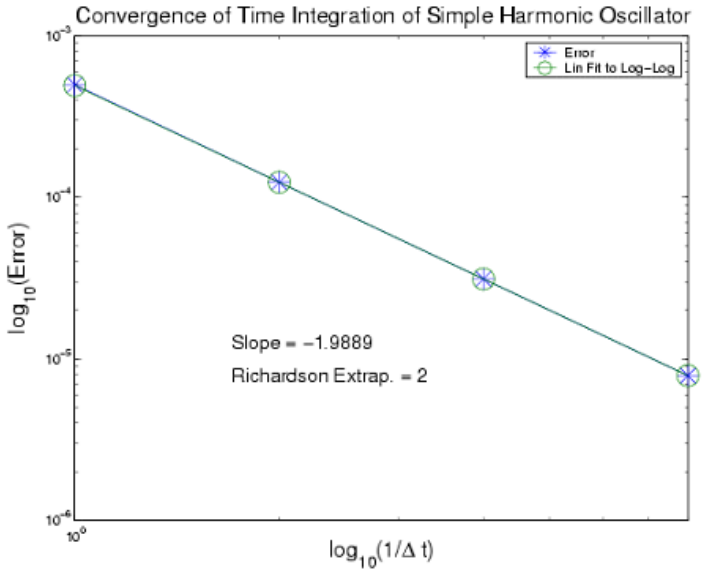


Figure 2-20. – Convergence of Simple Harmonic Oscillator.

2.10. Prescribed acceleration capability

Prescribed accelerations are verified using a cantilever beam model 10 meters in length, with a square cross-section of 1 meter dimension. The beam is subjected to an end-loaded acceleration in the axial direction given by

$$a(t) = cos(\omega t),$$

where $\omega = 2\pi f$, and $f = 16$Hz. The initial conditions, including initial displacement and initial velocity of the beam are set to zero. Given these conditions, we can integrate the acceleration equation twice to obtain the following expression for the displacement at the loaded end

$$D(t) = \frac{1}{(32\pi)^2}(1 - cos(32\pi t)).$$

Figure 2-21 shows a comparison of the analytical solution for displacement against the **Sierra/SD** result. Excellent agreement is observed. We note that this example can be found in the test suite at the following location.

`Salinas_rtest/verification/transient/bar_prescribed.xml`

For example inputs, see Appendix 9.7. The model is shown in Figure 2-36.

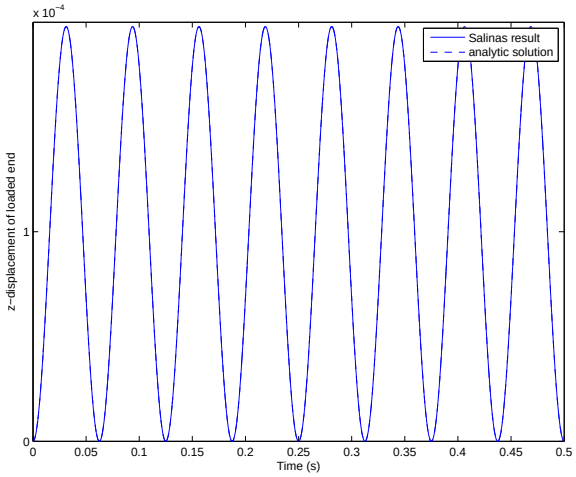


Figure 2-21. – Comparison of **Sierra/SD** result with analytical solution of a beam with end-loaded prescribed acceleration.

2.11. Modal Transient

The modal solution method is a standard Newmark-beta integrator that is applied in parallel on the modal space. It is limited to boundary conditions that are space/time separable, and all outputs must fit on single processor. Verification is applied to four cases.

1. Constant force on a floating body, with limited modal interaction. The behavior is rigid body only, and analytic solutions are trivial.
2. Repeat the above, but eliminate the rigid body motion. A comparison with the standard modal solution provides the verification.
3. We repeat case 2, but add modal damping. Again, the analytic solution is straightforward.
4. A complex loading.

The above examples exercise the primary elements of the software. The model is shown in Figure

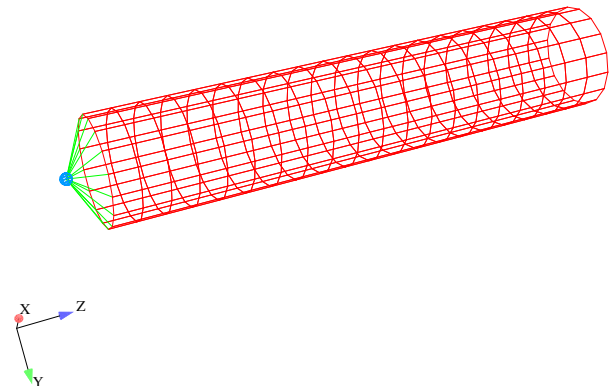


Figure 2-22. – A thin cylinder has beams on its base attached to a large mass, and the load is applied to the mass.

2-22.

2.11.1. Constant Force Applied to Floating Structure

In this example the load activates only a rigid body mode, and the body behaves as a point mass. The analytic solution for a constant force applied to a point mass is,

$$
\begin{aligned}
a(t) &= \frac{F_o}{m} \\
v(t) &= \int_o^t a(t')dt' \\
&= \frac{F_o}{m}t \\
d(t) &= \int_o^t v(t')dt' \\
&= \frac{F_o}{2m}t^2
\end{aligned}
$$

The dimensionless load is set to 10^5 in the input file, and the result file indicates that the total dimensionless mass of the structure is $1001.25 \cdot wtmass$=2.5932375.

Figure 2-23 compares the analytic and numerical solutions for displacement. Figure 2-24 provides similar results for acceleration. While the agreement is excellent, a small discrepancy is observed if differencing the solutions. This occurs because the numerically integrated solution tends to lag the analytic solution by a half step.

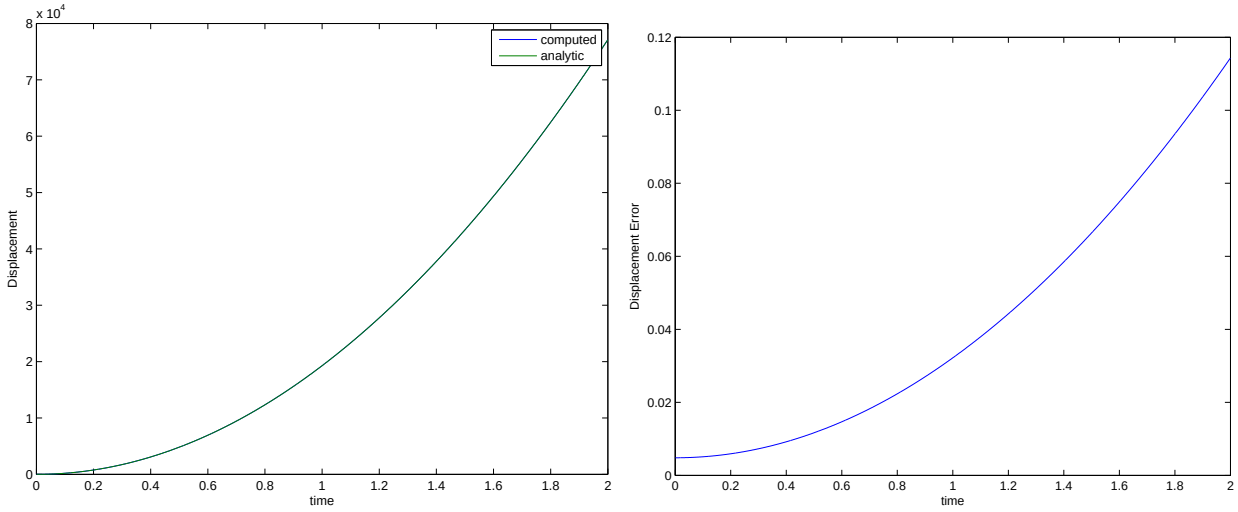


Figure 2-23. – Response of Rigid Body Mode.

This example ensures that the modal force is being computed properly for rigid body modes. As they are identical to elastic modes, that follows as well. It verifies the behavior of the integrator, except that there are contributions from the damping matrix which are not considered.

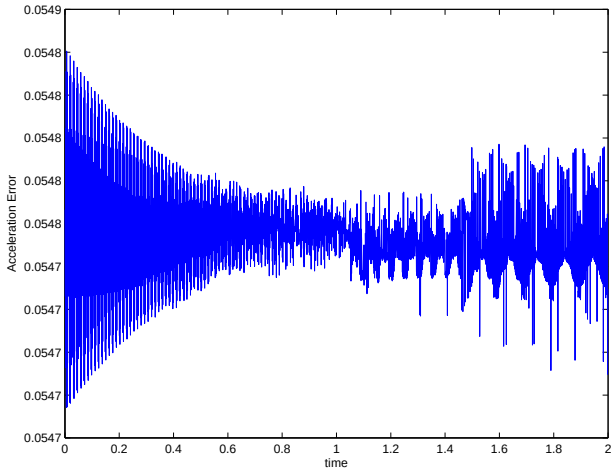


Figure 2-24. – Acceleration Response of Rigid Body Mode. The analytic acceleration is a constant of $F/m \approx 38561.8$. The error in the figure is much smaller than this, and represents the elastic response of higher frequency elastic modes, that are just slightly active in the analysis.

2.11.2. A single Elastic Mode

While the analytic expression for an analytic mode is not quite as complete as for a rigid body response, we may still proceed with verification. We assume that the eigenvalues are computed correctly. We also assume that the modal force, $f_q(t) = \phi^T f(x,t)$, has been verified. The previous example addresses this. Then, the analytic response may be computed.

$$
\begin{aligned}
a(t) &= F_o \alpha \cos(\omega_i t) \\
v(t) &= \int_o^t a(t')dt' \\
&= F_o \omega_i \alpha \sin(\omega_i t) \\
d(t) &= \int_o^t v(t')dt' \\
&= \frac{F_o}{\omega_i^2} \alpha (1 - \cos(\omega_i t))
\end{aligned}
$$

where α represents the modal contribution from mode i at natural frequency ω_i, i.e., $\alpha = \phi_{ij}^2$. The analytic and numeric results for this case are shown in Figure 2-25.

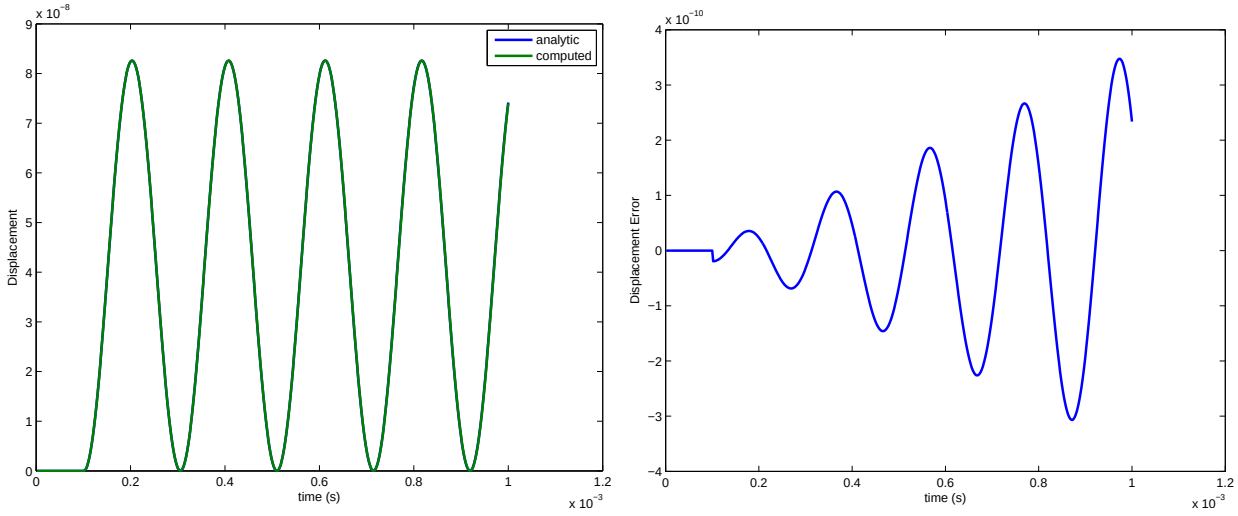


Figure 2-25. – Step Function response of Undamped Oscillator.

2.11.3. Damped Simple Harmonic Oscillator

The solution of the previous solution can be neatly modified by applying damping. The phase ϕ satisfies $\cos\phi = \zeta$. The analytic solution is,

$$x(t) = A\left(1 - e^{-\zeta\omega_i t}\frac{\sin\left(\sqrt{1-\zeta^2}\right)\omega_i t + \phi}{\sin(\phi)}\right) \tag{2.11.1}$$

Results for the analytic and numeric solutions are shown in Figure 2-26.

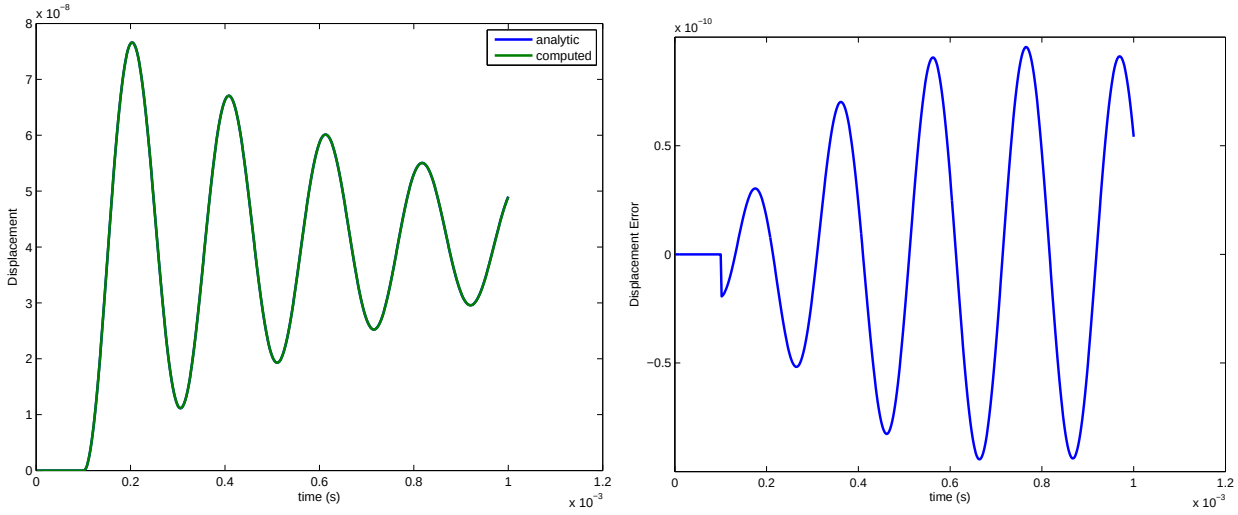


Figure 2-26. – Step Function response of Damped Oscillator.

2.11.4. Complex Loading

The last verification example (case 4) utilizes code to code comparison. We apply a triangle pulse of unit amplitude and duration 1 ms. Comparison is with the standard modaltransient method. This boundary condition is essentially an impulse which causes a linear increase in displacement. There is no difference between the modaltrans and modaltrans solutions. For input deck see Appendix 9.6.

2.12. Fluid Structure Interaction Added Mass

The following test is used to determine to what extent that SierraSD accounts for an added fluid mass to a structure when computing the angular frequencies. The test consists of a hollow steel sphere with a spring attached to the outer surface. Tests were run with the steel sphere submerged in water as shown in Figure 2-27 and a steel sphere with no added mass. The fluid is an acoustic medium.

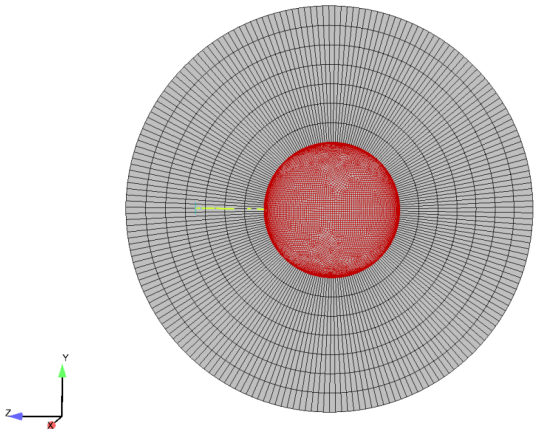


Figure 2-27. – Model of the hollow sphere and spring submerged in water.

The analytical solution for this test is based on the natural frequency equation of an object attached to an oscillating spring. Assuming that the spring is ideal, weightless, and without damping ω =:

$$\omega = \sqrt{\frac{K}{m}} \tag{2.12.1}$$

When the fluid is added around the sphere and is submerging the spring, the added mass must be accounted for. This changes ω to:

$$\omega = \sqrt{\frac{K}{m + m_a}} \tag{2.12.2}$$

The formulas for various shapes are documented [10]. In the case of a spherical structure, the added mass is given by:

$$m_a = \frac{2}{3}\pi\rho a^3 \tag{2.12.3}$$

The first mode computed in **Sierra/SD** should match the analytical solution.

There are several parameters for this test. A steel sphere is fixed in the x and y directions, allowing displacements only in the direction of the spring. Also, the steel sphere is constrained from rotating. The outer surface of the fluid region follows the Dirichlet boundary conditions where p=0. A fixed node attached to the end of the spring prevents translation. The only displacement

allowed in the system is in the direction of the spring. As a result, the global structure has no rigid body modes. The steel sphere has a high modulus of elasticity to ensure a very stiff structure. For this verification problem the steel sphere is essentially rigid.

The coupled structural acoustic system modes are determined by the quadratic eigenvalue problem, with gyroscopic (skew) coupling matrix, C, C is the gyroscopic coupling matrix.

$$(K + C\lambda + \lambda^2 M)u = 0. \tag{2.12.4}$$

As if there was no damping at all, $\lambda = i\omega$. Two solution methods are applied. Like ABAQUS, the modal projection algorithm for structural SA_eigen generates and solves the reduced sense problem. The Anasazi method the full problem without the a modal projection approximation. Modal projection is easier to use, but assessing its accuracy is more difficult.

Shell elements were investigated using and inner sphere of thickness as thin as 1.e-4. The number of modes, refinements, and test parameters varied to maximize accuracy. Table 2-10 summarizes the results. A visual representation of the frequencies in SierraSD using SA-eigen and Anasazi

Table 2-10. – Frequency results for SA-eigen, Anasazi, and analytical results.

Model	**Frequencies**									
Sphere size	0.1	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005	0.00025	0.0001
analytic	0.1529	0.1945	0.2385	0.2881	0.3136	0.3293	0.3400	0.3438	0.3458	0.3470
sa-eigen	0.1522	0.2040	0.2628	0.3381	0.3825	0.4123	0.4340	0.4419	0.4459	0.4480
anasazi	0.1477	0.1934	0.2412	0.2955	0.3237	0.3412	0.3532	0.3574	0.3595	0.3605

compared to the analytical solution is shown in Figure 2-28. For Anasazi, when the shell begins to get thick (above 0.010), the parameters have to be changed in order for the test to converge. The conditioning of the matrices begins to act up, so changing parameters such as young's modulus will help this. SA-Eigen will work for all models and parameters.

This figure shows that the impact of the fluid loading is largest for thin shells. The Anasazi method tracks the analytical solution well. Although SA_eigen solution is somewhat less accurate, but does capture the impact of the coupling.

Validation of the SierraSD code is most visible when the size of the inner steel sphere is the thinnest. When the steel sphere is very thin, the added mass has a greater impact on the results. The weight of the steel sphere will be considerably less than the weight of the surrounding fluid and the ratio between the mass added and the mass of the structure has an immense impact on the frequencies of the system. Figure 2-29 shows the comparison of having an added mass to your system and shows the results between SierraSD and the analytical solution.

This model was also investigated using hexahedron and tetrahedron solid elements with a QEVP/SA-eigen solution case. For thicker models using either solid element produced more accurate results. However, the overall system was to be modeled as a rigid body and when using the solid elements this process increased complexity as the steel sphere became increasingly thin. The number of elements increased exponentially with the thinner the structure. Also, adjustments to the parameters of the model had to be constantly maintained to ensure a stiff structure. For the shell elements, the thickness is defined in the input deck and the stiffness is easily accounted for. For input deck see Appendix 9.8.

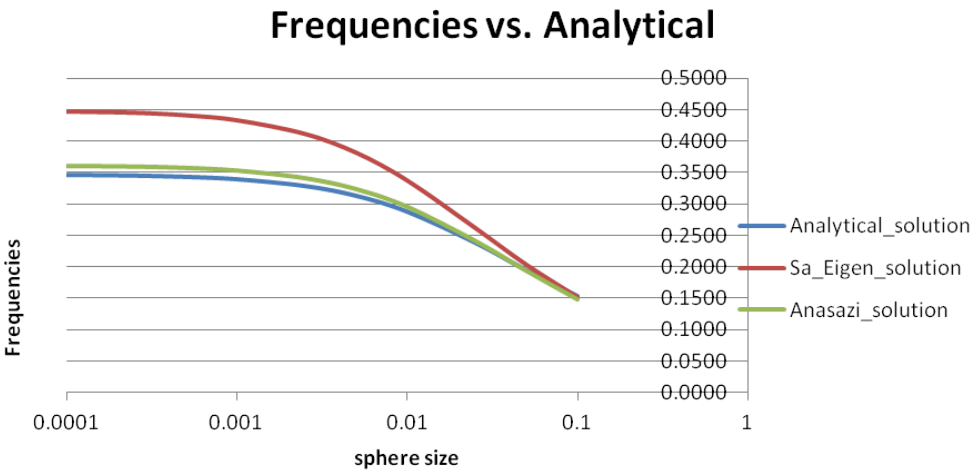


Figure 2-28. – Frequencies in SierraSD compared to the analytical solution.

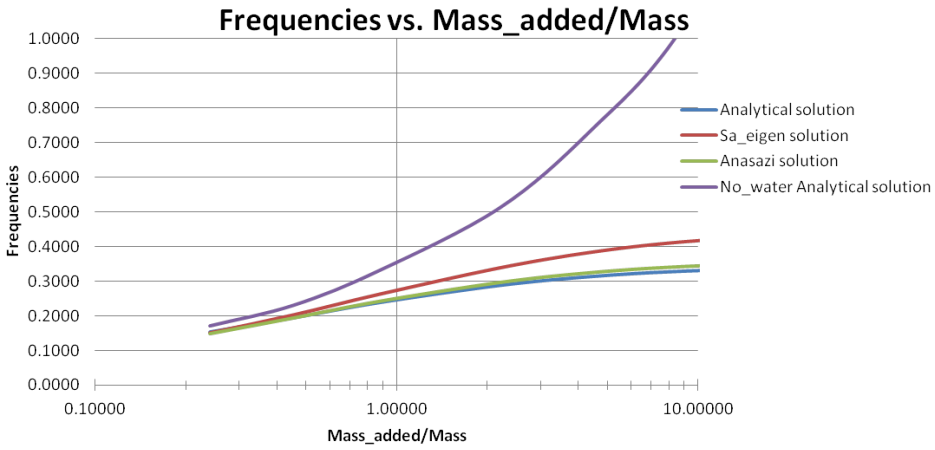


Figure 2-29. – Frequencies in SierraSD vs the mass ratio of the system.

2.13. Fluid Structure Cavitation

An important class of fluid-structure interaction (FSI) problems involve the numerical calculation of the response of a structure that is excited by a transient acoustic pressure wave. These complex models have been created and well represented with the development of the doubly asymptotic approximations that describe the fluid-structure interaction in terms of a radiation boundary that truncates the fluid-volume mesh to finite extent. In Sierra-SD we do not use the DAA, but apply a volumetric acoustic mesh with infinite elements representing the radiation boundary. A model was created in Sierra-SD that represents a solution that has already been obtained [9]. This is a one-dimensional problem, which involves a flat plate initially resting on the surface of a half space of fluid. An acoustic pressure wave is prescribed on the plate causing excitations that consist of a step-exponential plane wave superimposed upon an ambient hydrostatic pressure field. Figure 2-30 is an illustration of the model.

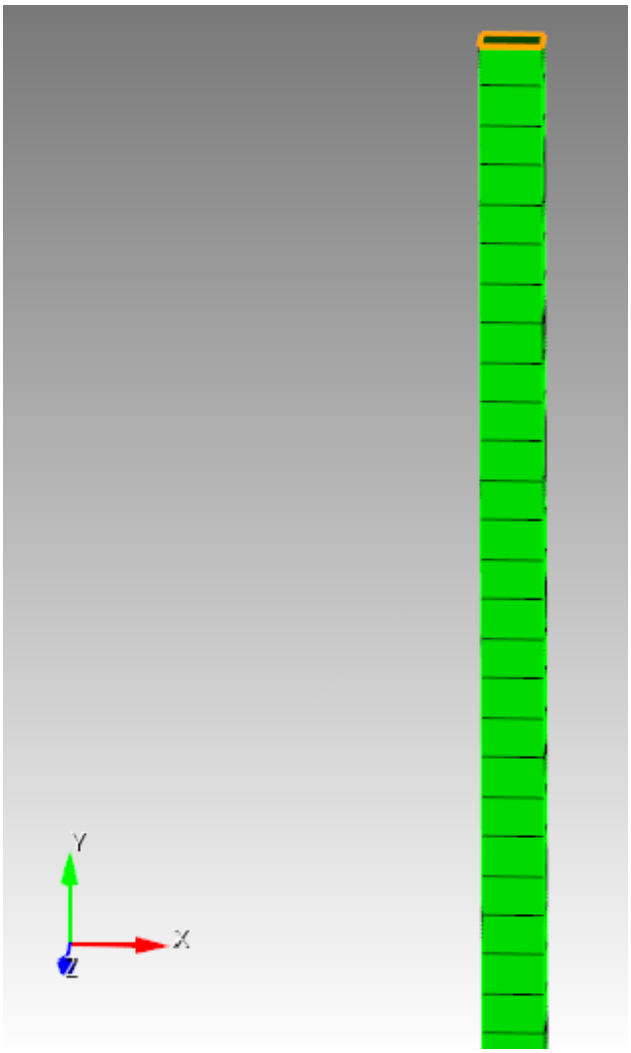


Figure 2-30. – 1D FSI Plate Shell Model in SD.

The model is a 1.5 inch by 1 inch rectangular plate. QuadT shell elements of unit thickness are

used. The fluid volume is 300 hex elements of similar rectangular dimensions. The boundary of the fluid mesh has infinite elements to serve as absorbing boundary conditions, as well as far-field calculators. Imperial units are used. The mass density of the plate was 5.329686e−4 lb sec^2in^4, while that of the fluid was 9.3455e−5 lb sec^2in^-4. The speed of sound of the fluid was 57120 in/sec.

A peak pressure of the incident wave that is applied to the plate is 103 psi with a decay time of 0.9958e−3 sec. For the transient analysis, 1200 time steps were used, with time step size 1.313e−5 sec.

The Sierra-SD results were compared to and verified against published results [21]. Figure 2-31 of the y component of velocity versus time reproduces the published results.

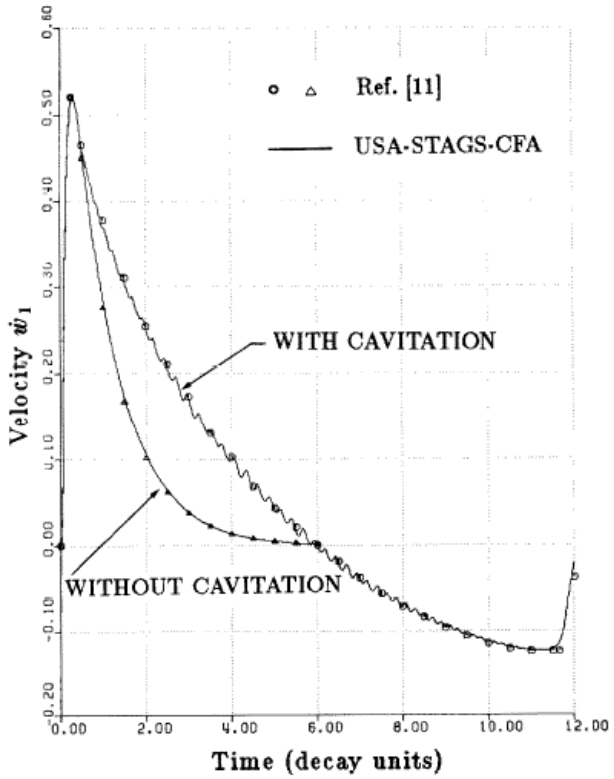


Figure 2-31. – Velocity vs Time, Results from Felippa and DeRuntz.

The model without cavitation was reproduced in Sierra-SD and compared to [21]. This is shown in Figure 2-32. The actual velocities in in/sec can be obtained by multiplying by 57.12, while the time scale is given in decay time units. The decay time units can be expressed as $t = 1/\lambda * (time)$. The velocity of the plate is essentially zero by six decay times.

Comparisons of the models is very good. For input deck see Appendix 9.9.

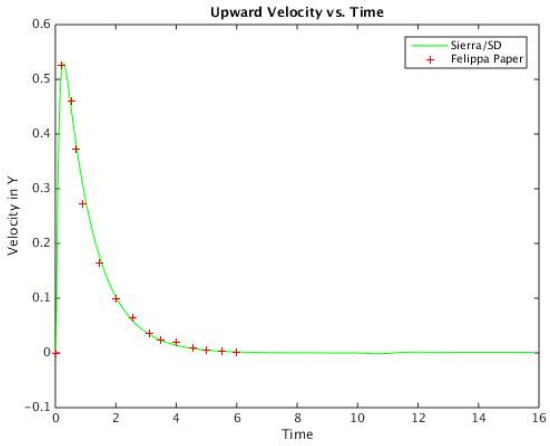


Figure 2-32. – Velocity vs Time, Results from Sierra-SD.

2.14. Buckling of Constant Pressure Ring

Most analytic solutions for linear buckling are derived using Euler-Bernoulli beam theory. These solutions are ideal for meshes built with beam and shell elements, but are only approximate verification examples for 3D solid meshes. In this section we present the buckling analytic solution of buckling of a circular ring. We only present the results using 3D solid elements. The model is shown in Figure 2-33.

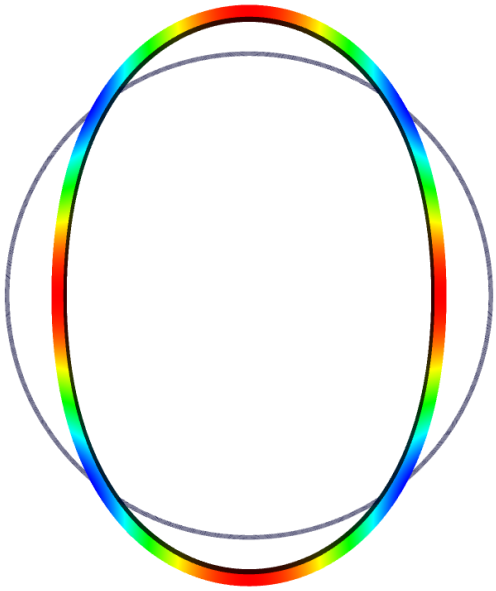


Figure 2-33. – Buckling Ring Example. Model parameters

Diameter: 40

Material: aluminum

Cross Section I: 1/12

Cross Section Area: 2.0

Cross Section Thickness: 1.0

In this example, we consider buckling of a circular ring subjected to a uniform, external pressure. The critical buckling pressure is given [50] as

$$P_{cr} = \frac{3EI}{R^3} \tag{2.14.1}$$

For the geometry of the problem, the critical buckling load is predicted to be

$$P_{cr} = \frac{3 \times 10^7 \times \frac{1}{12}}{20^3} = 312.5 \tag{2.14.2}$$

The computed buckling load was 395.408. Since the exact solution is for Euler-Bernoulli beam theory we expect some difference, however this may be a little too high. We will re-try with beam elements once they are on-line for buckling.

For input see Appendix 9.10

2.15. Buckling of a Cantilever Beam

The buckling of a cantilever beam modeled using solid elements is verified. The geometry for this example consists of a cantilever beam with one end clamped, and with the other subjected to a compressive load P. The buckling load predicted by Euler-Bernoulli beam theory is

$$P_{cr} = \frac{2.4674EI}{L^2}. \tag{2.15.1}$$

A simple mesh of this example was created, consisting of an $2 \times 2 \times 20$ hex elements. The critical buckling load is predicted to be

$$P_{cr} = \frac{2.4676 \times 30 \times 10^6 \times \frac{1}{12}}{10^2} = 61675 \tag{2.15.2}$$

The computed buckling load was 61370.1. The model is shown in Figure 2-34.

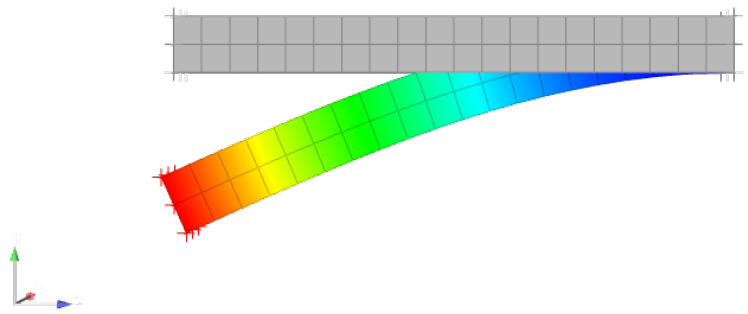


Figure 2-34. – Cantilever Beam Buckling Model parameters.

Material: steel

Length: 20

Area: 2×2

For input see Appendix 9.11

2.16. Eigenvalue Restart with Virtual Nodes and Elements

A transient restart model was created and tested including virtual nodes and elements, tied joints and superelements. The model is shown in Figure 2-36. For restart analysis two solution cases and input decks are needed. An initial simulation with restart=write has the output needed for to restart. The next simulation has restart=read. A 20 step simulation restarted from a 10 step simulation. This test includes superelements, infinite elements, and tied joints. A truth model was constructed with no restart and used for verification. Figure 2-35 shows the comparison of the truth model with no restart and the model with restart.

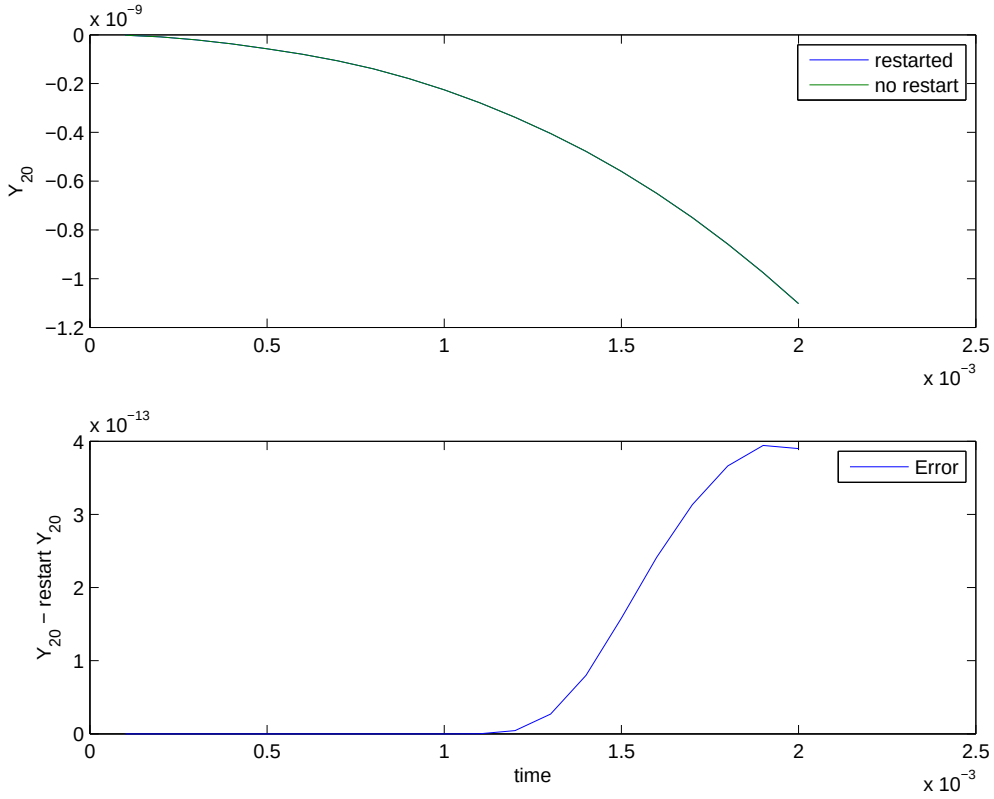


Figure 2-35. – Comparison between truth model and restart.

In Figure 2-35 the 20th node in the y direction was compared between the two cases. The error is on the magnitude of $10^{-}13$ which is expected due to the solvers, therefore, validating transient restart capability in Sierra-SD.

Eigen Restart This model was also analyzed using an eigen restart capability. The difference in this model is that there are no infinite elements only superelements and tied joints. This model was compared to a truth model and is showing accurate results. The transient and eigen restart tests were created and run in serial and in parallel.

For input deck see Appendix 9.12.

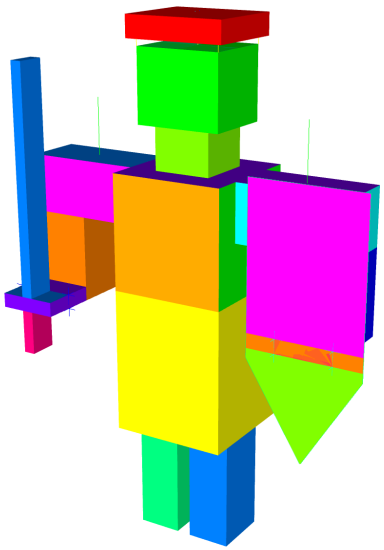


Figure 2-36. – Restart “Ninjabot” Mesh.

2.17. Solutions in Rotating Coordinate Frames

Sierra/SD supports solutions in a rotating coordinate frame. Tests in this section address this verification.

2.17.1. Rotating rigid bodies: statics

2.17.2. Model Description and Purpose

The model consists of a symmetric bar 6 units long with equal masses on either end. The bar is stationary in a rotating coordinate frame. To avoid singularities, the center point of the bar is clamped. The mass of the bar is zero. See Figure 2-37.

The test evaluates a very simple geometric problem, and insures that centrifugal forces are correctly applied to concentrated masses. It insures that rotations will work properly about the default coordinate axis.

Figure 2-37. – Dumbbell Geometry.

Analysis Type	linear statics
Element Type	Hex8
Loading	centrifugal
Keyword	centrifugal force

Each mass on either end of the rotating bar should experience only centrifugal boundary conditions. The left-hand side includes the centrifugal softening matrix (but no geometric stiffening). The magnitude of the loading is,

$$F_{axial} = \Omega \times (\Omega \times \vec{r})\, \Delta M$$

where,

Ω = 1.1 in the Z direction.

$\vec{r}$ is 3.0, radial direction.

ΔM is 2.0

The resulting force is 7.26 units in the radial direction. It is applied only at the end nodes where the concentrated mass is located. For input deck see Appendix 9.13.

2.17.3. Rotating rigid bodies: transient

It is useful to verify a simple point mass in a rotating frame. We consider a system rotating at a constant angular velocity $\vec{\Omega} = \Omega\hat{k}$. The angular acceleration is zero. A single point mass, m, is observed in the rotating frame. The point mass is frictionless. The geometry is illustrated in Figure 2-38.

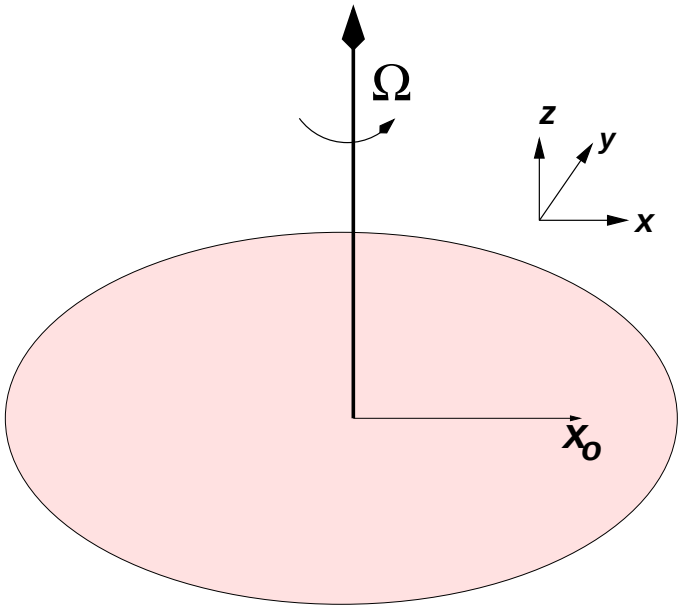


Figure 2-38. – Rotating Frame Geometry.

2.17.4. Mass at Rest in Inertial Frame

This is by far, the simplest case. In the inertial frame we have a mass located at $(x_o, 0)$. It does not move. In the rotating frame, $r = x_o$ and $\theta' = -\Omega t$', or in the Cartesian rotating frame,

$$x' = x_o \cos(-\Omega t) \tag{2.17.1}$$

$$y' = x_o \sin(-\Omega t) \tag{2.17.2}$$

2.17.5. Mass Initially at Rest in Rotating Frame

We consider a mass initially at the point $(x_o, 0)$ with an initial velocity of $\vec{v} = \Omega x_o \hat{e}_y$. In the rotating frame this mass appears initially at rest at location $(x'_o, 0)$. However, because of the rotation of the frame, the mass will begin to move away from the center of the rotating frame.

2.17.5.1. Without Coriolis Contribution

In the rotating frame, the only force acting on the body is the centrifugal force, $\vec{\Omega} \times (\vec{\Omega} \times \vec{r})$. As all the forces are in the radial direction, the differential equation reduces to a single degree of freedom system.

$$m\ddot{r} = \Omega^2 r$$

This equation is very similar to that of a harmonic oscillator. With the given initial conditions the solution is,

$$r = x_o cosh(\Omega t)$$

where $cosh()$ is the hyperbolic cosine.

This solution is not physical, as there is no Coriolis force. At time progresses, the velocity continues to grow unbounded, but the angular position remains zero.

2.17.5.2. With Coriolis Contribution

We solve this by computing the solution in the inertial coordinate system, and transforming back to the rotating frame.

In the inertial frame, there are no forces acting on the body. The solution in Cartesian frame is,

$$\begin{aligned} x &= x_o && (2.17.3)\\ y &= v_o t && (2.17.4)\\ &= x_o \Omega t && (2.17.5) \end{aligned}$$

This may be transformed to polar coordinates, still in the inertial frame.

$$\begin{aligned} r &= \sqrt{x^2 + y^2} && (2.17.6)\\ &= x_o \sqrt{1 + (\Omega t)^2} && (2.17.7)\\ \theta &= \tan^{-1}(y/x) && (2.17.8)\\ &= \tan^{-1}(\Omega t) && (2.17.9) \end{aligned}$$

We use the relation that $\theta' = \theta - \Omega t$ and $r' = r$. Then,

$$\theta' = \tan^{-1}(\Omega t) - \Omega t$$

This solution may then be transformed to rotating cartesian frame in the usual way.

Small Angle Approximations

For solutions with $\Omega t << 1$ the solutions in this section may be compared with the previous section. We use,

$$\tan^{-1}(\theta) \approx \theta - \frac{\theta^3}{3} + \frac{\theta^5}{5} + \ldots \tag{2.17.10}$$

$$\sqrt{1+\epsilon^2} \approx 1 + \frac{\epsilon^2}{2} \tag{2.17.11}$$

$$\tag{2.17.12}$$

For both solutions,

$$r' \approx x_o \left(1 + \frac{(\Omega t)^2}{2}\right),$$

while $\theta' = 0$ with no Coriolis term. Including the Coriolis term we obtain,

$$\theta' \approx \frac{-(\Omega t)^3}{3}$$

Figure 2-39 shows the solution to this problem. Both analytic and finite element solutions are shown. A good degree of agreement is obtained even for a very large displacement.

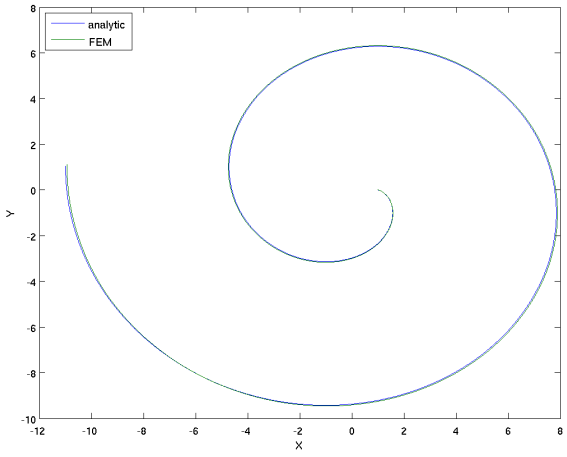


Figure 2-39. – Point Mass initially at rest in rotating frame.

2.17.6. Mass Moving in the X axis

This example starts at the same location, i.e., $(x_o, 0)$ in the inertial frame, but the initial velocity in the inertial frame is $-2\,x_o\Omega/\pi\hat{e}_x$. Thus, at time $\Omega t = \pi/2$, the mass will be at the origin. At time $\Omega t = \pi$, the particle will be located at $(-x_o, 0)$. In the inertial frame,

$$y = 0 \tag{2.17.13}$$

$$x = x_o(1 - 2\Omega t/\pi) \tag{2.17.14}$$

or,

$$r = x_o(1 - 2\Omega t/\pi) \tag{2.17.15}$$
$$\theta = 0 \tag{2.17.16}$$

In the rotating frame, $r' = r$ and $\theta' = -\Omega t$. The Cartesian description is therefore,

$$x' = x_o(1 - 2\Omega t/\pi)\cos(\Omega t) \tag{2.17.17}$$
$$y' = x_o(1 - 2\Omega t/\pi)\sin(\Omega t) \tag{2.17.18}$$

For input deck see Appendix 9.14.

2.17.7. Angular Velocity, Beams, Statics

Consider a beam of length L with constant cross-sectional area A, elastic modulus E, and mass density ρ. The root of the beam is at $x = 0$ and its tip at $x = L$. The axis of rotation passes through the origin and is in the z-direction. Further, the angular velocity is constant and denoted by Ω.

With the assumption of all mass being concentrated along the axis of the beam, the net force in the x-direction at radial position r is given by

$$\begin{aligned} F &= \Omega^2 \int_r^L x \, dm \\ &= \Omega^2 \int_r^L x \rho A \, dx \\ &= \rho A \Omega^2 (L^2 - r^2)/2. \end{aligned} \tag{2.17.19}$$

Thus, the axial stress at r is

$$\sigma(r) = F/A = \rho \Omega^2 (L^2 - r^2)/2. \tag{2.17.20}$$

The axial strain is assumed constant across each cross-section and given by

$$\epsilon(r) = \sigma(r)/E = \rho \Omega^2 (L^2 - r^2)/(2E). \tag{2.17.21}$$

The axial displacement is obtained by integrating the axial strain. Since the axial displacement vanishes at $x = 0$, we obtain

$$\begin{aligned} u(r) &= \int_0^r \epsilon(x) \, dx \\ &= \rho \Omega^2/(2E) \int_0^r (L^2 - x^2) \, dx \\ &= \rho \Omega^2/(2E)(L^2 x - x^3/3)|_0^r \\ &= \frac{\rho \Omega^2 L^3}{6E} [3(r/L) - (r/L)^3]. \end{aligned} \tag{2.17.22}$$

For input deck see Appendix 9.15

2.17.8. Angular Acceleration, Statics

We consider a beam with its root at the origin and its free end at $(L, 0, 0)$. Unit vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ are fixed in a rotating coordinate system, which is attached to the root of the beam. The angular acceleration of the rotating coordinate system is $\dot{\Omega} \boldsymbol{b}_3$, while axial and transverse displacements are denoted $u_1(x)\boldsymbol{b}_1$ and $u_2(x)\boldsymbol{b}_2$, respectively, where x is the position along the beam. The axis of rotation of the rotating coordinate system is located at $(0, d, 0)$

When considering the effects of angular acceleration (see Theory Manual), the governing differential equations for static analysis of an Euler-Bernoulli beam model are given by

$$-\frac{d}{dx}\left(AE\frac{du_1(x)}{dx}\right) - \rho A\dot{\Omega}u_2(x) = -\rho A\dot{\Omega}d,$$
$$\frac{d^2}{dx^2}\left(EI\frac{d^2u_2(x)}{dx^2}\right) + \rho A\dot{\Omega}u_1(x) = -\rho A\dot{\Omega}x,$$

where A is the cross section area, I is the cross section moment of inertia, E is Young's modulus, and ρ is the mass density. All these properties are assumed to be constant.

The presence of $\dot{\Omega}$ causes coupling between $u_1(x)$ and $u_2(x)$. As a result, closed-form solutions for $u_1(x)$ and $u_2(x)$ are not obvious. Thus, we consider approximate solutions of the form

$$u_1(x) = \sum_{i=1}^{n} u_{1i}\phi_i(x), \quad u_2(x) = \sum_{i=1}^{n} u_{2i}\phi_i(x),$$

where $u_{1i} = u_1(x_i)$, $u_{2i} = u_2(x_i)$, and $x_1, \ldots, x_n$ are the positions of the nodes. The $\phi_i(x)$ are shape functions associated with the spectral element method. That is, each $\phi_i(x)$ is a polynomial of degree $n-1$ and the x_i are the Gauss-Lobatto points transformed from $\eta \in [-1, 1]$ to $x \in [0, L]$. We consider only a single element.

A Galerkin method is used whereby the test functions are the same as the trial functions $\phi_i(x)$. The basic idea is to substitute the above expressions for $u_1(x)$ and $u_2(x)$ into the governing equations for axial and transverse displacements, multiply each equation by ϕ_j for $j = 1, \ldots, n$, integrate by parts, and apply boundary conditions. The net result is the linear system of equations

$$(K_m + K_a)u = g,$$

where $u = [u_{11}, u_{12}, \ldots, u_{1n}, u_{21}, u_{22}, \ldots, u_{2n}]^T$ is a vector of nodal displacements,

$$K_m = \begin{bmatrix} \int_0^L AE\phi'(x)\phi'^T(x)\,dx & 0 \\ 0 & \int_0^L EI\phi''(x)\phi''^T(x)\,dx \end{bmatrix}$$

with $\phi(x) = \begin{bmatrix} \phi_1(x) & \ldots & \phi_n(x) \end{bmatrix}^T$ and $\phi'(x)$ denotes the derivative of $\phi(x)$ with respect to x. Similarly,

$$K_a = \begin{bmatrix} 0 & -\dot{\Omega}\int_0^L \rho\phi(x)\phi^T(x)\,dx \\ \dot{\Omega}\int_0^L \rho\phi(x)\phi^T(x)\,dx & \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\Omega}M \\ \dot{\Omega}M & 0 \end{bmatrix},$$

where

$$M = \int_0^L \rho\phi(x)\phi^T(x)\,dx.$$

The right hand side vector g is obtained using standard methods and is given by

$$g = \begin{bmatrix} P\phi(L) - \dot{\Omega}\int_0^L \rho A d\phi(x)\,dx \\ -\dot{\Omega}\int_0^L \rho Ax\phi(x)\,dx \end{bmatrix},$$

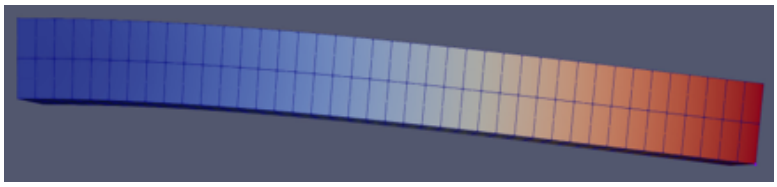


Figure 2-40. – HEX20 mesh used for angular acceleration statics verification problem.

where $P\boldsymbol{b}_1$ is the concentrated axial load applied to the beam tip at $x = L$. The boundary conditions $u_1(0) = 0$, $u_2(0) = 0$, and $u_2'(0) = 0$ can be enforced by constraint equations of the form

$$Cu = 0$$

where the constraint matrix C has 3 rows and $2n$ columns. The total potential energy for the structure is given by

$$W = u^T(K_m + K_a)u/2 - u^T g,$$

and the associated Lagrangian is

$$G = W + \lambda^T Cu,$$

where λ is a vector of Lagrange multipliers. The stationarity conditions for the Lagrangian are $\partial G/\partial u = 0$ and $\partial G/\partial \lambda = 0$, which give us

$$\begin{bmatrix} K & C^T \\ C^T & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}, \tag{2.17.23}$$

where $K = K_m + K_a$. We wrote a Python script to construct and solve the linear system in (2.17.23).

Several different Sierra/SD element types were considered for this verification problem. As an illustration, the HEX20 mesh used is shown in Figure 2-40. Comparisons between Sierra/SD and our semi-analytical solution (Python) for axial and transverse deformations are shown in Figure 2-41 for meshes of BEAM2, HEX8, HEX20 and TET10 elements. Notice the very good agreement for all these element types. Similar results are shown in Figure 2-42 for TRIA3 shell elements. The results for the shell model are different from the previous ones because cross section properties needed to be adjusted in order for thin shell assumptions to hold. Again, there is very good agreement between the semi-analytical and Sierra/SD results.

For the BEAM2 input deck, see Appendix 9.16
For the HEX20 input deck, see Appendix 9.17
For the TRIA3 input deck, see Appendix 9.18

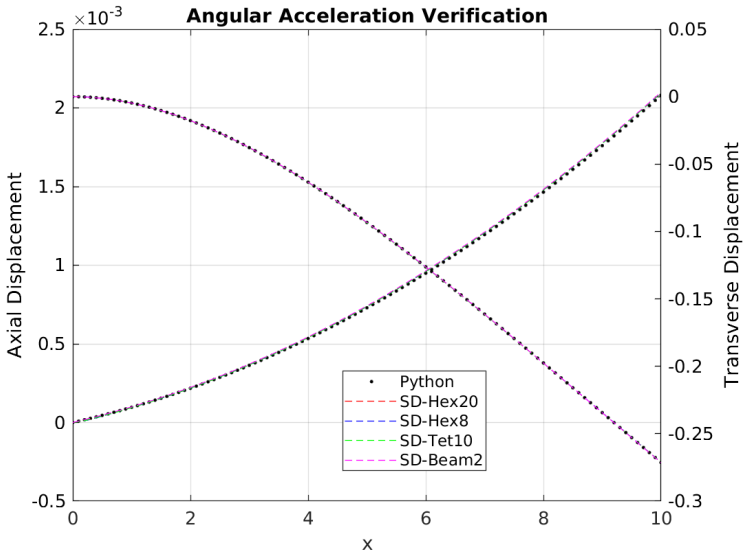


Figure 2-41. – Comparisons of axial and transverse displacements of BEAM2, HEX8, HEX20 and TET10 models with semi-analytical solution.

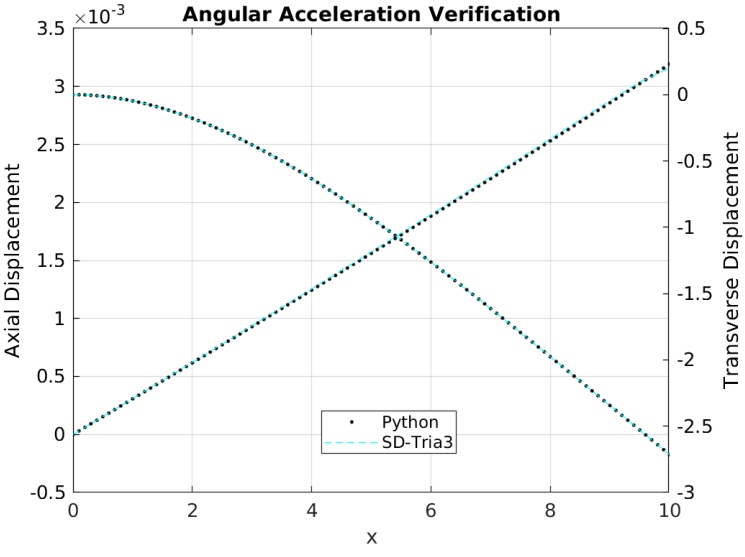


Figure 2-42. – Comparisons of axial and transverse displacements of a TRIA3 shell model with semi-analytical solution.

2.17.9. Rotating Shell Statics

The rotating shell example is similar to the rotating beam 2.17.7 with $E = 19.5 \times 10^{10}$, $L = 10$, $\rho = 7700$, $A = 1$, and $\Omega = 5$. The beam is aligned with the x-axis, and has dimensions $10 \times 1 \times 1$. A 50 by 5 by 5 tensor product mesh, a 50 by 5 quadrilateral mesh and a 50 element beam mesh are studied. The 2D mesh is in the x y plane. Comparisons of axial deformations for three finite element analyses are shown in Figures 2-43 and 2-44. Notice that all three finite element results are close to the exact solution, with the `QUADT` results being the least accurate. We note that much more accurate results were obtained when the `QUADT` elements were replaced by `NQUAD` elements. We think that the less accurate predictions for the mesh of `QUADT` elements is caused by anisotropies introduced by representing each quadrilateral element as the union of two triangular elements.

For input deck see Appendix 9.19.

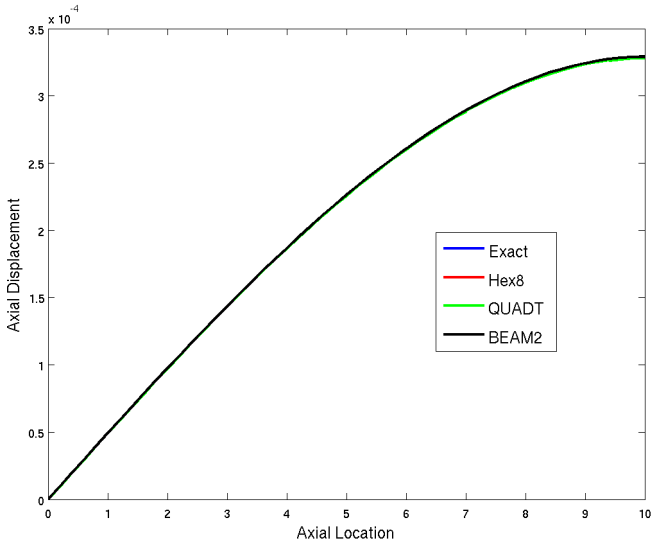


Figure 2-43. – Comparisons of axial deformations with exact solution for a beam.

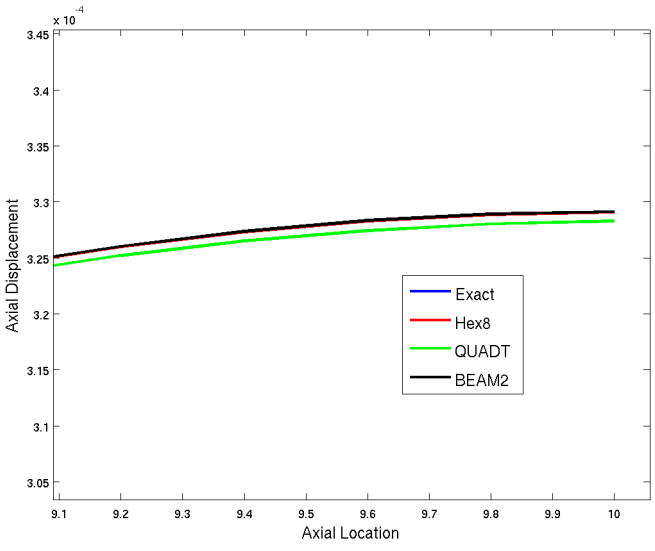


Figure 2-44. – Zoomed in view of Figure 2-43 showing differences for QUADT elements.

2.17.10. Rotating Ring Statics

This test examines a simple ring in a rotating frame. Verification of the *forces* is made here. We intentionally do *not* verify displacements as these depend on the element formulation.

The ring, shown in Figure 2-45 is a two unit radius thin structure. A constant angular velocity, $\vec{\Omega}$, is applied at 1.1 radians per second in the Z direction. The ring is not centered on the origin, but is centered on a user defined coordinate system.

The resulting forces are given by,

$$\begin{aligned}\vec{F} &= \int \rho\vec{\Omega}\times\vec{\Omega}\times\vec{r}\,dV \\ &= 1.1^2 2.0\,(\rho V_n)\,\hat{r}\end{aligned}$$

Where ρV_n represents the mass associated with a node. For this model, there are 148 nodes on the ring which each share equally the total ring mass of 12.5626 units. The resulting force is 0.2054 units outward.

For the Euler force,

$$\begin{aligned}\vec{F} &= \int \rho\frac{d}{dt}\vec{\Omega}\times\vec{r}\,dV \\ &= 1.12.0\,(\rho V_n)\,\hat{r}\end{aligned}$$

and the resulting force is 0.2054/1.1 units outward.

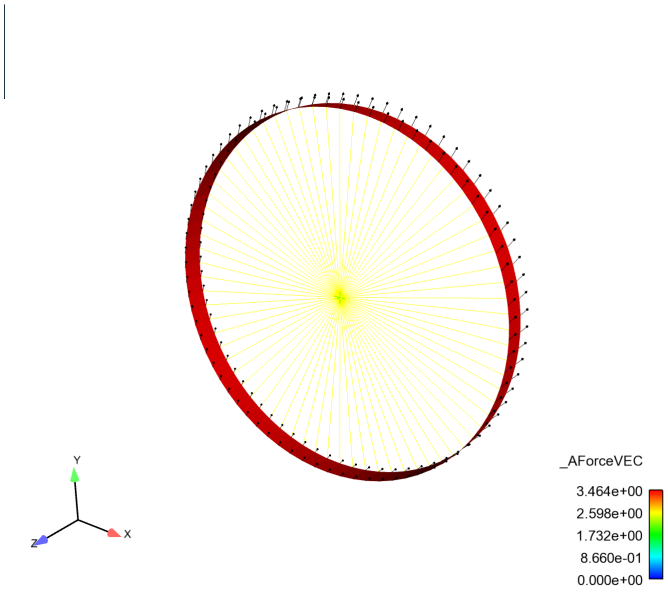


Figure 2-45. – Rotating Ring Geometry and Results.

The test evaluates:

- Centrifugal force in a rotating system,

- Force on shells with rotational degrees of freedom. The moment vanishes,
- And coordinate translation.

It does *not* test,

- Coordinate rotation.
- Solid or point mass elements.
- Solution when there is no symmetry.

For input deck see Appendix 9.20.

2.17.11. Rotating Ring Acceleration

This is a variation on the static analyses of a rotating ring described in Section 2.17.10. Here an angular acceleration is applied instead of the angular velocity in 2.17.10. By hand, the angular acceleration is .2054/1.1 force units. For input deck see Appendix 9.21.

2.17.12. Rotating Superelement Statics

Models (including Superelements) must be loaded by centrifugal forces if they are to be analyzed in a rotating coordinate frame. However, as discussed in the analysis section, there are inherent problems in loading a Superelement with a centrifugal force. In this test, we examine one case where the loading is exact.

The model is a single hex element which is rotated about an edge. The unreduced model force may be computed as,

$$\begin{aligned} F_{centrifugal} &= \vec{\Omega} \times (\vec{\Omega} \times \vec{r}) \\ &= [\Omega]^T [M][\Omega][r] \end{aligned}$$

where $\vec{\Omega}$ is the angular velocity vector, $[\Omega]$ is a rotation matrix, $[M]$ is the mass matrix and $[r]$ represents the position coordinates. This solution is as accurate as possible for a finite element representation of the continuous model. See details in the theory manual.

The geometry is shown in Figure 2-46.

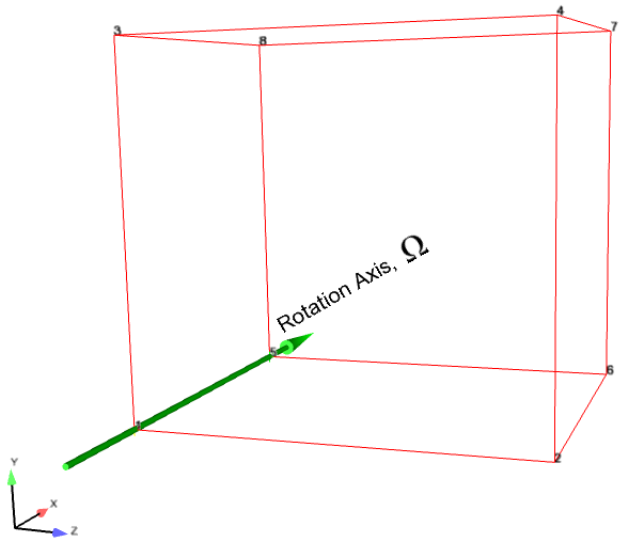


Figure 2-46. – Rotating Hex Geometry.

2.17.13. Tests

We evaluate several steps of the test.

1. We look at the loading of a single hex in rotation. This is our truth model.
2. We ensure that the model reduction process is consistent.
3. We examine the loading of a Superelement equivalent to the single hex element.

Each test is described in a little more detail in what follows.

Single Hex Rotation

Equation 2.17.12 describes the load calculation for a single hex in a coordinate frame rotating at a constant angular velocity. Visually the results of the loading appear reasonable. As expected, the loading is not purely radial.

The analysis is singular, i.e., the body has a zero energy mode and is free to rotate about the axis of rotation. The displacements are solutions of a singular linear system. Because of this, only the forces are evaluated. The force response is shown in Figure 2-47.

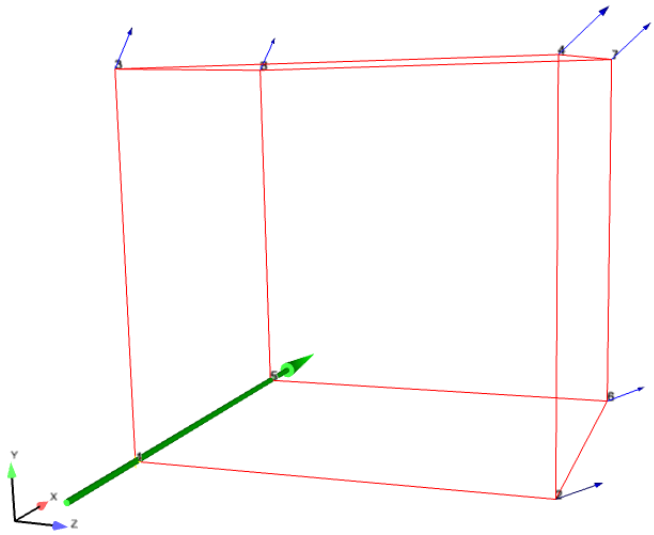


Figure 2-47. – Rotating Hex Response.

2.17.13.1. Superelement Reduction

A critical part of this evaluation is "reduction" of the hex to a Superelement. In most such reductions, a combination of interior "fixed interface" modes are combined with interface or "constraint" modes to generate a reduced basis. Here we have no interior modes and all the interface nodes are retained. One of those nodes has only 2 degrees of freedom, so there is a slight reduction. As a consequence, the Superelement model is of dimension 23, while the original hex has 24 degrees of freedom. The most important point is that the Superelement model may now be run through the software, which follows an entirely different path from the original.

In the final stage, the Superelement is inserted into a new model. In this case, we re-use the original mesh. However, the block definitions are those of a Superelement. The Superelement is loaded using the centrifugal force routines. The result *must* be identical to the original test.

2.17.14. Analysis

Superelements are problematic for computation of internal integrals. Typically, all the internal shape functions and data are available *only* during the Superelement reduction stage. During subsequent analyses, only the interface information and reduced order matrices are retained.

For computation of the centrifugal force, an integral must be evaluated over the volume of the element.

$$\vec{f} = \rho \int_{element} \vec{\Omega} \times (\vec{\Omega} \times \vec{r}) \, dV$$

If the shape functions are available, this can be evaluated using equation (2.17.12). However, the model reduction process condenses out information from the interior of the Superelement to the nodes of the interface. The total mass is conserved, but information required to compute the interior integrals is no longer available. This verification test is structured so that no internal information is lost and the integrals may be computed exactly.

This set of tests insures the following:

- The software can successfully exercise a Superelement.
- Identical results are obtained to the original hex, indicating no transposing of degrees of freedom.
- All the nodes on the interface are being exercised.
- Superelements are supported with other than 3 DOFs on a node. Node 1 has 2 degrees of freedom, and there are 23 degrees of freedom total.

However, because of the details of the test, we do not evaluate the following:

- Superelements with internal degrees of freedom.
- Superelements with a reduced set of interface nodes.

For input deck see Appendix 9.22.

2.17.15. Rotating Superelement Beam Statics

We build on the analysis of a rotating hex beam. As shown in the hex beam example, the analytic solution can be written,

$$\begin{aligned} u(r) &= \int_0^r \epsilon(x)\, dx \\ &= \rho\Omega^2/(2E) \int_0^r (L^2 - x^2)\, dx \\ &= \rho\Omega^2/(2E)(L^2 x - x^3/3)|_0^r \\ &= \frac{\rho\Omega^2 L^3}{6E}[3(r/L) - (r/L)^3]. \end{aligned}$$

We next consider an example with $E = 19.5 \times 10^{10}$, $L = 10$, $\rho = 7700$, $A = 1$, and $\Omega = 5$. A superelement is generated by extracting all the nodes down the center of the beam. There are 101 nodes retained in the superelement, with 40 generalized degrees of freedom associated with fixed interface modes. Comparison of axial deformations for the finite element analysis is shown in Figure 2-48. Finite element results are close to the exact solution, but there differences because the superelement integration is not fully accurate for computation of centrifugal force moments.

For input deck see Appendix 9.23.

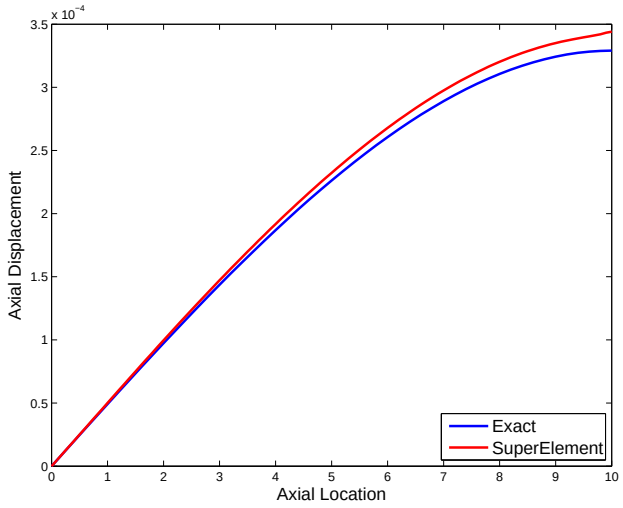


Figure 2-48. – Comparisons of axial deformations with exact solution for a beam.

2.18. High Cycle Fatigue and Damage

A single hex (and single degree of freedom) model is used to verify the computations of the random vibration problem. Four nodes of the 8 node brick are clamped. The remaining nodes are constrained to move in only the X direction. In addition, multipoint constraints tie three nodes to a single master node. The model has only one active degree of freedom, and a single element. Each of the results may be examined individually without a need for a summation over mode shapes.

Comparison is made to a MATLAB calculation found in "byhand.m". Each result is listed in following paragraphs.

Eigenvector normalization is checked using the mass matrix. From the output of `Maa.m`, the mass is 8.6333e-5. The eigenvector, ϕ, is of length 1, and value 107.6244. Then,

$$\phi^T m \phi = 107.6244 \cdot 8.6333\text{e-}5 \cdot 107.6244 = 1$$

The eigenvalues and vectors may be compared with results in `onehex-eig.exo`.

The modal transfer function at frequency ω describes the contribution of one mode to the resulting displacement,

$$H_i(\omega) = \frac{1}{\omega^2 - \omega_i^2}, \quad u = \sum_i^{Nmodes} H_i(\omega) F_i(\omega).$$

In our example the sampling frequency is 10:100 Hz, while the modal frequency is 62,846. Thus, $\omega_i >> \omega$. We can approximate,

$$H_i = 1/(2\pi \cdot 62846)^2 \approx 6.4133\text{e-}12$$

Thus, the modal amplitude is,

$$u_i = H_i F_i \approx \phi^T F / \omega_i^2 \approx 6.2121\text{e-}9.$$

The modal amplitude for FRF is not directly output, but the physical amplitude is output.

2.18.1. Determine the physical transfer function, H(ω) and Displacement

Physical space is simply related to modal space, $x = \phi q$. Likewise,

$$U = HF$$

or,

$$H(\omega) = \phi H_i \approx \phi^2 / (2\pi f)^2$$

Thus, the physical transfer function, $H \approx 7.4286\text{E-}8$. Likewise, the amplitude is the transfer function multiplied by the force.

$$U(\omega) = H(\omega) * Force \approx \phi F \phi^T / \omega_i^2 \approx 6.6857\text{E-}7,$$

and is essentially independent of frequency. This physical amplitude may be compared with results in `onehexran-frf.frq`. Salinas computes: 6.6857E-07.

Likewise, the acceleration response can be predicted. The acceleration is simply ω^2 times the displacement. At $f = 10$, $\ddot{U} = 4\pi^2\phi^2 F/\omega_i^2$. At $f = 10$, $\ddot{U}(10) = 0.0026394$. At the top end of frequency band, $\ddot{U}(100) = 0.26394$.

Salinas computes 0.0026394 and 0.26394.

2.18.2. Determine the Displacement and Acceleration Spectral Density

The output is generated by a computation of a modal sum.

$$X_{rms} = \sqrt{\sum_{i,j}^{Nmodes} \phi_i \phi_j \Gamma_{ij}}$$

Here Γ contains the integral of the frequency component of the load.

$$\Gamma_{ij} = \int_0^\infty H_i(\omega) H_j(\omega) S(\omega) d\omega$$

And, S is the PSD of the input force. The similar relation for acceleration includes ω^4,

$$\Gamma A_{ij} = \int_0^\infty \omega^4 H_i(\omega) H_j(\omega) S(\omega) d\omega$$

We use a simple trapezoidal integration strategy. Thus, we can weight the final and initial intervals at half the value of the central intervals.

$$\begin{aligned}
\Gamma &\approx 2\pi \sum_f H_q^2 S \Delta f w_i \\
&\approx 2\pi H_q^2 9.0[5\ 10\ 10\ 10\ 10\ 10\ 10\ 10\ 10\ 5] \\
&\approx 2\pi (2.7249\text{e-}8)^2 (9.0)(90) \\
&\approx 3.7789\text{e-}12
\end{aligned}$$

Likewise

$$\begin{aligned}
\Gamma_A &\approx (2\pi)^5 \sum_f H_q^2 S f^4 \Delta f w_i \\
&\approx (2\pi)^5 H_q^2 9.0[10\ 20\ 30\ 40\ 50\ 60\ 70\ 80\ 90\ 100]^4 [5\ 10\ 10\ 10\ 10\ 10\ 10\ 10\ 10\ 5] \\
&\approx (2\pi)^5 (2.7249\text{e-}8)^2 (9.0) 2.0332\text{e}9 \\
&\approx 0.13306
\end{aligned}$$

The ratio of GammaA/Gamma is 3.5211e10. This same ratio should be found in the square of A_{rms}/X_{rms} found in the random vibration output of `onehex-ran.exo`.

Salinas has: $X_{rms} = 1.4799\text{E-}5$. $A_{rms} = 2.7770$. These are found in onehex-ran.exo. The ratio $(A_{rms}/X_{rms})^2 = 3.5212\text{E}10$.

2.18.3. Fatigue Parameters

For fatigue life predictions, we are interested in several parameters. The first of these is the stress moments, M_o, M_2 and M_4. These are important as the ratios of these moments provide information on the rate of zero crossing, ν_o^+, and the number of zero crossings, $n_c = \nu_o^+ \tau$.

The ratio of `Vrms2/Vrms` is related to ratios of moments. In particular, $VRMS2/VRMS = \sqrt{M_2/M_o}$. These are related to the ratios of Γ_v/Γ.

$$\begin{aligned}\Gamma_V &\approx (2\pi)^4 \sum_f H_q^2 S f^2 \Delta f w_i \\ &\approx (2\pi)^3 H_q^2 9.0\,[10\,20\,30\,40\,50\,60\,70\,80\,90\,100]^2\,[5\,10\,10\,10\,10\,10\,10\,10\,10\,5] \\ &\approx 5.5447\text{E-}7\end{aligned}$$

Salinas has $VRMS$ = 1.1384E2 and $VRMS2$ = 4.3607E4. Therefor $(VRMS2/VRMS)^2$ = 1.4673E5, which can be compared to the closed form ratio 5.5447E-7 / 3.7789E-12 = 1.4673E5.

This is `Salinas_rtest/verification/fatigue/onedof/onehexran.test`.

2.18.4. Fatigue Solution

One unit cube Hex8 element is used in the fatigue analysis verification. This is an entirely contrived example, with material properties invented to simplify the calculation. Results from within Sierra/SD are compared to independent MATLAB computations.

2.18.4.1. Assumptions

We begin the solution with a previously verified random vibration solution with results in Table 2-11.

Variable	Value
Vrms	113.8421029
Vrms2	4.360736489E+04
Vrms4	2.136176695E+07

Table 2-11. – Input Moments.

We also *construct* a fictitious material with S-N curves that make computation simple. The S-N curve is represented in Figure 2-49. It is constructed such that with an RMS value of stress equal to 113.8421029, a solution of N of 1 million is obtained. The associated material parameters are listed in Table 2-12.

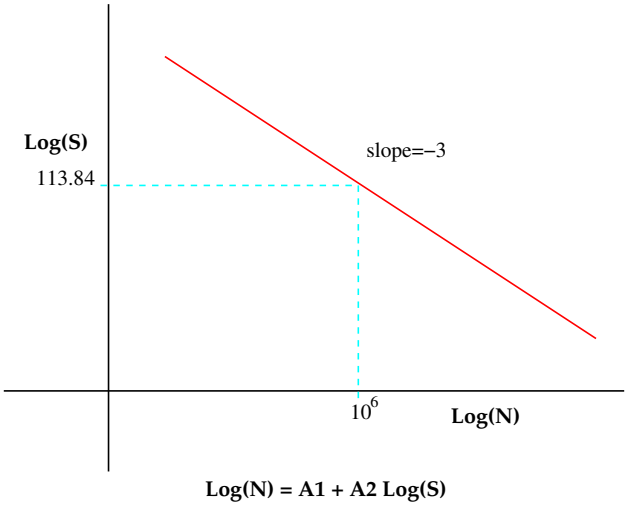


Figure 2-49. – S-N Curve for Fictitious Material.

Variable	Value
A2	-3
m	3
A1	Log(N)-A2*Log(113.84) $\approx$ 12.1

Table 2-12. – Fatigue Material Parameters.

Damage Rate Calculation The narrow band damage rate is,

$$D_{NB} = \frac{\nu_o^+}{10^{A_1}} \left(\sqrt{2} V_{rms}\right)^m \Gamma\left(1 + m/2\right)$$

This may be evaluated in terms of the above parameters.

$$\begin{aligned} \nu_o^+ &= \frac{V_{rms2}}{2\pi V_{rms}} \approx 61.0 \\ 10^{A_1} &\approx 1.475 \times 10^{12} \\ (\sqrt{2} V_{rms})^m &\approx 4.173 \times 10^6 \\ \Gamma\left(1 + m/2\right) &= \Gamma(2.5) \approx 1.3293 \end{aligned}$$

For which we have $D_{NB} \approx 2.2919 \times 10^{-4}$.

This is the test `Salinas_rtest/verification/fatigue/onedof/onehexfatigue.test` `onehexran.test`. For input deck see Appendix 9.24.

2.18.5. Fatigue Stress Scaling

We verify the fatigue analysis scaling on a single, 1x1x1 Hex8 element. This is an entirely contrived example, with material properties invented to simplify the calculation. The model is

identical to a fatigue example previously verified, we simply scale the geometry and loads, and verify the solution. The experimental material data is unchanged.

2.18.5.1. Model Definition and Scaling

- The model is a 1x1x1 in^3 cube. It is scaled to SI units 0.0254 meters on a side.
- Input pressure is 7 psi, multiplied by a frequency function. In SI units, this becomes $7 \times 6894.76 = 48263.32$ pascals, multiplied by the same function.
- Young's Modulus of 1e7 psi becomes 68.947573e9 pascals. Handbook value is 69 GPa.
- Density of 0.000259 slinch/in^3 (0.1000776 lbm/in^3) becomes 2770.138 kg/m^3. Handbook values of 2700 kg/m^3.

2.18.5.2. Results

The damage rate and stress must be independent of units. This is ensured by using the same comparison file for both. In addition, we have the following correspondence.

Result	English Units	SI Units	Status
Eigen Frequency	62846.1 Hz	62820.8 Hz	✓
max(Axrms)	2.7770 in/s^2	0.070537 m/s^2	✓
Vrms	113.84 psi	0.78492×10^6 Pa	✓
ZeroCrossingRate	60.965	60.965	✓
PeakFrequency	77.965	77.965	✓
NbDamageRate	2.2923E-13	2.2923E-13	✓
DamageRate	1.9324E-13	1.9324E-13	✓

2.18.6. *Fatigue Output of Dogbone Test*

Scope: Verification of **Sierra/SD** in the frequency domain builds upon a verification of "Siesta", a python post-processing tool for evaluation of high cycle fatigue damage. These solutions represent evaluation of the same damage quantities through a variety of means. Sierra/SD will evaluate the damage using frequency domain methods only.

Siesta has been evaluated using both the time domain and frequency domain. Computations were performed to individually compare both domains to an analytical solution for the simple case of a 5 Hz sine wave input. Two additional computations were conducted with the same model verifying that time domain and frequency domain both result in the same solution when provided more complex inputs. These evaluations were conducted on an element by element basis, and so some discrepancies to the single DOF analytical solution are expected.

Methodology: The dog-bone specimen described by Anes et al.[4] was chosen as a sufficiently simple model to solve damage analytically, with the additional benefit that experimentally derived results were available for our load case. Note that calculations were done using English units: IPS in Salinas, converted to Ksi during import into Siesta.

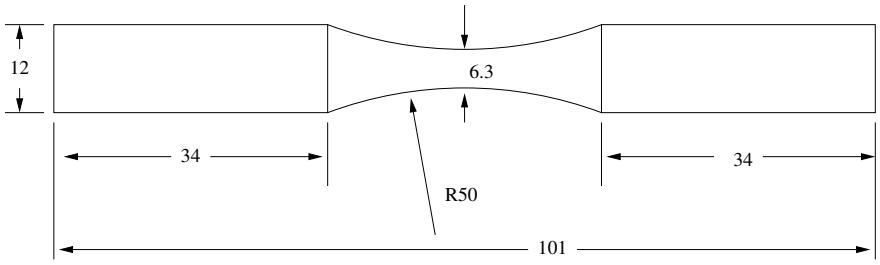


Figure 2-50. – Dog-bone Specimen Dimensions (mm).

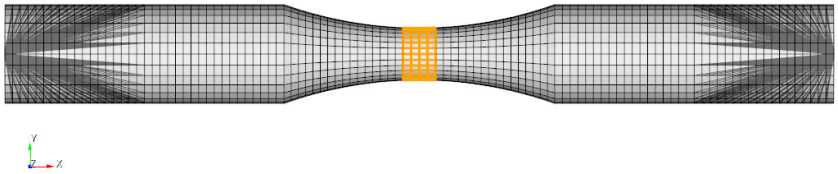


Figure 2-51. – Boundary conditions of mesh.

For all tests, the mesh is constrained via two points at either end of the specimen. Both points are fixed in all degrees of freedom except axial translation, and affixed to the mesh by rigid elements to the surfaces of the mesh near the ends. Results are output only for the narrow highlighted potion at the center of the model. Forces are applied at the end points with equal and opposite magnitudes. There are no point masses in the system; frequency domain input PSDs are truly provided as force squared per Hz.

To verify the results in both time domain and frequency domain, three test scenarios were evaluated, as illustrated by the PSDs shown in Figure 2-52, with details in Tables 2-13 through

2-15. The first was a 5 Hz fully reversed sine wave with 3141 lbf peak magnitude, the second was an example input matching a test specification with relatively narrow band frequency content, and the third was an example test specification with a wide band of frequency content. Note that the first elastic mode of the system occurs at 929 Hz, and modal random vibration solves included calculation of 150 modes to capture what is effectively a static solution at 5 Hz. Modes are computed to about 340 KHz.

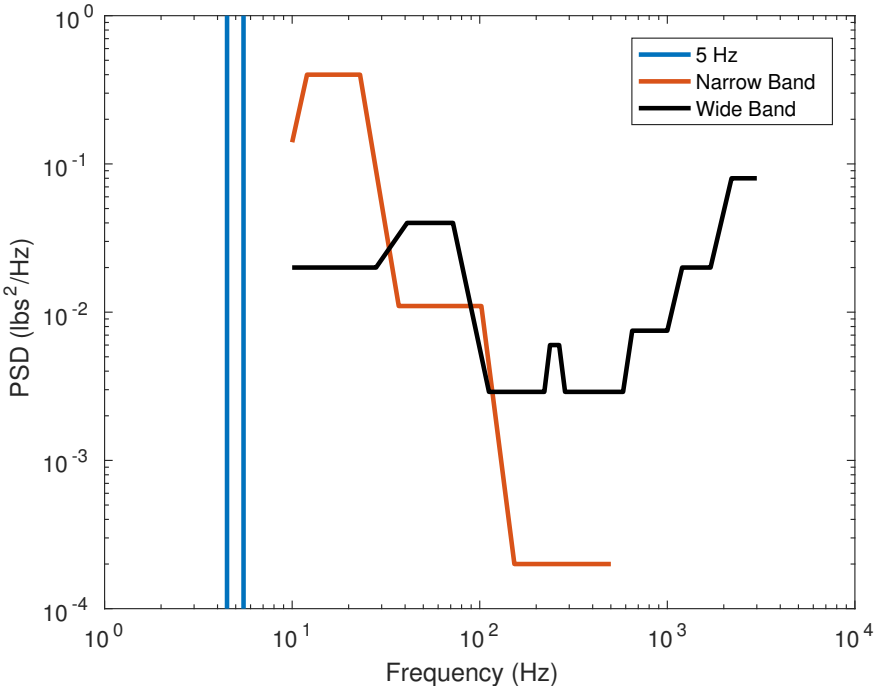


Figure 2-52. – Power Spectral Density of Input Force.

In the absence of an easy way to define a single-frequency PSD, the 5 Hz test was represented in the frequency domain using a PSD with the appropriate RMS magnitude, centered around 5 Hz, and with a band width of 1 Hz. Time domain realizations of the wide and narrow band test PSDs were generated such that their RMS values could not differ from the specification by more than 1 dB, the PSD of the generated signals could not differ by more than 6 dB at any frequency, and could not differ by more than 3 dB over 80% of the frequency range.

Frequency (Hz)	**PSD (lbs^2/Hz)**
4.00	1e-13
4.49	1e-13
4.50	4.93128e6
5.50	4.93128e6
5.51	1e-13
6.00	1e-13

Table 2-13. – 5 Hz PSD representation.

Frequency (Hz)	PSD (lbs^2/Hz)
10	0.1400
12	0.4000
23	0.4000
37	0.0110
102	0.0110
153	0.0002
500	0.0002

Table 2-14. – Narrow-Band PSD.

Frequency (Hz)	PSD (lbs^2/Hz)
10	0.0200
28	0.0200
41	0.0400
72	0.0400
112	0.0029
221	0.0029
237	0.0060
265	0.0060
285	0.0029
581	0.0029
650	0.0075
1000	0.0075
1200	0.0200
1700	0.0200
2200	0.0800
3000	0.0800

Table 2-15. – Wide-Band Force PSD.

Domain	Damage Model	Vrms (ksi)	Cycling Rate (Hz)	Damage Rate
Time	Minor's Rule	44.3-47.1	4.17	1.8E-6 - 4.2E-6
Frequency	Steinberg	42.9-46.3	5.01	3.0E-3 - 8.7E-3
	Narrow Band			9.4E-3 - 2.7E-2
	Wirsching-Light			4.8e-3 - 1.4e-2
Documentation	Experiment	46.36	5.00	5.0E-6

Table 2-16. – Preliminary 5 Hz Results.

Table 2-16 shows the preliminary results of the 5 Hz test of frequency and time domains. The ranges indicate spatial changes. Sierra/SD and Siesta results are very close for this model. However, it is important to note that neither domain's damage formulations are intended to be used on a sine input. Because this is a sine input, three adjustments must be made to the raw data.

1. The rainflow algorithm consistently misses one half cycle on the input, and interprets a 0.6 second 5 Hz tone as a 4.17 Hz tone instead. As the time history increases, the recorded cycling rate converges to 5 Hz, so we will act as though it detected 5 Hz. It is recommended that you use the longest time history feasible, preferably 50-100 cycles of the lowest frequency.
2. Narrow band damage, and Wirsching-Light by extension, includes a scale factor of $\Gamma(1 + m/2)$ on the damage, where Γ is the gamma function, and m is the fatigue exponent. For a sine input, this is not appropriate, as it makes the calculated damage wildly conservative, so we will reduce the damage by this same factor.
3. The Steinberg method for calculating damage includes the assumption that the magnitude of Vrms is a one sigma event, and adjusts the damage to reflect the influence of 2-sigma and 3-sigma events as well. These cycles do the majority of the damage on a system, and so this approach is not appropriate for modeling a strictly controlled experiment with 100% of the cycles at the same value.

After adjusting the results and removing Steinberg from the chart, we are left with Table 2-17. It is worth noting that the Wirsching-Light damage metric is intended to compensate for conservatism on wide-band signals; as this signal is very narrowband, the correction is unnecessary. In summary, the narrow band results are as expected.

✓ The preliminary results for Siesta and Sierra/SD agree very well.

✓ With appropriate corrections, these results are consistent with both rainfall computations and with experiment.

Domain	Damage Model	Vrms (ksi)	Cycling Rate (Hz)	Damage Rate
Time	Minor's Rule	44.3-47.1	5	2.2E-6 - 5.0E-6
Frequency	Narrow Band	42.9-46.3	5.01	1.7E-6 - 5.0E-6
	Wirsching			0.9E-6 - 2.5E-6
Anes	Experiment	46.36	5.00	5.0E-6

Table 2-17. – 5 Hz test after adjustments. Ranges indicate spatial variation.

2.18.6.1. Narrowband and Wide-Band Evaluation

Tables 2-18 and 2-19 show the results under representative wide and narrow-band PSD inputs. Narrow band damage represents the time domain solution well, and is strictly conservative in our selected band of elements, but the wide band test revealed that the frequency domain is only an estimate of damage expected from the time domain analysis under wide-band loading. Why this

discrepancy exists is not well understood, but may be caused by the shape of the wide-band PSD used. It may be possible to better represent the wide-band test with 2-3 narrow band tests under the order-independent assumption of Minor's rule, but this was not tested.

As with the 5 Hz test, the rainflow algorithm used in the time domain calculated an inaccurate cycling frequency when provided with a narrow-band signal. This is not considered to be a problem because the overall damage appears to be well accounted for.

Domain	Damage Model	Vrms (ksi)	Cycling Rate (Hz)	Damage Rate
Time	Minor's Rule	0.046-0.049	532	5.6E-43 - 1.7E-42
Frequency	Steinberg	0.051-0.055	47	2.0E-43 - 5.8E-43
	Narrow Band			6.3E-43 - 1.8E-42
	Wirsching			2.9E-43 - 8.4E-43

Table 2-18. – Narrow-Band Test Results. Ranges indicate spatial variation.

Domain	Damage Model	Vrms (ksi)	Cycling Rate (Hz)	Damage Rate
Time	Minor's Rule	0.22 - 0.24	2486 - 2487	21E-33 - 56E-33
Siesta results				
Frequency	Steinberg	0.20 - 0.22	2293	2.2E-33 - 6.3E-33
	Narrow Band			6.8E-33 - 20E-33
	Wirsching			3.1E-33 - 9.0E-33
Sierra/SD results				
Frequency	Narrow Band	0.201 - 0.217	2293	6.5E-33 - 19E-33
	Wirsching			3.0E-33 - 8.6E-33

Table 2-19. – Wide-Band Test Results. Ranges indicate spatial variation.

Narrow-band and Wide-band results are very similar for Siesta and Sierra/SD, but they are not identical. We expect that there are round off errors and integration differences leading to those differences. These are particularly difficult in an undamped system with numerical integration crossing peak resonance. Table 2-20 compares these results. Overall, the comparison is good, and well within the differences of the other methods.

Parameter	Narrow-Band			Wide-Band		
	Siesta	Sierra	Diff	Siesta	Sierra	Diff
Vrms (psi)	55	55.45	0	220	220	0%
ν_o^+	47	47	0	2293	2293	0%
NB Damage	1.8e-42	1.74-e42	3%	2.0e-32	1.9E-32	5%

Table 2-20. – Maximum of Siesta and Sierra/SD Computations.

2.18.6.2. Integration and Damping

The PSD spectrum is integrated through frequency to determine the RMS stress and the stress moments. For undamped systems, that function is singular at the resonance points. Two factors influence the accuracy of that solution. First, damping removes the singularity in the solution. Second, the size of the frequency step addresses the accuracy of the integral.

Figure 2-53 provides some information on the convergence of the solution as these parameters are varied. The variation of the narrowband damage, D_{NB}, versus damping is shown on the left. For damping below 1%, there is no significant impact on the solution. The graphic on the right illustrates the same data, sliced another way. We observe that the frequency step, ΔF, has a significant effect on the solution. For our problem, independent of damping, the frequency step should be below 1 Hz. However, with no damping and a small frequency step, very different (non-convergent) results are obtained. This is consistent with numerical integration across a singularity. For input deck see Appendix 9.25.

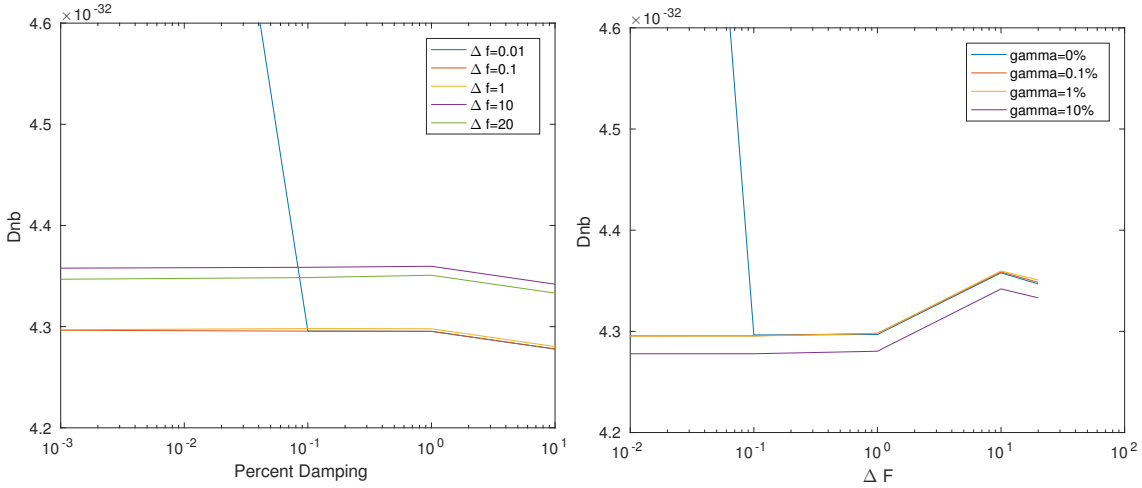


Figure 2-53. – Convergence of PSD Integration.

2.18.7. Fatigue Output of Pinned Shell

A narrowband and wideband example are tested to ensure,

- The stress is evaluated at all three surfaces (top, middle, bottom), and the *larger* of these values is used for evaluation of damage.
- The zero crossing and peak frequency make sense in the context of the PSD input. This is easier to evaluate for narrowband processes.
- Von Mises stress is consistent between `modalranvib` and `FRF` solutions.
- The von Mises stress is consistent with a static solution.
- Damage Rate is consistent with independent MATLAB calculations.

We do not have a comparison with time domain rain fall calculations. We also have no convergence study, either with mesh, or with modes.

2.18.8. Narrow Band Pinned Plate

The model is a simple rectangular plate, shown in Figure 2-54. The plate is 10 units in X, 1 unit in

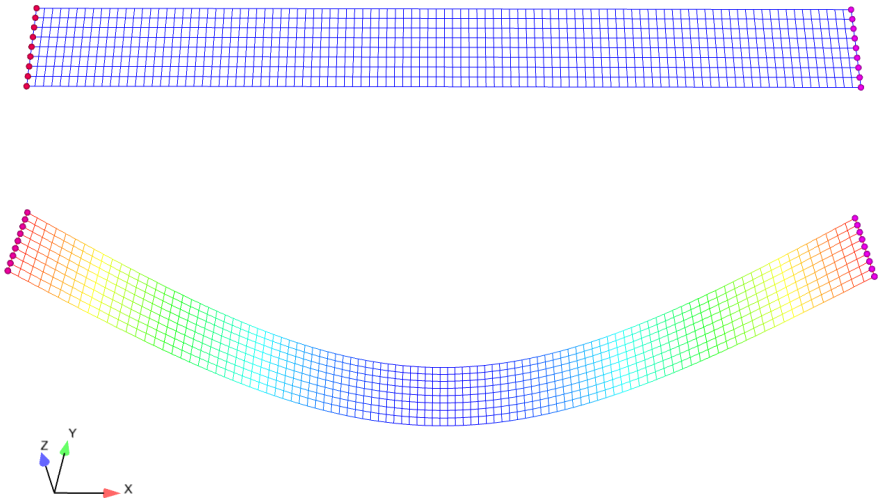


Figure 2-54. – Pinned Plate Geometry, and First Mode.

Y, 0.01 units thick, and all deformation is in the Z direction. In modal analysis, only the first mode is retained, which is a bending mode, shown in the lower portion of Figure 2-54. The +/- X surfaces are pinned, with no other Dirichlet boundary conditions. Loading is a uniform pressure in the $-Z$ direction. The narrowband loading is shown in Figure 2-55, where the entire loading is in the 4 Hz to 5 Hz range. The first mode is at about 8.9 Hz, so this loading is below that first mode.

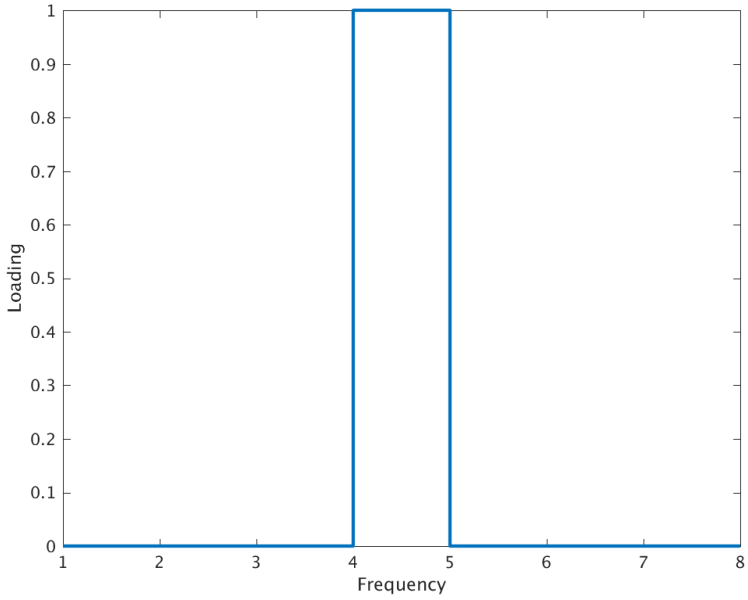


Figure 2-55. – Pinned Plate. Random Vibration Loading.

2.18.8.1. Statics:

The static response on the bottom surface, to a uniform pressure load is shown in Figure 2-56. Stress on the top surface is the negative of this, and there is no stress on the midplane.

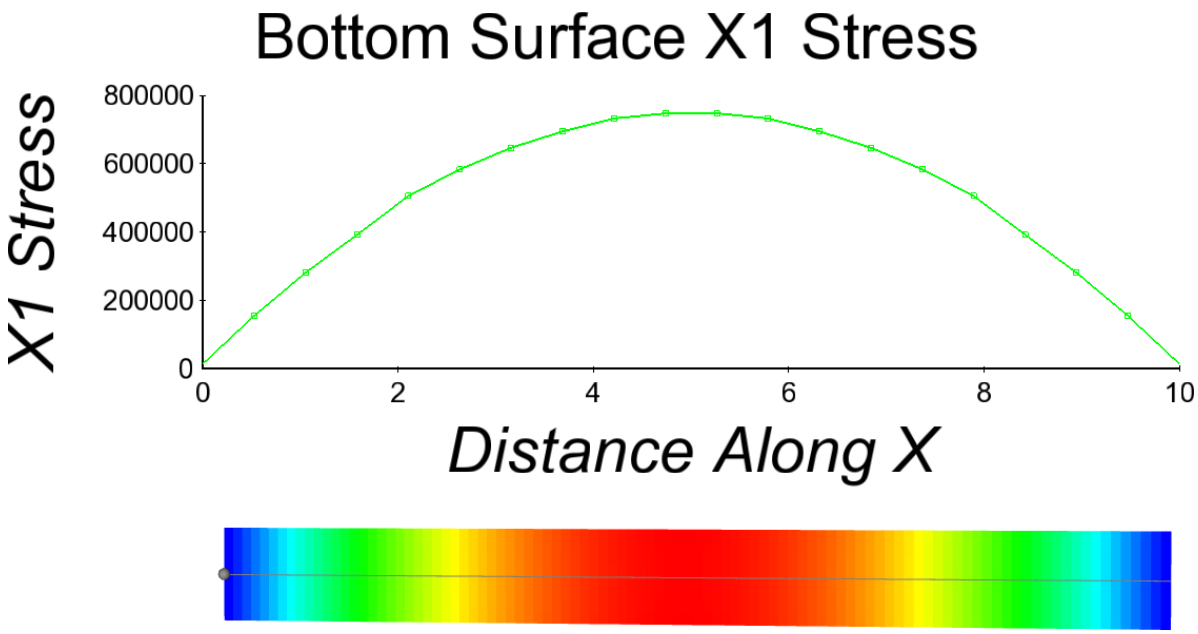


Figure 2-56. – Pinned Plate. Statics Response.

2.18.8.2. FRF:

The input is modified, and a modal FRF computed from 0.01 to 8 Hz, as shown in Figure 2-57. The stress response is very similar to the static solution, as evidenced in Figure 2-58. There are expected deviations, as the FRF response includes only a single mode. However, the stresses are as expected, and they increase at the sample frequency of 4.55 Hz, as the solution approaches resonance.

✓ FRF and Statics displacements and stresses are consistent.

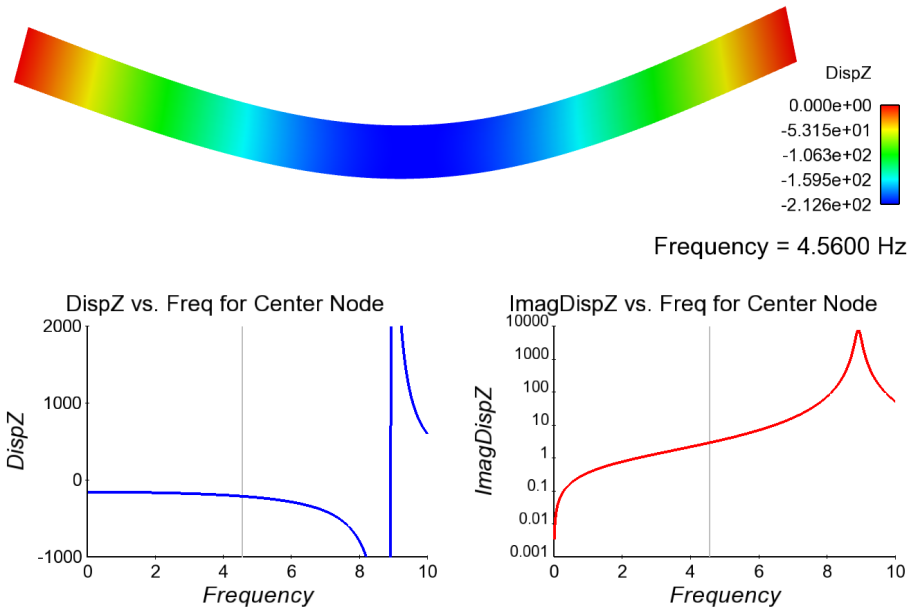


Figure 2-57. – Pinned Plate. Modal FRF Response.

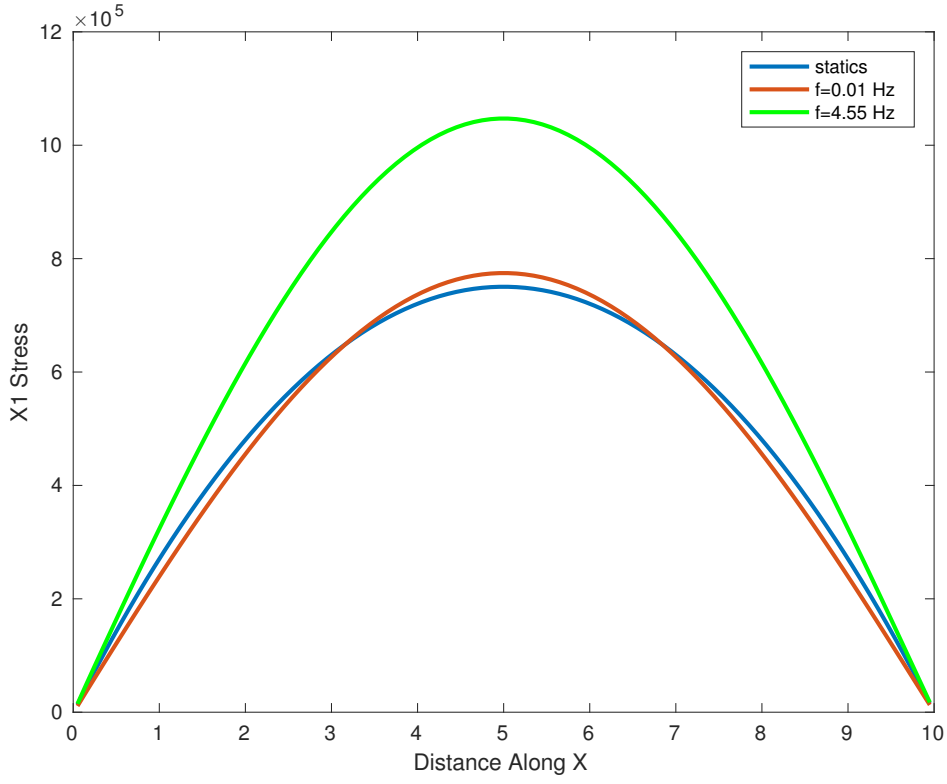


Figure 2-58. – Pinned Plate. Comparison of Static and FRF Solutions.

2.18.8.3. Random Vibration Analysis:

The next step of the computation is evaluation of the RMS von Mises stress through the modal random vibration analysis. The peak value of this stress is 1.037×10^6, which occurs in the center of the structure. This value is consistent with the stresses computed in the FRF and Statics portions of the analysis. The following are confirmed.

- ✓ The zero crossing and peak frequency, determined from Vrms_i, are both about 4.5 Hz, consistent with the narrow band sweep in this analysis.
- ✓ RMS stresses are consistent with the FRF values. Note however, that these are all axial stresses.

2.18.8.4. Fatigue Damage Analysis:

The final step is the fatigue analysis. Output of this analysis confirms,

- ✓ The zero crossing and peak frequency are correct.
- ✓ Damage rates are consistent with hand calculations.

$$D_{NB} = \frac{\nu_o^+}{A}(\sqrt{2}\sigma_s F_{SS})^m \Gamma\left(\frac{m}{2}+1\right)$$

For our structure, $D_{NB} \approx 5$ in the center of the plate.

$$\begin{aligned}
A_1 &= 12.1689 \\
A &= 10^{A_1} = 1.475\times10^{12} \\
m &= 3 \\
F_{SS} &= 0.0001 \\
\nu_o^+ &= 4.534 \\
\Gamma(5/2) &= 1.3293 \\
\sigma &\approx 1.0377\times10^6 \\
D_{NB} &= 1.2911\times10^{-5} \qquad \text{(from hand calculations)}
\end{aligned}$$

The value from the output is $D_{NB} = 1.291125933\times10^{-5}$.

2.18.9. Wideband Calculations

Wideband calculations use the same model as narrowband. Only a single mode is retained as shown in Figure 2-54, however the band selected is from 10 -100 Hz. Figure 2-59 shows the displacement response over this band, with a 1% damping. Above the 8.9 Hz mode, the response rolls off.

For this model, the zero crossing rate at all locations is $\nu_o^+ = 12.351$. The peak frequency is somewhat higher (as expected), at $\nu_p = 20.115$ Hz. Both reflect the much higher energy at lower frequency because the dominant mode is at 8.9 Hz.

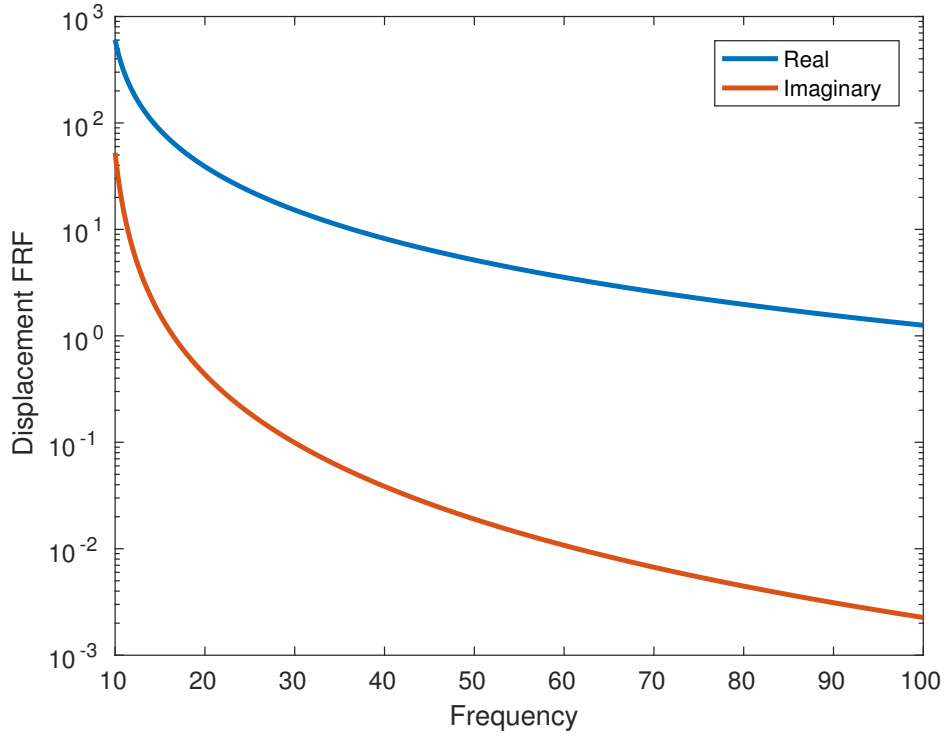


Figure 2-59. – Pinned Plate. Wide Band FRF Response.

✓ The zero crossing and peak frequency are reasonable.

The peak Damage occurs in the middle of the plate. Peak values for `NbDamageRate` and `DamageRate` are 6.8259×10^{-4} and 5.6715×10^{-4} respectively.

The RMS von Mises stress can be computed in two ways. First, the modal random vibration method can be used. Second, a frequency response method is used. Each method is applied here for the element of maximum stress, in the center, with id 51.

ModalRanVib: This method, described in the Sierra/SD manuals, computes the RMS von Mises stress. The value from the method is $\hat{V}_{RMS} = 2.7886 \times 10^6$.

FRF: This method uses the transfer functions. From the output of the modalFRF calculation,

$$\bar{V}^2_{RMS} = \int_0^\infty H^\dagger(\omega) S_{FF}(\omega) H(\omega) d\omega$$

where $H(\omega)$ is a stress transfer function, and S_{FF} is the force input power spectral density. For element 51, $\bar{V}_{RMS} = 2.8695 \times 10^6$. Here we assume that the stress is uniaxial, and H applies to σ_{xx}, the axial portion of the stress. The MATLAB code to approximate this integral is,

```
h1  =  evar23(51,:) + sqrt(-1)*evar01(51,:);
h1  = h1.';
df  =  0.1;
Sff  = 1;
```

```
Vrms2 =  h1'*Sff*h1*df;
Vrms  = sqrt(Vrms2)
```

The difference between these two values is about 3 percent. That would appear to be too large. However, evaluation of convergence as the frequency step is decreased indicates much less error in the modal random vibration solution. See Figure 2-60. The RMS stress depends on damping. Setting the damping ratio to 50%, results in stresses of 1.2721e6 and 1.262251e6 using an FRF and random vibration method respectively. As expected, the integration error is lower for these values, and relative error is about 0.8%.

✓ Computation of the RMS stress is consistent between the two methods.

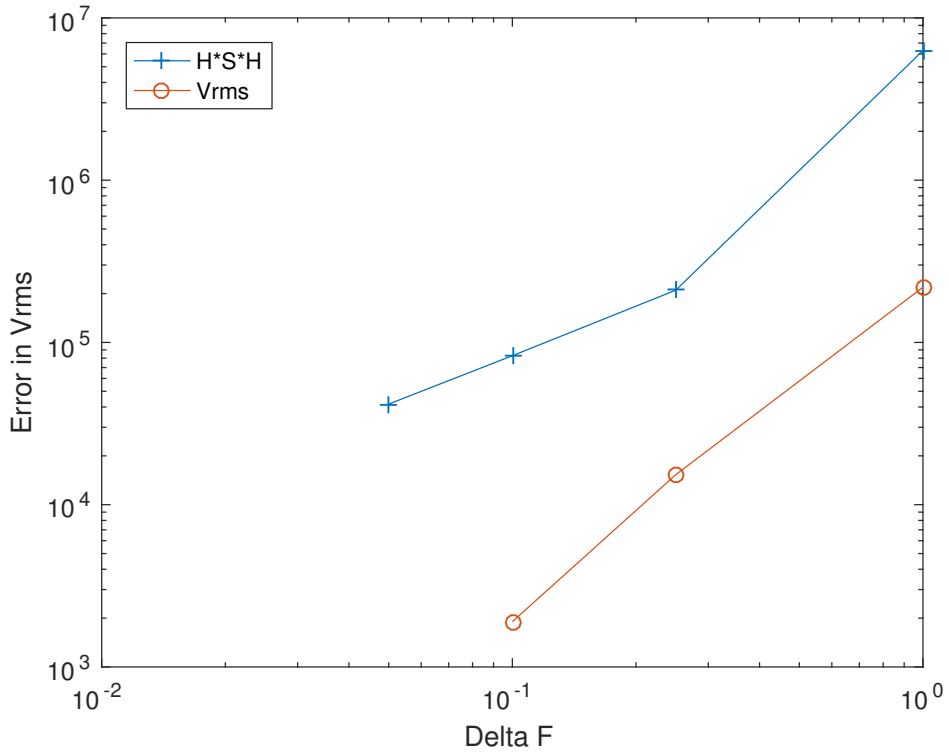


Figure 2-60. – Convergence of Frequency Integrals.

For input deck see Appendix 9.26.

2.18.10. Nodal Loading vs Sideset Loading for Modal Random Vibration

Modal random vibration verification test of the flat plate of hexshells is shown in Figure 2-61 for pressure loading on the top surface. The plate is fixed at nodeset 3 and nodeset 4 and is fixed in the z direction at nodeset 1 and 2.

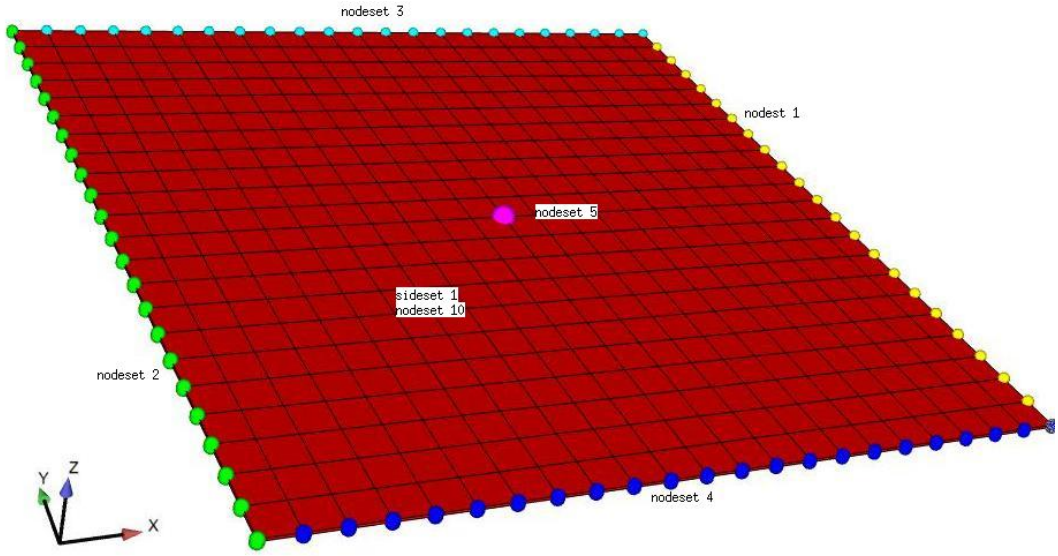


Figure 2-61. – Schematic of flat plate geometry with nodesets and sidesets labeled. A pressure load is applied to the top surface on sideset 1 and a force load is applied to the top surface on nodeset 10. Frequency output shown in Figure2-62 is taken at nodeset 5.

A pressure loading is compared to three other loading scenarios for verification. The four solutions to this problem are presented in Figure 2-62 for measurements taken at the center node of the plate, nodeset 5 in Figure 2-61.

These four solutions are:

1) Modalranvib with pressure applied to the top surface of the plate, sideset 1 in Figure 2-61. Results are shown in blue in Figure 2-62.

2) Modalranvib with nodal forces applied to the entire top surface, nodeset 10 in Figure 2-61. The load is scaled to be equivalent to the pressure load in case 1. Note that the exodus mesh for nodeset 10 has a distribution factor of 0.5 so the force load is scaled by 2 in the input file. Results are shown in red in Figure 2-62. Note that there are no noticeable differences for cases 1 and 2.

3) A time history solution post-processed to give the power spectral density shown in red.

4) NASTRAN solution for modalranvib shown in green.

All the input files and MATLAB scripts required to run these simulations are in the test directory.

The Salinas results for pressure loaded modalranvib are the same as the other solution methods.

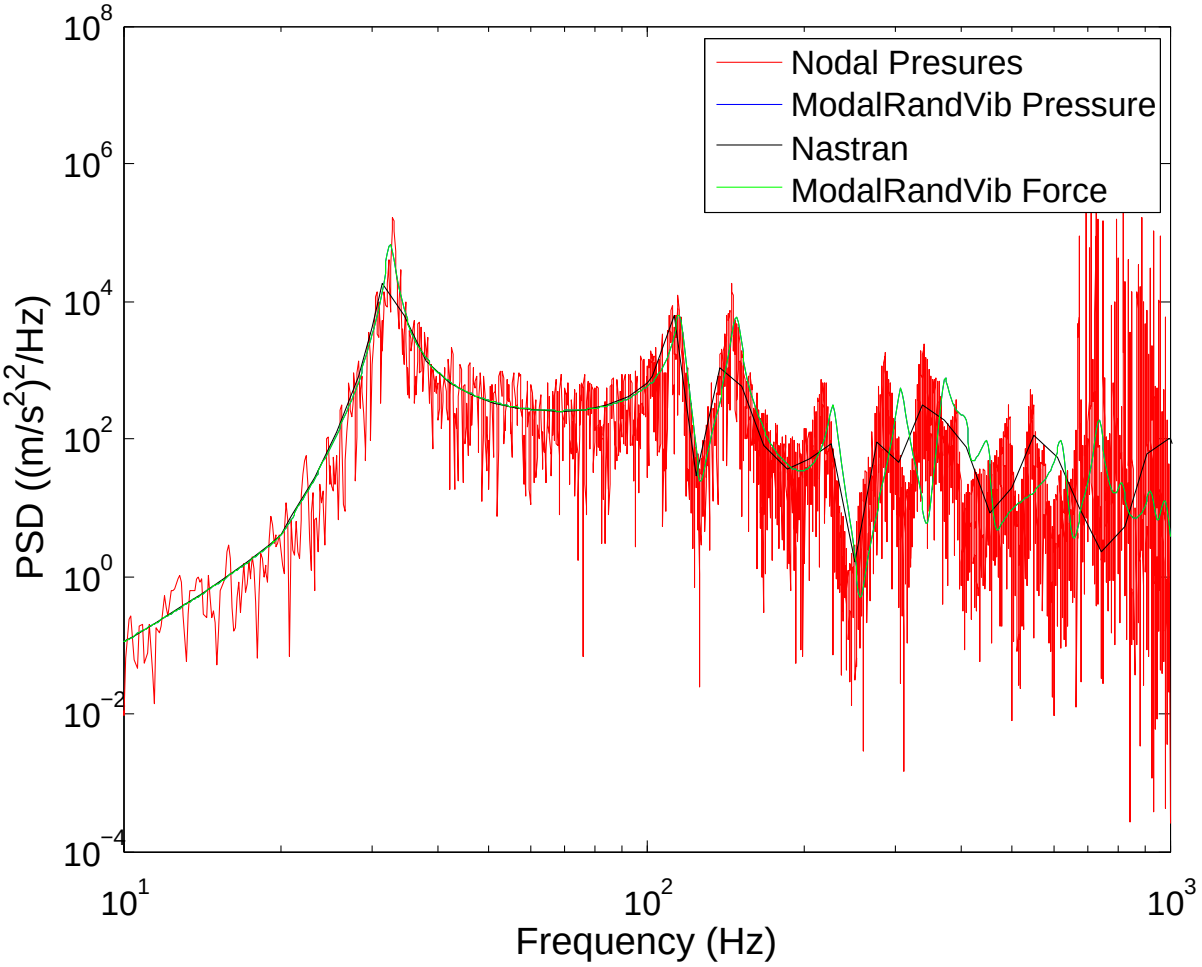


Figure 2-62. – PSD Comparison of modalranvib, nodal transient, and NASTRAN modalranvib.

For input deck see Appendix 9.27.

2.19. Coupled Electro-Mechanical Physics

2.19.1. Static Response for Electric Field Induced Beam Deformation

In this section, we verify the electro-mechanical coupling in the stiffness matrix using the static response of two bimorph beam models (2-63) that were presented in work by X.D. Zhang and C.T. Sun [55]. The first test verifies the part of the stiffness matrix that couples the electric field to transverse strain, and the second verifies the coupling of the electric field to shear strain. To verify, we compared the transverse displacements generated from **Sierra/SD** with the analytic solutions derived in the referenced paper [55].

2.19.2. Bimorph Beam in Bending

The first model is an aluminum cantilevered beam pressed between two piezoelectric strips (Figure 2-63a). The piezoelectric strips are PZT5H and are polarized in the direction parallel to their thickness (z-axis). The constitutive properties of the piezoelectric and aluminum materials are presented in Table 2-21. Ten volt voltages ($V_{in} = V_{out} = 10$) are prescribed to the outermost surfaces parallel to the length of the beam and the aluminum core is grounded ($V_g = 0$). Figure 2-64 presents the analytic and **Sierra/SD** generated transverse displacements over the length of the beam.

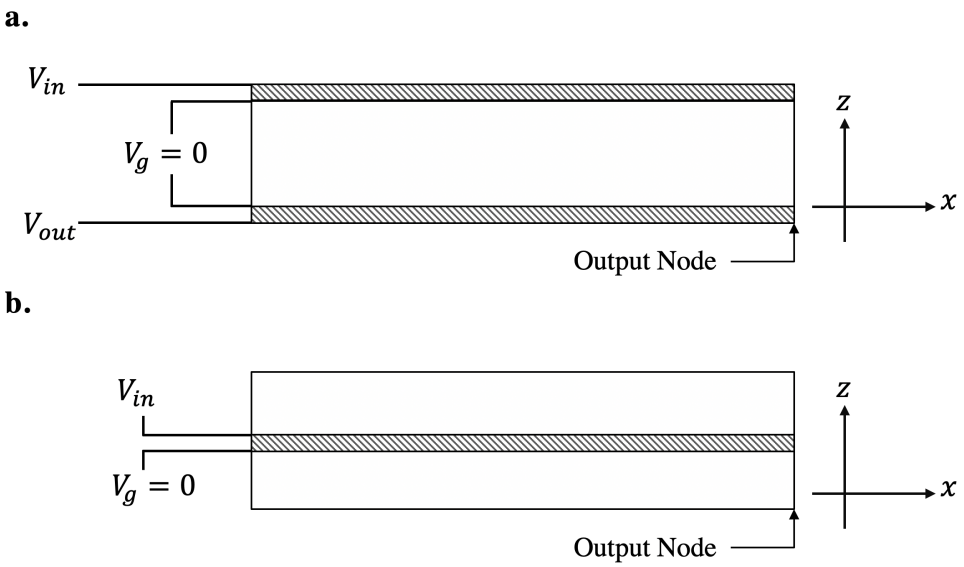


Figure 2-63. – Cantilevered bimorph beams with piezoelectric layer (hatch) and aluminum layer (solid). Model (a) verifies the electric field transverse strain coupling, and model (b) verifies the electric field shear strain coupling.

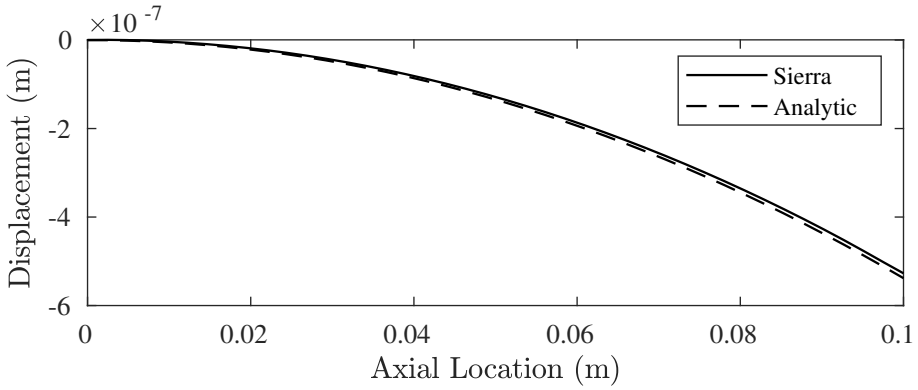


Figure 2-64. – The FE and analytic transverse displacements along the length of the bimorph beam from Figure 2-63.

Table 2-21. – Material Properties for PZT5H [55].

PZT5H								Al	
GPa					C/m^2			GPa	
c_{11}	c_{12}	c_{13}	c_{33}	c_{44}	e_{31}	e_{33}	e_{15}	E	ν
126	79.5	84.1	117	23	-6.5	23.3	17	70.3	0.345

2.19.3. Sheared Bimorph Beam

The second model is a piezoelectric cantilevered beam pressed between two aluminum strips (Figure 2-63b). The piezoelectric material is PZT5H and it is polarized in the direction parallel to the length of the beam (x-axis). A twenty volt voltage (V_{in} = 20) is prescribed to the upper interface between the aluminum and piezoelectric strip, while the lower interface is grounded (V_g = 0). Figure 2-65 presents the analytic solution superimposed over the transverse displacements generated from **Sierra/SD** over the length of the beam.

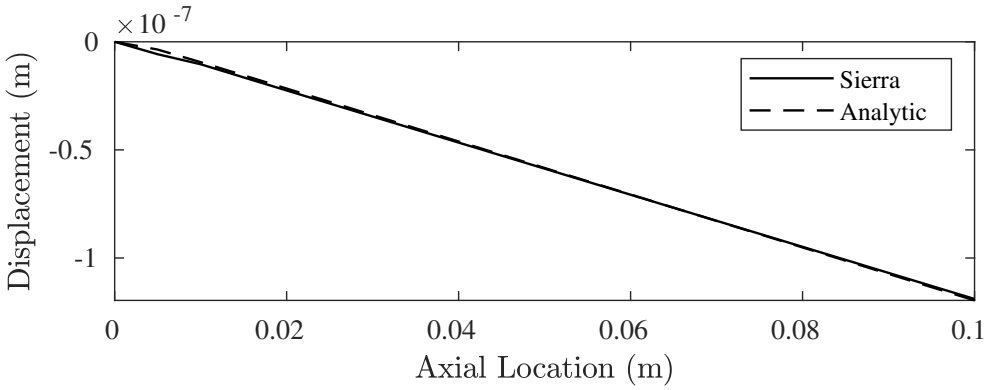


Figure 2-65. – The FE and analytic transverse displacements along the length of the shear beam from Figure 2-63.

2.19.4. Transient Response for Electric Field Induced Beam Deformation

In this section, we verify the transient solution method for a piezoelectric model subject to prescribed time-varying voltage boundary conditions. The details of the bimorph beam are provided in Figure 2-63a and Section 2.19.1. A constant voltage of 20 volts (V_{in} = 20) is prescribed to the top surface, and the aluminum core is grounded. An equipotential surface is enforced at the bottom surface (V_{out}) with a voltage rigid set. In other words, the voltage is spatially constant along the equipotential surface. The piezoelectric material is isotropic in permittivity where its permittivity is set to the permittivity of free space.

To verify the transient solution method, we performed a transient solve using an identical model with the FE software COMSOL [14] and compared its generated time-histories with those generated from **Sierra/SD**. Figure 2-66 presents the displacement time-history of the output node (see Figure 2-63a) in the transverse direction (z-axis) and the axial direction (x-axis). Figure 2-67 presents the voltage time-history at (V_{out}). We observe excellent agreement between **Sierra/SD** and COMSOL.

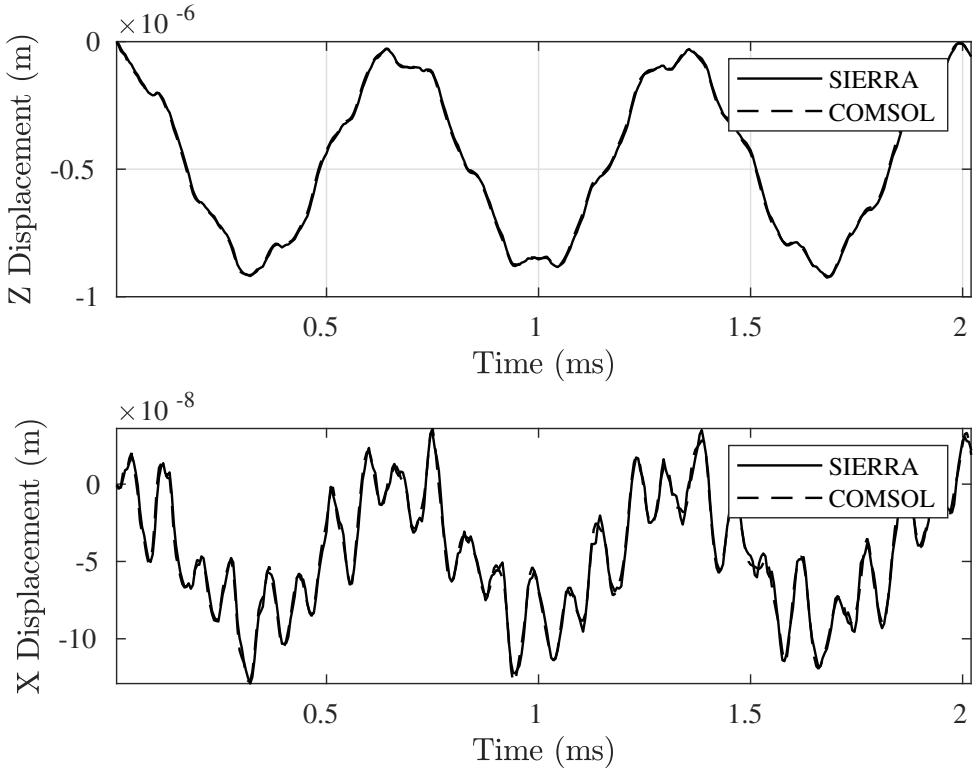


Figure 2-66. – Time-histories of transverse (z) and axial (x) displacements generated from COMSOL (dashed) and **Sierra/SD** (solid).

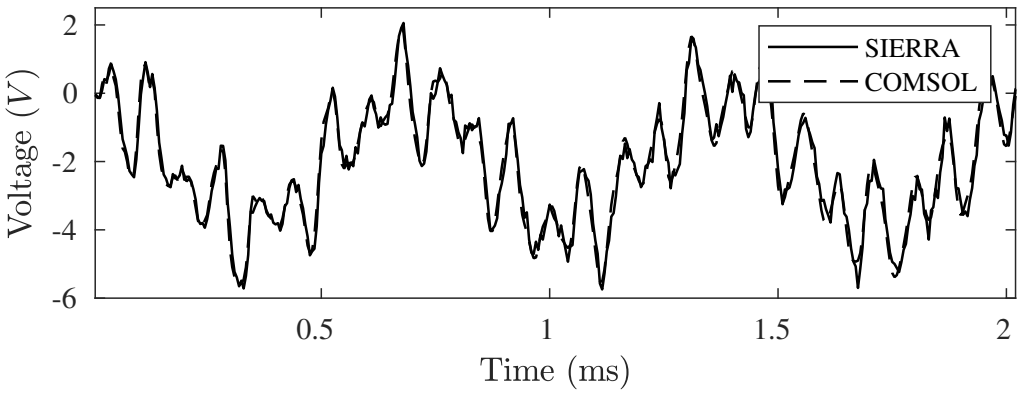


Figure 2-67. – Voltage time-history of V_{out} generated from COMSOL (dashed) and **Sierra/SD** (solid).

2.19.5. Frequency Response for Electric Field Induced Beam Deformation

The frequency response of the bimorph beam subjected to a prescribed frequency dependent voltage boundary condition is verified. The details of the bimorph beam are provided in Figure 2-63a and Section 2.19.1. In this example, a voltage of 20 volts is prescribed to the top surface of the beam ($V_{in} = 20$) and the aluminum core is grounded. Like the transient example, we enforce an equipotential surface at the bottom surface of the beam (V_{out}) using a voltage rigid set. Stiffness proportional damping, with coefficient $\beta = 8e - 7$, is prescribed to the piezoelectric blocks. The aluminum core is undamped.

To verify the direct frequency response solution method, we performed a frequency domain solve using an identical model with the FE software COMSOL [14] and compared its solution with the solution generated from **Sierra/SD**. Figure 2-68 presents the response amplitudes of the output node (see Figure 2-63a) for the following: 1) the transverse direction (z-axis), 2) the axial direction (x-axis), and 3) the voltage at V_{out}. We observe excellent agreement between **Sierra/SD** and COMSOL.

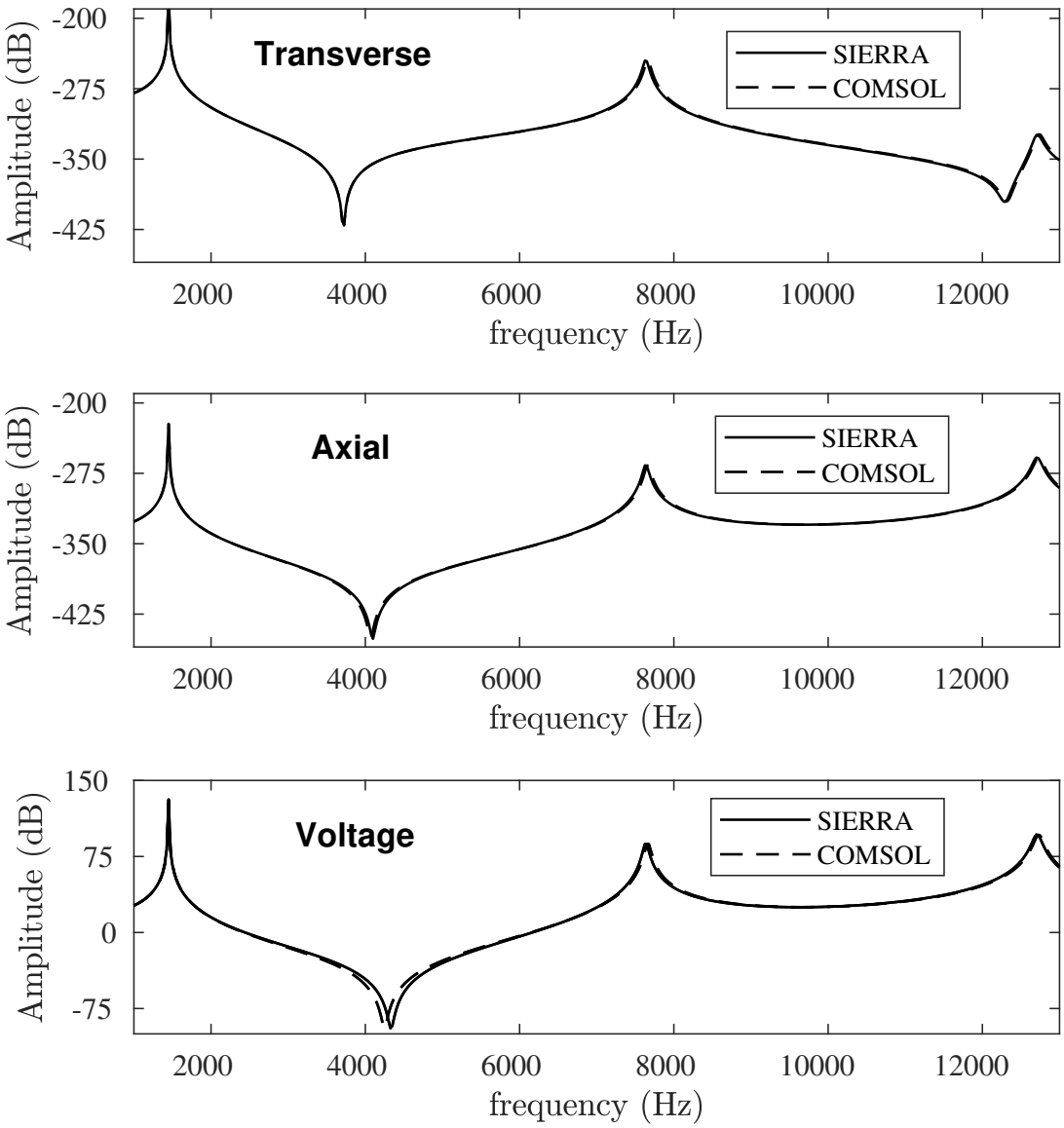


Figure 2-68. – The frequency response amplitudes generated from **Sierra/SD** and COMSOL for 1) the transverse direction (z-axis), 2) the axial direction (x-axis), and 3) the voltage at V_{out}.

2.19.6. Eigenvalue Verification of a PZT5A Disc

The eigenvalues of coupled electro-mechanical physics are verified based on a published example [25]. The specified problem is a PZT5A disc with D/T ratio of 20. The properties used are in Table 2-22. The disc is 40.10mm in diameter and 2.03mm thick, giving it a D/T ratio of 20, which is the same ratio as many transducers. The piezoelectric material is polarized in the Z (or 3) axis.

Table 2-22. – Properties of PZT-5A [25].

Property	Units	PZT-5A
ε_0	F/m	8.854×10^{-12}
$\varepsilon^S_{11}/\varepsilon_0$		916
$\varepsilon^S_{33}/\varepsilon_0$		830
c^E_{11}	$10^{10}N/m^2$	12.1
c^E_{33}	$10^{10}N/m^2$	11.1
c^E_{12}	$10^{10}N/m^2$	7.54
c^E_{13}	$10^{10}N/m^2$	7.52
c^E_{44}	$10^{10}N/m^2$	2.11
c^E_{66}	$10^{10}N/m^2$	2.26
e_{31}	C/m^2	-5.4
e_{33}	C/m^2	15.8
e_{15}	C/m^2	12.3
ρ	$10^3 kg/m^3$	7.75

Thus, the elasticity matrix:

$$C^E = 10^{10} \times \begin{bmatrix} 12.1 & 7.54 & 7.52 & 0 & 0 & 0 \\ 7.54 & 12.1 & 7.52 & 0 & 0 & 0 \\ 7.52 & 7.52 & 11.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.26 \end{bmatrix} N/m^2 \quad (2.19.1)$$

the dielectric matrix:

$$\varepsilon^S = 8.854 \times 10^{-12} \times \begin{bmatrix} 916 & 0 & 0 \\ 0 & 916 & 0 \\ 0 & 0 & 830 \end{bmatrix} F/m \quad (2.19.2)$$

and the piezoelectric coupling matrix:

$$e = \begin{bmatrix} 0 & 0 & 0 & 0 & 12.3 & 0 \\ 0 & 0 & 0 & 12.3 & 0 & 0 \\ -5.4 & -5.4 & 15.8 & 0 & 0 & 0 \end{bmatrix} C/m^2 \quad (2.19.3)$$

Figure 2-69. – PZT5A disc verification problem.

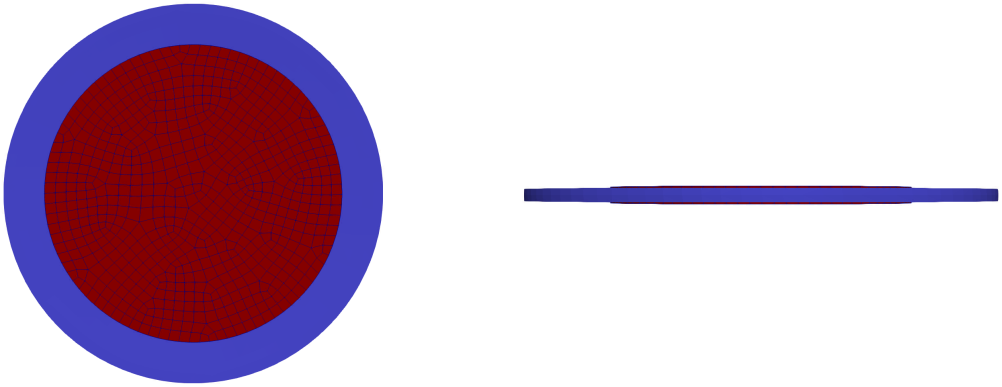


Figure 2-70. – First Radial Model of PZT-Disk. Literature 49.56 kHz, Sierra/SD 49.603 kHz. Red shows undeformed disk, blue shows radial extension mode shape.

The referenced paper [25] uses radially symmetric elements, thus only the radially symmetric modes are presented. Additionally, the authors claim that the bending modes can not be used to excite the mechanical system, so only the radial extension modes are presented [25] . **Sierra/SD** calculated all modes, including radially symmetric bending modes, as well as non-symmetric modes. In the referenced paper[25], the first two radial extensional modes occur at 49.56 kHz and 128.1 kHz. From the **Sierra/SD** runs, the first radial mode is represented by the 35th eigenpair, and has a natural frequency of 49.603 kHz. The second radial mode is represented by the 104th eigenpair, and has a natural frequency of 128.757 kHz. When the material is modeled as a purely elastic orthotropic material, with no consideration of the piezoelectric effect, these modes also appear at the same frequencies. In the pure elastic-orthotropic case, other modes change frequency, ordering and numbering, but the radial modes remain at the same frequency. Figure 2-70 shows the first radial extension mode calculated in **Sierra/SD**, and Figure 2-71 shows the second radial extension mode calculated in **Sierra/SD**.

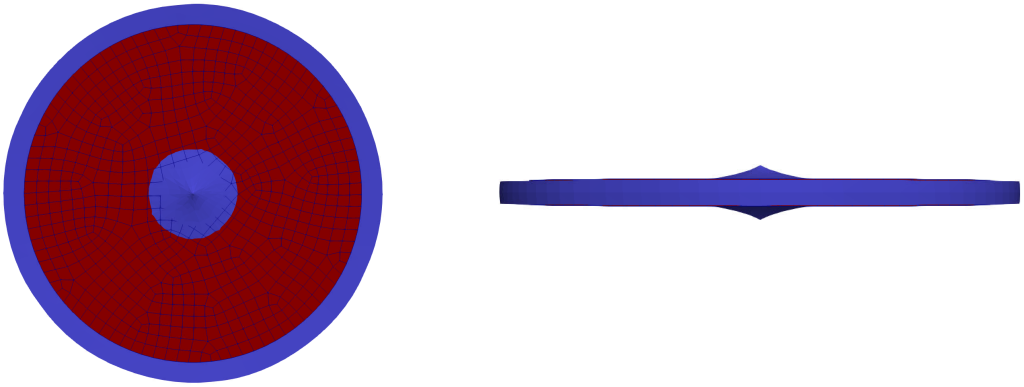


Figure 2-71. – First Radial Model of PZT-Disk. Literature 128.1 kHz, Sierra/SD 128.757 kHz. Red shows undeformed disk, blue shows radial extension mode shape.

2.20. Sierra/SM to Sierra/SD Coupling

2.20.1. Deflection of Axially Loaded Beam

This verification test computes deflections of beams with axial preload. Comparisons are made between analytic solutions, nonlinear static **Sierra/SM** analysis, and linear static **Sierra/SD** analysis with geometric stiffness from preload.

An idealized model, see Figure 2-72, has the geometry shown in Figure 2-73. A uniform elastic material is colored red, and green dots represent nodes at which boundary conditions are applied. The left node is fixed in x and y to represent a pin, and the right node is fixed in y and has the applied axial force P, and the whole model is fixed in z. A very stiff material, colored gray, is used on the beam ends to prevent large local deformations around the pinned nodes. The small yellow sideset at the center of the beam is used for applying a traction to generate the applied lateral force F.

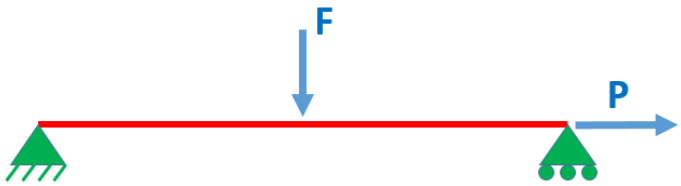


Figure 2-72. – Idealized Model Setup.

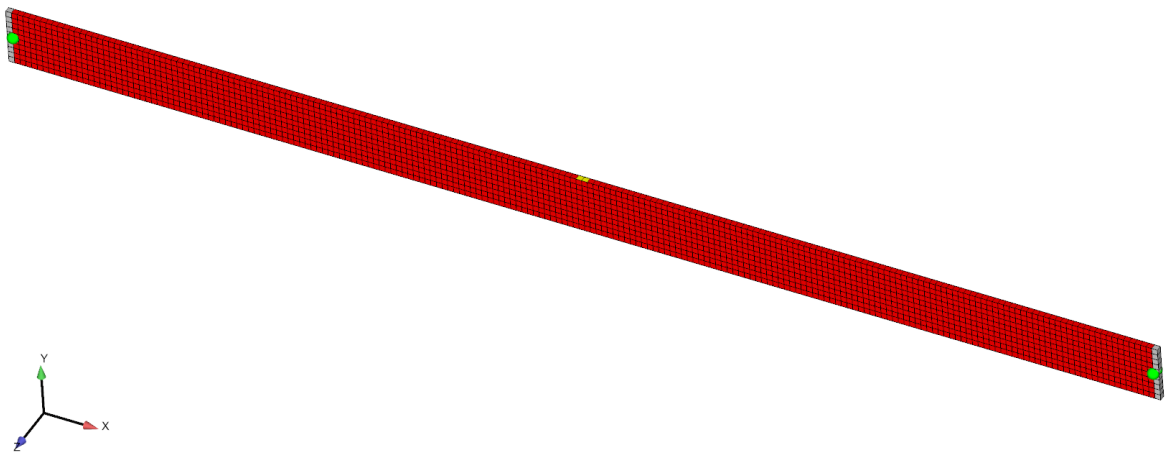


Figure 2-73. – Meshed Beam.

Verification checks of the model vs. analytic solutions are made looking at the maximum lateral deformation δ under various conditions. This maximum lateral deformation always occurs at the beam mid-span. Deflections are kept small to approximate the linearized simple beam equations. The specific geometry and loads considered are given in Table 2-23. The test directory contains a Mathematica notebook file that gives the beam analytic solutions and detailed numeric results for the specific geometry and loads considered.

Table 2-23. – Model Parameters.

Parameter	Value	Description
L	20	Beam Length in X direction
H	1	Beam Height in Y direction
W	0.1	Beam Width in Z direction
E	7.8e+7	Material Young's Modulus
F	2000	Lateral Force
P	-5000, 0, or 5000	Axial Force

2.20.1.1. Basic Beam Deflection

The beam moment of inertial I is given by Equation 2.20.1.

$$I = \frac{WH^3}{12} \tag{2.20.1}$$

In absence of an axial load, the expected beam deflection δ_{bend} is given by Equation 2.20.2.

$$\delta_{bend} = \frac{FL^3}{48EI} \tag{2.20.2}$$

The expected numeric result for the geometry is $\delta_{bend} = 0.512$. **Sierra/SM** computes a value of 0.521 and **Sierra/SD** a value of 0.516. The discrepancy is due to limited mesh resolution and small deviations between idealized Euler-Bernoulli beam theory and the 3d model. Generally, the comparison is good, indicating that the finite element model closely aligns with the beam theory assumptions.

2.20.1.2. Beam Deflection with Axial Preload

With the addition of an axial preload P, an additional $P - \delta$ effect becomes relevant. A lateral deformation at the end of the beam causes the axial force P to generate an extra moment on the beam. When P is compressive, this extra moment magnifies the lateral displacement. When P is tensile, this extra moment reduces the lateral displacement. The analytic solution can be found by

considering half the beam as a cantilever as shown in Figure 2-74. Equation 2.20.3 describes end loaded cantilever beam deformation.

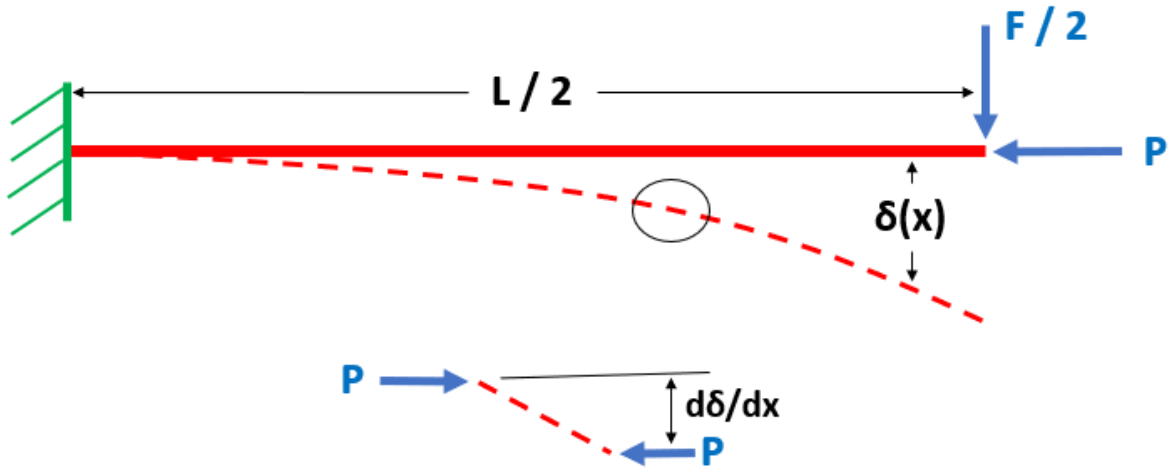


Figure 2-74. – Cantilever Beam With Axial Load.

$$\delta_{bend}(x) = \frac{3FLx^2 - Fx^3}{6EI} \tag{2.20.3}$$

The axial preload applied to the deformed shape generates a distributed moment given in Equation 2.20.4.

$$M_1(x) = P\frac{d\delta_{bend}(x)}{dx} = \frac{PF(2L - x)x}{2EI} \tag{2.20.4}$$

The additional increment of deformation δ_1 of a cantilever subjected to the extra moment from Equation 2.20.4 is given in Equation 2.20.5.

$$\delta_1(x) = \int\int \frac{M_1(x)}{EI} d^2x = -\frac{FP(20L^3x^2 - 5Lx^4 + x^5)}{120E^2I^2} \tag{2.20.5}$$

This additional deformation causes additional cycles of moment and deflection given in Equations 2.20.6 and 2.20.7.

$$M_{n+1}(x) = P\frac{d\delta_n(x)}{dx} \tag{2.20.6}$$

$$\delta_{n+1}(x) = \int\int \frac{M_{n+1}(x)}{EI} d^2x \tag{2.20.7}$$

The exact result for expected deformation can be found by summing the series shown in Equation 2.20.8.

$$\delta(x) = \delta_{bend}(x) + \delta_1(x) + \delta_2(x) + ... + \delta_n(x) \tag{2.20.8}$$

This series converges to a high-precision value after a few terms (unless P is above the buckling load, in which case the series diverges). A Mathematica notebook included in the test directory sums many terms of the series to produce the solution. The analytic and computed solution for tensile and compressive preload are given in Table 2-24.

The **Sierra/SM** nonlinear static solution can directly solve for the preloaded beam deflection. To solve in **Sierra/SD**, first **Sierra/SM** is used to apply the axial preload. The preload stress state is imported into **Sierra/SD** with the `receive_sierra_data` solution case. A lateral load is then applied as a linear static load in **Sierra/SD**. The preloaded stress state alters the element geometric stiffness allowing the correct ultimate deflection to be extracted in **Sierra/SD**.

Table 2-24. – Max Displacement Preloaded Beam.

Preload	Exact Result	**Sierra/SM**	**Sierra/SD** after **Sierra/SM** Preload
Compressive	0.741	0.757	0.748
Tensile	0.392	0.399	0.394

A close match is obtained using both the direct **Sierra/SM** solution method and the **Sierra/SM** axial preload followed by **Sierra/SD** lateral load on the preloaded structure.

2.20.1.3. Preload Equilibrium and Preload Options

By default, the **Sierra/SD** `receive_sierra_data` solution case imports the preload stress state and computes the initial internal force produced by that stress state. The internal force computed from a stress state should be in equilibrium with the external force that caused that stress state. If **Sierra/SD** imports a stress state and applies the same external loads as were applied in **Sierra/SM**, the model should be in equilibrium and compute zero displacement in the **Sierra/SD** solution. The result for this case is given in the first row of Table 2-25. Some small deformation happens in **Sierra/SD**, but it is small compared to the nominal displacement of the system (0.512). The small discrepancy is related to incompatibilities in element formulation between **Sierra/SM** and **Sierra/SD** as well as minor geometric nonlinearities in the **Sierra/SM** solution that cause the **Sierra/SM** equilibrium state to not exactly match the **Sierra/SD** equilibrium state.

As a corollary to maintaining equilibrium, if the preloaded stress state is read into **Sierra/SD** using default options, but no loads are applied in **Sierra/SD**, then the initial forces computed from the stress preload should snap the deformation back to the unloaded state. This result is given in

the second row of Table 2-25. The comparison is good, with small discrepancies related to the formulation differences between **Sierra/SD** and **Sierra/SM**.

The option `include_internal_force` for the `receive_sierra_data` solution case controls whether the internal force associated with the stress state is added to the solution right-hand side. By default, this option is true and this force is included, which accounts for the expected results in the first two columns of Table 2-25. This option can be turned off, in which case the resultant internal force from integrating the stress state is not included. If a model is preloaded in **Sierra/SM** and the preload-causing external forces are not included in the **Sierra/SD** model definition, then the `include_internal_force` option should be off. Effectively, this asserts that the preloaded state should be treated as an exact equilibrium state. The check on this result is given in the third row of Table 2-25. When the initial model state is treated as an exact equilibrium, exactly no displacement is produced in the unloaded **Sierra/SD** model.

A second check of the `include_internal_force` option is given in the fourth row of Table 2-25. In this example, a tensile preload is applied in **Sierra/SM**, and **Sierra/SD** imports this stress state. The **Sierra/SD** model turns off the initial internal force calculation, and consistent with that, does not apply the tensile external force. When this **Sierra/SD** model is subjected to lateral force, it produces the expected deformation of a cantilever with axial tension.

Table 2-25. – Equilibrium Test Cases.

Sierra/SM Load	**Sierra/SD** Load	Options **Sierra/SD** Result	Analytic Expected Max Displacement Result	Computed **Sierra/SD**
Lateral	Lateral	Defaults	0.0	0.0023
Lateral	None	Defaults	-0.512	-0.513
Lateral	None	`include_internal_force off`	0.0	0.0
Tensile	Lateral	`include_internal_force off`	0.392	0.395

2.20.1.4. Geometric Stiffness Options

Usually, the primary reason to include preload is to take into account the geometric stiffing or softening effects of that preload stress. **Sierra/SD** has an option `no_geom_stiff` to ignore this effect. This option can be used to debug models and to see whether the geometric stiffness is causing issues for solvers. Sufficiently large compressive stress can cause stiffness to go negative (physically buckling). Such states will often not solve or could cause stability problems. With the geometric stiffness turned off, **Sierra/SD** will still import the deformed shape and parameters that relate to the material tangent stiffness. A check of the `no_geom_stiff` option is given in

Table 2-26. Here, the deformation of the beam ignoring geometric stiffness is nearly the deformation of a beam without any axial preload. The slight deviation between the tensile and compressive preload state relates to reading the initial model geometry from the **Sierra/SM**-deformed shape, which is slightly different for the compressive and tensile preloads. This result demonstrates that for this example, correct calculation of geometric stiffness is very important to obtain analytic results, while use of the deformed state has very little effect.

Table 2-26. – Geometric Stiffness Test Cases.

Sierra/SM preload	**Sierra/SD** load	Options	Analytic Expected Max Displacement **Sierra/SD** Result	Computed **Sierra/SD** Result
none	lateral		0.512	0.516
compressive	lateral	no_geom_stiff	0.741	0.515
tensile	lateral	no_geom_stiff	0.392	0.517

For input deck see Appendix 9.28.1.

2.20.2. Preloaded Beam Eigen Mode, Abaqus Comparison

This is a verification test comparing Abaqus to Salinas using selective deviatoric hex8 elements. The geometry of this model can be seen in Figure 2-75. The model is a bar that is fixed on one end and constrained in the y and z direction on the other. A prescribed displacement is applied in the x direction in Adagio, and then a modal analysis is performed in Salinas. For verification, the first 4 modes are compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 2-27.

Figure 2-75. – Geometry of Bar.

Table 2-27. – Beam Preload Verification.

Mode Number	Salinas	Abaqus
1	1834.47	1834.50
2	10175.2	10176.0
3	12469.1	12472.0
4	12469.1	12472.0

2.20.3. Preloaded Plate Eigen Mode, Abaqus Comparison

This example is a similar to the previous model, except that it has the geometry of a plate, as shown in Figure 2-76. The plate consists of selective deviatoric hex8 elements and is fixed on one side and constrained in the y and z directions on the other. A prescribed displacement is applied in the x direction in Adagio, and then a modal analysis is performed in Salinas. For verification, the first 5 modes are compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 2-28.
The path to these verification tests is
`Salinas_rtest/verification/adagio_coupling/barModelPreload`.

For input deck see Appendix 9.28.2.

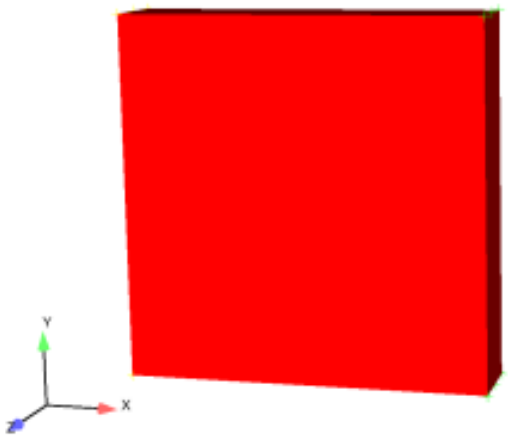


Figure 2-76. – Geometry of plate.

Table 2-28. – Plate Preload Verification.

Mode Number	Salinas	Abaqus
1	1380.37	1406.60
2	1834.47	1834.50
3	5208.10	5212.80
4	7234.86	7236.60
5	8911.89	8914.00

2.20.4. Comparison of Sierra/SD and Sierra/SM Element Formulations

This verification test compares the equivalency of SD and SM element formulations. When developing code-to-code comparison tests, the use of a compatible element formulation greatly aids drawing meaningful conclusions. Additionally, when performing hand-off between Sierra/SM and Sierra/SD, the use of similar element formulations reduces the differences between the approximate solutions.

In this test the response of an arbitrary statically loaded structure is compared between Sierra/SM and Sierra/SD using a wide variety of element formulations. The loading used induces very small strains to minimize any SM to SD difference related to geometrically nonlinear response. An equivalent response between SM and SD indicates the same stiffness matrix is being calculated and used in the two codes.

A moderately refined (r3) solid model for the study is shown in figure 2-77. The solid model is a solid cantilever beam, and the shell model is effectively a hollow cantilever box-beam. Figure figure 2-78 shows the coarse shell element mesh (r1). One end is fixed, and a traction is applied at the opposite end. Comparison is made using the displacement at the end of the body (the marked red dot.) The model is purposely generated with a mixture of skewed elements, curved edges, and a variety of material properties to expose any differences in SD vs. SM handling of these features.

The comparison metric between two solutions with end displacements **u** and **v** is given in equation (2.20.9). A value of $weight$ of 0.0 indicates the two displacements (and thus element formulations) are identical while a $weight$ of 1.0 indicates the two displacements are totally unrelated.

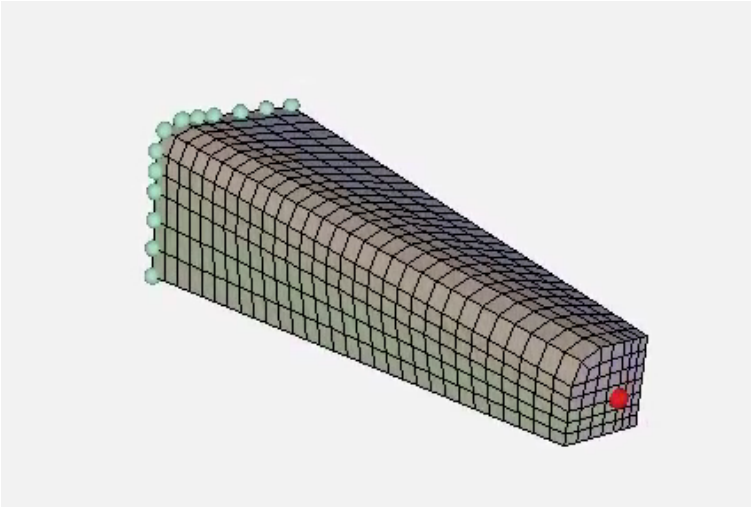


Figure 2-77. – Solid Element Model (R3) Refinement.

$$weight = 1.0 - mag \cdot cosine \tag{2.20.9}$$

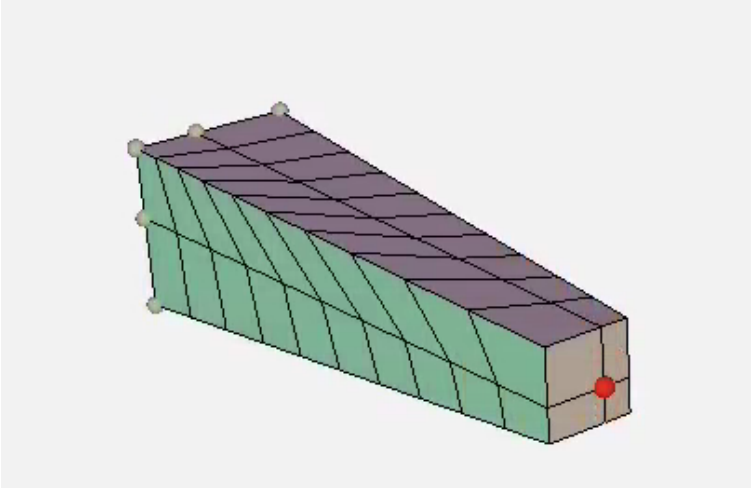


Figure 2-78. – Shell Element Model (R1) Refinement.

where

$$mag = min(\|\mathbf{u}\|, \|\mathbf{v}\|)/max(\|\mathbf{u}\|, \|\mathbf{v}\|)$$
$$cosine = (\mathbf{u} \cdot \mathbf{v})/(\|\mathbf{u}\| \cdot \|\mathbf{v}\|)$$

Several of the shell elements were not functional for SM static solutions at the time of this study. Thus, when generating shell element results in SM a long, damped, explicit dynamics analysis was done. Every attempt was made to equilibrate this result to the true static solution. When comparing results where the element formulation is known to be identical between SM and SD, a difference metric of below 0.00005 was obtained. The small remaining difference between SM and SD can be attributed to slight geometric nonlinearity in the SM solution as well as the finite solver tolerance used in both codes.

The intent is to study most formulations considered in a production state for mesh topologies generated. To limit scope, only the default parameters for these formulations where considered. Additionally, only element topologies that could be handled by both SD and SM were considered.

2.20.4.1. Solid Element Results

Tables 2-29 and 2-30 list details of the element formulations studied. SM has a Hex27 element formulation while SD does not so this was excluded from the study. Additionally, wedge elements are not currently considered in the study, partially due to the difficulty of generating wedge element meshes.

A matrix for the difference metrics obtained comparing each code and element formulation to each other formulation is shown in figure 2-79. The matrix is color-coded such that a difference metric at or below 0.0001 is green (assumed exact formulation match) a metric at or below 0.01 is

	SD_Hex20	SM_Hex20	SD_Hex8_B	SD_Hex8_Full	SD_Hex8_	SM_Hex8	SM_Hex8	SM_Hex8	SM_Hex8	SD_Tet10	SD_Tet10	SM_Tet10	SM_Tet10	SD_Tet4	SM_Tet4
SD_Hex20		0.0000	0.0490	0.1792	0.0641	0.1792	0.0584	0.0641	0.0712	0.0013	0.0013	0.0066	0.0061	0.2545	0.2545
SM_Hex20	0.0000		0.0490	0.1792	0.0641	0.1792	0.0584	0.0641	0.0712	0.0013	0.0013	0.0066	0.0060	0.2545	0.2545
SD_Hex8_B	0.0490	0.0490		0.1373	0.0157	0.1373	0.0095	0.0157	0.1158	0.0503	0.0503	0.0552	0.0432	0.2168	0.2168
SD_Hex8_Full	0.1792	0.1792	0.1373		0.1238	0.0000	0.1294	0.1238	0.2375	0.1803	0.1803	0.1846	0.1742	0.0918	0.0918
SD_Hex8_U	0.0641	0.0641	0.0157	0.1238		0.1238	0.0063	0.0000	0.1295	0.0653	0.0653	0.0701	0.0583	0.2047	0.2047
SM_Hex8_Full	0.1792	0.1792	0.1373	0.0000	0.1238		0.1294	0.1238	0.2375	0.1803	0.1803	0.1846	0.1742	0.0917	0.0918
SM_Hex8_Q1P0	0.0584	0.0584	0.0095	0.1294	0.0063	0.1294		0.0063	0.1240	0.0596	0.0596	0.0645	0.0526	0.2099	0.2099
SM_Hex8_SelDev	0.0641	0.0641	0.0157	0.1238	0.0000	0.1238	0.0063		0.1295	0.0653	0.0653	0.0702	0.0584	0.2046	0.2046
SM_Hex8_UG	0.0712	0.0712	0.1158	0.2375	0.1295	0.2375	0.1240	0.1295		0.0699	0.0699	0.0648	0.0767	0.3081	0.3081
SD_Tet10_Cu	0.0013	0.0013	0.0503	0.1803	0.0653	0.1803	0.0596	0.0653	0.0699		0.0000	0.0053	0.0074	0.2555	0.2555
SD_Tet10	0.0013	0.0013	0.0503	0.1803	0.0653	0.1803	0.0596	0.0653	0.0699	0.0000		0.0053	0.0074	0.2555	0.2555
SM_Tet10_Comp	0.0066	0.0066	0.0552	0.1846	0.0701	0.1846	0.0645	0.0702	0.0648	0.0053	0.0053		0.0126	0.2595	0.2595
SM_Tet10_Full	0.0061	0.0060	0.0432	0.1742	0.0583	0.1742	0.0526	0.0584	0.0767	0.0074	0.0074	0.0126		0.2500	0.2500
SD_Tet4	0.2545	0.2545	0.2168	0.0918	0.2047	0.0917	0.2099	0.2046	0.3081	0.2555	0.2555	0.2595	0.2500		0.0000
SM_Tet4	0.2545	0.2545	0.2168	0.0918	0.2047	0.0918	0.2099	0.2046	0.3081	0.2555	0.2555	0.2595	0.2500	0.0000	

Figure 2-79. – Solid elements comparisons on coarsest models (R1).

Table 2-29. – Details for SD solid element formulations.

Formulation	Block input	Description
SD_Hex20 (default)	hex20	Standard fully integrated Hex20
SD_Hex8_B (default)	hex8b	Bubble function Hex8
SD_Hex8_Full	hex8f	Standard fully integrated Hex8
SD_Hex8_U	hex8u	Selective deviatoric Hex8
SD_Tet10 (default)	tet10	Standard fully integrated Tet10
SD_Tet10_Cu	cutet10	Tet10 formulated by Carlos Felippa (CU stands for U.C. Boulder)
SD_Tet4 (default)	tet4	Standard fully integrated Tet4

yellow (a close but not exact match) and values above 0.02 are red (substantial deviation between formulations.)

Based on the coarse model results a handful of solid elements have identical behavior in Sierra/SD and Sierra/SM. These include the Tet4, Hex20, and fully integrated Hex8 elements. These elements are all fully integrated using standard published element integration rules. Additionally, the selective deviatoric Hex8 element formulations (hex8u in SD) exactly match between SM and SD. This indicates that not just the formulation matches for this element, but also the default parameters for the formulation such as the factor mixing the bulk and deviatoric response.

Unsurprisingly, the default Hex8 (bubble in SD, uniform gradient in SM) and default Tet10 (fully integrated in SD, composite tetrahedron in SM) show substantial differences. Up to 15% difference in results were seen on the coarse meshes for the metric used in this study. In these cases the formulations are indeed fundamentally numerically different. As such care must be taken when handing results between SD and SM or comparing results from SD to SM when using these default elements.

Another interesting finding from figure 2-79 is that the Tet4 result is vastly different from all other results (roughly 25% different in the metric used.) This is consistent with the usual experience of Tet4 element being much worse than other element formulations and producing overly stiff results for a given refinement. However, note that in general no attempt has been made to ascertain which

Table 2-30. – Details for SM solid element formulations.

Formulation	Section input	Description
SM_Hex20 (default)	fully_integrated	Standard fully integrated Hex20
SM_Hex8_Full	fully_integrated	Standard fully integrated Hex8
SM_Hex8_Q1P0	q1p0	Fully integration of deviatoric terms with single point integration of pressure terms
SM_Hex8_SelDev	selective_deviatoric	Fully integration of deviatoric terms with volumetric averaging of pressure terms
SM_Hex8_UG (default)	mean_quadrature	Uniform gradient Hex8 with hourglass control
SM_Tet10_Comp (default)	< blank >	Composite tetrahedral formulation
SM_Tet10_Full	fully_integrated	Full integration of deviatoric terms with volumetric averaging of pressure terms
SD_Tet4 (default)	fully_integrated	Standard fully integrated Tet4

element formulations are potentially more accurate in the study; the purpose of the study is solely to determine which formulations are numerically compatible between Sierra/SD and Sierra/SM.

It is mildly surprising that the fully integrated Tet10 responses do not match between Sierra/SM and Sierra/SD. On a coarse mesh, these Tet10 responses are roughly 1% different. The root cause of this difference for the tet10 elements is that SD uses a traditional fully integrated Tet10 formulation while the SM Tet10 formulation introduces additional volume averaging operations akin to the selective deviatoric element. Without volume averaging, the fully integrated SM tet10 will exactly match the SD tet10.

Though not shown directly in the tables it was observed that the SM total Lagrange formulations exactly match the fully integrated formulations at the small strains used in this study. Additionally, it was observed that the SM total Lagrange hex8 with volume averaging turned on exactly matches the SM and SD selective deviatoric elements.

2.20.4.2. Shell Element Results

Tables 2-31 and 2-32 list details of the element formulations studied. SD has Tri6, Quad8, and Quad9 shell formulations where SM does not have formulations for these topologies. Thus, these higher order shell topologies were excluded from the study.

A matrix for the difference metrics obtained comparing each code and element formulation to each other formulation is shown in figure 2-80. The matrix is color-coded such that a difference metric at or below 0.0001 is green (assumed exact formulation match) a metric at or below 0.01 is

yellow (a close but not exact match) and values above 0.02 are red (substantial deviation between formulations.)

	SD_Quad4_NQUAD	SD_Quad4	SM_Quad4_	SM_Quad	SM_Quad	SM_Quad	SD_Tri3_T	SD_Tri3_T	SM_Tri3_C	SM_Tri3_C
SD_Quad4_NQUAD		0.0958	0.0014	0.1005	0.0926	0.0009	0.0673	0.0512	0.0428	0.0469
SD_Quad4_QUADT	0.0958		0.0945	0.0051	0.0036	0.0950	0.0306	0.0470	0.1345	0.1383
SM_Quad4_BL	0.0014	0.0945		0.0992	0.0913	0.0005	0.0660	0.0499	0.0441	0.0483
SM_Quad4_BT	0.1005	0.0051	0.0992		0.0087	0.0996	0.0355	0.0519	0.1389	0.1427
SM_Quad4_KH	0.0926	0.0036	0.0913	0.0087		0.0917	0.0271	0.0436	0.1314	0.1352
SM_Quad4_NQUAD	0.0009	0.0950	0.0005	0.0996	0.0917		0.0664	0.0503	0.0437	0.0478
SD_Tri3_TRIA3	0.0673	0.0306	0.0660	0.0355	0.0271	0.0664		0.0169	0.1072	0.1111
SD_Tri3_TRISHELL	0.0512	0.0470	0.0499	0.0519	0.0436	0.0503	0.0169		0.0918	0.0958
SM_Tri3_C0	0.0428	0.1345	0.0441	0.1389	0.1314	0.0437	0.1072	0.0918		0.0044
SM_Tri3_ORIG	0.0469	0.1383	0.0483	0.1427	0.1352	0.0478	0.1111	0.0958	0.0044	

Figure 2-80. – Shell elements comparisons on coarsest models (R1).

Table 2-31. – Details for SD shell element formulations.

Formulation	**Block input**	**Description**
SD_Quad4_NQUAD	nquad	Isoparametric shell with bending and membrane stiffness. The 'N' is for Navy
SD_Quad4_QUADT (default)	quadt	Effectively two triangles stuck together. The 'T' is for triangle
SD_Tri3_TRI3 (default)	tria3	Triangular shell formulation from Carlos Felippa of C.U. Boulder
SD_Tri3_TRISHELL	triashell	Combines Allman's triangle formulation for membrane behavior and discrete Kirchhoff triangle for bending behavior

It was observed that no shell formulations were exactly equivalent between SD and SM. Though for quad elements the SD nquad, SM nquad, and SM BL formulations are all very similar and for most purposes could be considered equivalent. Again the default quad formulations (SD QuadT and SM BT) are substantially different. The QuadT is effectively two triangles joined at the diagonal while the BT is the Belytschko-Tsay formulation which is formulated for explicit dynamics speed and is missing certain terms for transverse shear found in the nquad and BL (Belytschko-Leviathan) formulation. Care should be taken using these default quad elements when handing off or comparing results between SD and SM. For triangular shells there are no formulations that appear to even be approximately similar between SD and SM.

Table 2-32. – Details for SM shell element formulations.

Formulation	Section input	Description
SM_Quad4_BL	bl_shell	Belytschko-Leviathan formulation
SM_Quad4_BT	bt_shell	Belytschko-Tsay formulation
SM_Quad4_KH	kh_shell	Key-Hoff formulation
SM_Quad4_NQUAD	nquad	Isoparametric shell with bending and membrane stiffness. The 'N' is for Navy
SM_Tri3_C0	c0_tri_shell	Kennedy, Belytschko, Lin formulation
SM_Tri3_ORIG	orig_tri_shell	Original default triangular shell formulation later replaced by C0. Formulation from Sam Key

2.20.4.3. Refinement Study

An additional refinement study was performed to confirm the expected result that with refinement the differences between element formulations become less relevant as all well posed formulations coverage to the same result. The properties of the refined meshes are given in table 2-33. All refinements are approximately uniform.

Table 2-33. – Meshes for refinement study.

Topology	Refinement	Number of Elements
Hex20 / Hex8	r1	60
Hex20 / Hex8	r3	1440
Hex20 / Hex8	r6	7860
Tet10 / Tet4	r1	1018
Tet10 / Tet4	r3	10421
Tet10 / Tet4	r6	77703
Quad4	r1	84
Quad4	r3	756
Quad4	r6	2892
Tri3	r1	178
Tri3	r3	1618
Tri3	r6	6525

The metric comparison plots for the refinement study are given in figures 2-81 to 2-84. Qualitatively, it is observed that with refinement the SD and SM solid and shell results for all formulations become much more similar to one another. E.g. the difference metric plots transition from mostly red (greater than 2% difference in results) to mostly green (less than 0.01% difference in results.) This is expected and indicates that each element formulation converges towards the same result with refinement. However, no detailed quantitative study has been done looking at convergence rates.

	SD_Hex20	SM_Hex20	SD_Hex8_B	SD_Hex8_Full	SD_Hex8_	SM_Hex8_	SM_Hex8_	SM_Hex8_	SM_Hex8_	SD_Tet10_	SD_Tet10	SM_Tet10	SM_Tet10	SD_Tet4	SM_Tet4
SD_Hex20		0.0000	0.0089	0.0280	0.0146	0.0281	0.0135	0.0146	0.0018	0.0002	0.0002	0.0014	0.0004	0.0426	0.0426
SM_Hex20	0.0000		0.0089	0.0280	0.0145	0.0280	0.0135	0.0146	0.0019	0.0002	0.0002	0.0014	0.0004	0.0426	0.0426
SD_Hex8_B	0.0089	0.0089		0.0193	0.0057	0.0193	0.0046	0.0057	0.0107	0.0092	0.0092	0.0103	0.0085	0.0340	0.0340
SD_Hex8_Full	0.0280	0.0280	0.0193		0.0137	0.0000	0.0147	0.0137	0.0298	0.0282	0.0282	0.0294	0.0276	0.0150	0.0150
SD_Hex8_U	0.0146	0.0145	0.0057	0.0137		0.0137	0.0011	0.0000	0.0163	0.0148	0.0148	0.0159	0.0142	0.0285	0.0285
SM_Hex8_Full	0.0281	0.0280	0.0193	0.0000	0.0137		0.0148	0.0137	0.0298	0.0283	0.0283	0.0294	0.0277	0.0150	0.0150
SM_Hex8_Q1P0	0.0135	0.0135	0.0046	0.0147	0.0011	0.0148		0.0011	0.0153	0.0137	0.0137	0.0149	0.0131	0.0295	0.0296
SM_Hex8_SelDev	0.0146	0.0146	0.0057	0.0137	0.0000	0.0137	0.0011		0.0164	0.0148	0.0148	0.0160	0.0142	0.0285	0.0285
SM_Hex8_UG	0.0018	0.0019	0.0107	0.0298	0.0163	0.0298	0.0153	0.0164		0.0016	0.0016	0.0004	0.0022	0.0444	0.0444
SD_Tet10_Cu	0.0002	0.0002	0.0092	0.0282	0.0148	0.0283	0.0137	0.0148	0.0016		0.0000	0.0012	0.0006	0.0428	0.0428
SD_Tet10	0.0002	0.0002	0.0092	0.0282	0.0148	0.0283	0.0137	0.0148	0.0016	0.0000		0.0012	0.0006	0.0428	0.0428
SM_Tet10_Comp	0.0014	0.0014	0.0103	0.0294	0.0159	0.0294	0.0149	0.0160	0.0004	0.0012	0.0012		0.0018	0.0439	0.0440
SM_Tet10_Full	0.0004	0.0004	0.0085	0.0276	0.0142	0.0277	0.0131	0.0142	0.0022	0.0006	0.0006	0.0018		0.0422	0.0422
SD_Tet4	0.0426	0.0426	0.0340	0.0150	0.0285	0.0150	0.0295	0.0285	0.0444	0.0428	0.0428	0.0439	0.0422		0.0000
SM_Tet4	0.0426	0.0426	0.0340	0.0150	0.0285	0.0150	0.0296	0.0285	0.0444	0.0428	0.0428	0.0440	0.0422	0.0000	

Figure 2-81. – Solid elements comparisons on moderate refinement models (R3).

	SD_Hex20	SM_Hex20	SD_Hex8_B	SD_Hex8_Full	SD_Hex8_	SM_Hex8_	SM_Hex8_	SM_Hex8_	SM_Hex8_	SD_Tet10_	SD_Tet10	SM_Tet10	SM_Tet10	SD_Tet4	SM_Tet4
SD_Hex20		0.0000	0.0025	0.0081	0.0036	0.0082	0.0032	0.0036	0.0018	0.0001	0.0001	0.0004	0.0002	0.0131	0.0131
SM_Hex20	0.0000		0.0024	0.0081	0.0036	0.0081	0.0032	0.0036	0.0018	0.0002	0.0002	0.0004	0.0001	0.0130	0.0131
SD_Hex8_B	0.0025	0.0024		0.0057	0.0011	0.0057	0.0008	0.0012	0.0043	0.0026	0.0026	0.0029	0.0023	0.0106	0.0106
SD_Hex8_Full	0.0081	0.0081	0.0057		0.0045	0.0000	0.0049	0.0045	0.0099	0.0083	0.0083	0.0085	0.0080	0.0050	0.0050
SD_Hex8_U	0.0036	0.0036	0.0011	0.0045		0.0046	0.0004	0.0000	0.0054	0.0037	0.0037	0.0040	0.0034	0.0095	0.0095
SM_Hex8_Full	0.0082	0.0081	0.0057	0.0000	0.0046		0.0049	0.0045	0.0100	0.0083	0.0083	0.0085	0.0080	0.0049	0.0050
SM_Hex8_Q1P0	0.0032	0.0032	0.0008	0.0049	0.0004	0.0049		0.0004	0.0050	0.0034	0.0034	0.0036	0.0031	0.0099	0.0099
SM_Hex8_SelDev	0.0036	0.0036	0.0012	0.0045	0.0000	0.0045	0.0004		0.0054	0.0038	0.0038	0.0040	0.0035	0.0095	0.0095
SM_Hex8_UG	0.0018	0.0018	0.0043	0.0099	0.0054	0.0100	0.0050	0.0054		0.0017	0.0017	0.0014	0.0020	0.0148	0.0149
SD_Tet10_Cu	0.0001	0.0002	0.0026	0.0083	0.0037	0.0083	0.0034	0.0038	0.0017		0.0000	0.0003	0.0003	0.0132	0.0132
SD_Tet10	0.0001	0.0002	0.0026	0.0083	0.0037	0.0083	0.0034	0.0038	0.0017	0.0000		0.0003	0.0003	0.0132	0.0132
SM_Tet10_Comp	0.0004	0.0004	0.0029	0.0085	0.0040	0.0085	0.0036	0.0040	0.0014	0.0003	0.0003		0.0006	0.0134	0.0135
SM_Tet10_Full	0.0002	0.0001	0.0023	0.0080	0.0034	0.0080	0.0031	0.0035	0.0020	0.0003	0.0003	0.0006		0.0129	0.0129
SD_Tet4	0.0131	0.0130	0.0106	0.0050	0.0095	0.0049	0.0099	0.0095	0.0148	0.0132	0.0132	0.0134	0.0129		0.0000
SM_Tet4	0.0131	0.0131	0.0106	0.0050	0.0095	0.0050	0.0099	0.0095	0.0149	0.0132	0.0132	0.0135	0.0129	0.0000	

Figure 2-82. – Solid elements comparisons on high refinement models (R6).

	SD_Quad4_NQUAD	SD_Quad4	SM_Quad4_	SM_Quad	SM_Quad	SM_Quad	SD_Tri3_T	SD_Tri3_T	SM_Tri3_C	SM_Tri3_C
SD_Quad4_NQUAD		0.0058	0.0002	0.0133	0.0097	0.0008	0.0045	0.0053	0.0008	0.0023
SD_Quad4_QUADT	0.0058		0.0060	0.0075	0.0039	0.0066	0.0014	0.0005	0.0066	0.0081
SM_Quad4_BL	0.0002	0.0060		0.0135	0.0099	0.0006	0.0046	0.0055	0.0006	0.0021
SM_Quad4_BT	0.0133	0.0075	0.0135		0.0037	0.0141	0.0089	0.0081	0.0141	0.0155
SM_Quad4_KH	0.0097	0.0039	0.0099	0.0037		0.0105	0.0052	0.0044	0.0105	0.0119
SM_Quad4_NQUAD	0.0008	0.0066	0.0006	0.0141	0.0105		0.0052	0.0061	0.0000	0.0015
SD_Tri3_TRIA3	0.0045	0.0014	0.0046	0.0089	0.0052	0.0052		0.0008	0.0053	0.0067
SD_Tri3_TRISHELL	0.0053	0.0005	0.0055	0.0081	0.0044	0.0061	0.0008		0.0061	0.0075
SM_Tri3_C0	0.0008	0.0066	0.0006	0.0141	0.0105	0.0000	0.0053	0.0061		0.0014
SM_Tri3_ORIG	0.0023	0.0081	0.0021	0.0155	0.0119	0.0015	0.0067	0.0075	0.0014	

Figure 2-83. – Shell elements comparisons on moderate refinement models (R3).

	SD_Quad4_NQUAD	SD_Quad4	SM_Quad4_	SM_Quad	SM_Quad	SM_Quad	SD_Tri3_T	SD_Tri3_T	SM_Tri3_C	SM_Tri3_C
SD_Quad4_NQUAD		0.0023	0.0002	0.0038	0.0028	0.0004	0.0009	0.0008	0.0002	0.0006
SD_Quad4_QUADT	0.0023		0.0021	0.0061	0.0051	0.0019	0.0014	0.0031	0.0021	0.0017
SM_Quad4_BL	0.0002	0.0021		0.0041	0.0030	0.0001	0.0007	0.0011	0.0000	0.0004
SM_Quad4_BT	0.0038	0.0061	0.0041		0.0011	0.0042	0.0047	0.0030	0.0040	0.0044
SM_Quad4_KH	0.0028	0.0051	0.0030	0.0011		0.0032	0.0037	0.0020	0.0030	0.0034
SM_Quad4_NQUAD	0.0004	0.0019	0.0001	0.0042	0.0032		0.0005	0.0012	0.0002	0.0002
SD_Tri3_TRIA3	0.0009	0.0014	0.0007	0.0047	0.0037	0.0005		0.0017	0.0007	0.0003
SD_Tri3_TRISHELL	0.0008	0.0031	0.0011	0.0030	0.0020	0.0012	0.0017		0.0010	0.0014
SM_Tri3_C0	0.0002	0.0021	0.0000	0.0040	0.0030	0.0002	0.0007	0.0010		0.0004
SM_Tri3_ORIG	0.0006	0.0017	0.0004	0.0044	0.0034	0.0002	0.0003	0.0014	0.0004	

Figure 2-84. – Shell elements comparisons on high refinement models (R6).

Note that even when using identically formulated elements differences can arise between Sierra/SD and Sierra/SM for a variety of other reasons. At larger strains SD and SM results will diverge due to fundamentally different small strain vs. large strain assumptions. Additionally, when considering dynamics or gravity loads, SD and SM can diverge due to different default mass matrix representations (consistent in SD vs. lumped in SM.) Finally, for more complex models some boundary conditions, loads, or constraint conditions can have different formulations in the codes.

For input deck see Appendix 9.28.3.

2.20.4.4. One Dimensional Element Results

Spring elements A spring of unit length is used, fixed at one end. At the opposite end, tangential displacements are also fixed. Note that in SD zero length springs are recommended. The mass of the SM spring is set to 0 for consistency. The force function used in SM is also consistent with the stiffness. **The SD and SM spring elements match.**

Truss elements The geometry figure 2-85 consists of 3 disconnected aluminium bars with cross-sectional area $1/4$ and length 10. Eight elements are used per line. The mesh consists of two element blocks, each block containing half of each of the 3 lines. Each bar is fixed at one end. At the opposite end, the tangential displacements are also fixed, and an axial load of magnitude 1000 is applied. The deformations are tested, but only the x component is tabulated. **The SD and SM truss elements match.**

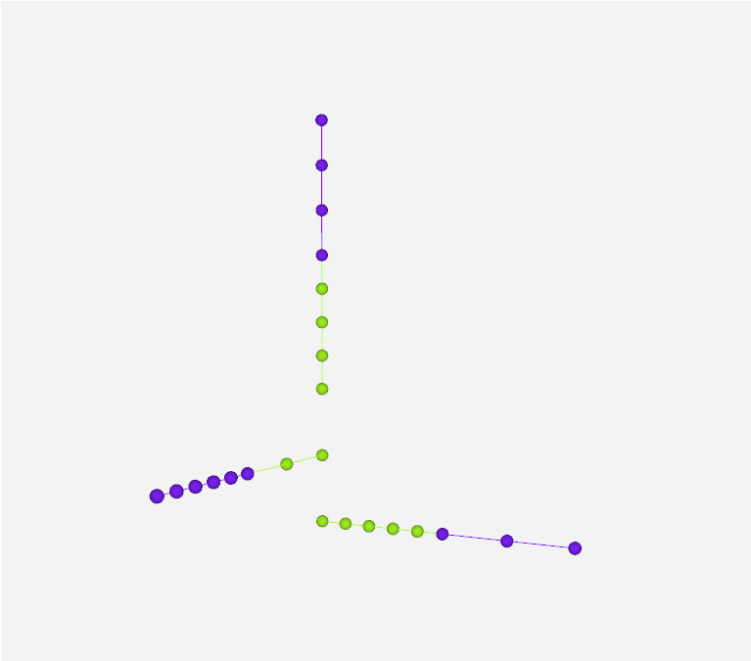


Figure 2-85. – Truss Element Model.

Beam elements with axial force Four cross sections are tested: bar, box, rod, and tube. The mesh consists of 4 disconnected beams, each of length 5, one per cross section. The dimensions of each cross section are chosen so that each cross section has unit area. The second moments of area, $I_1 = I_2$, are determined from the diameters. The thickness direction is chosen to a direction normal to the beam.**The SD and SM beam elements match for axial forces.**

Beams in bending The SD and SM beams do not match. The bar cross section is used here for simplicity. As was the case for the axially loaded beam, the material properties are chosen so that each beam cross section has unit area. Here the length is 20. Figure 2-86 compares the maximum displacements from the SM (maise) and SD (green) simulations. As the number of elements increases, the discrepancy decreases, from 4 percent on a two element mesh to 0.3 percent on a 32 element mesh. Figure 2-87 shows the maximum rotations from the SM (blue) and SD (green)

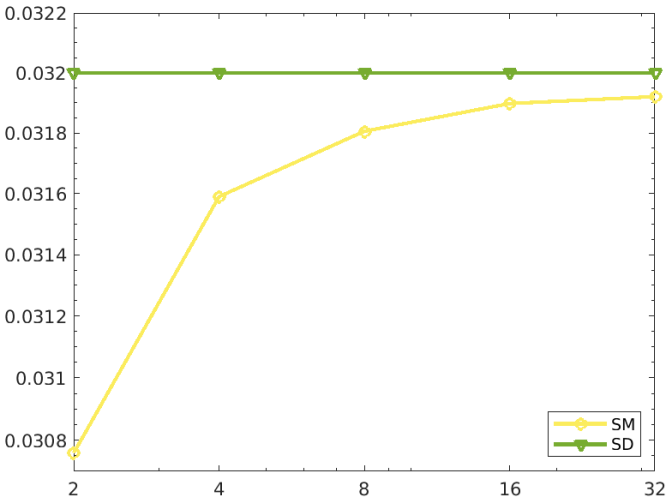


Figure 2-86. – The maximum deflection is plotted versus the number of elements. The displacements results converge under mesh refinement.

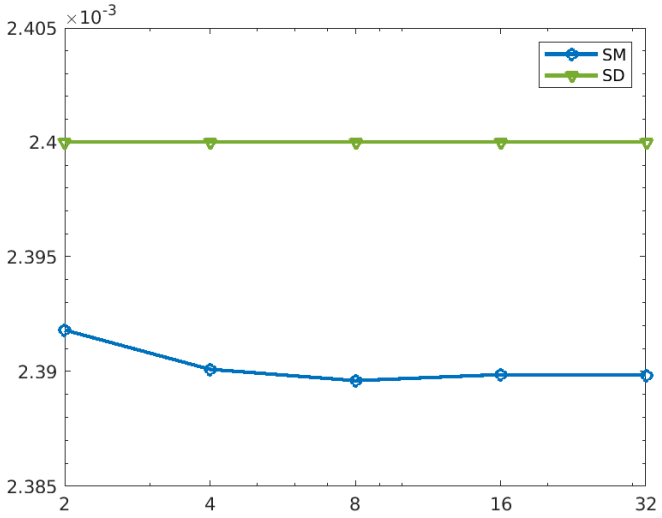


Figure 2-87. – Bending Comparison of Beam Elements: Rotations. The amount of rotation is plotted versus the number of elements. The inconsistency remains constant with mesh refinement.

simulations. The 0.3 percent discrepancy in the rotations appears to be independent of the number of elements, i.e. **The SD and SM beams do not match under a twisting force**.

Beam element torque The model is a beam of length 5. The mesh consists of one element. All dofs are fixed except for rotations about the x-axis. One end of the beam is fixed. A moment about the x-axis if magnitude 100 is applied at the opposite end.

For input decks see Appendix 9.28.3.

2.21. Sensitivity to Parameters

Sensitivity to parameters is available for multiple solution types. The primary application is in eigen analysis where the semi-analytic solutions can provide significant computation and accuracy benefit over a finite difference approach. A script was developed for testing different parameters using the finite difference method in Sierra-SD. The script checks that, as the step size decreases, the finite difference approximation to the modal sensitivity converges to the value provided in the code. A simple model was developed and analyzed for verification. This model is two hex elements that are connected via a tied joint. The $Kz = elastic1e7 + /-10$ parameter in the Joint2G block is where the sensitivity analysis is preformed. Figure 2-88 is a plot of the results and shows this capability. The Eigenvalue sensitivity information can be found in the result file and matches the value shown in Figure 2-88.

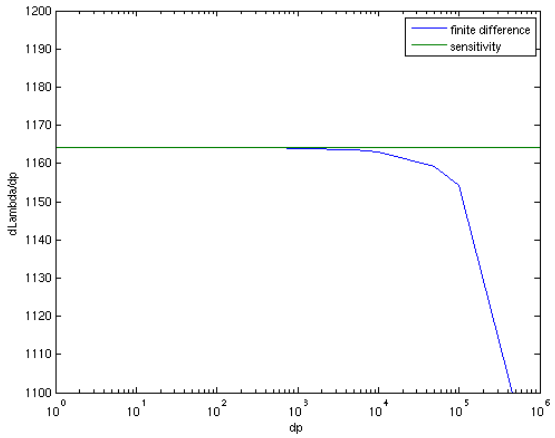


Figure 2-88. – $d\Lambda/dp$ vs. dp.

Figure 2-89 shows the frequency vs. dp. For input deck see Appendix 9.29.

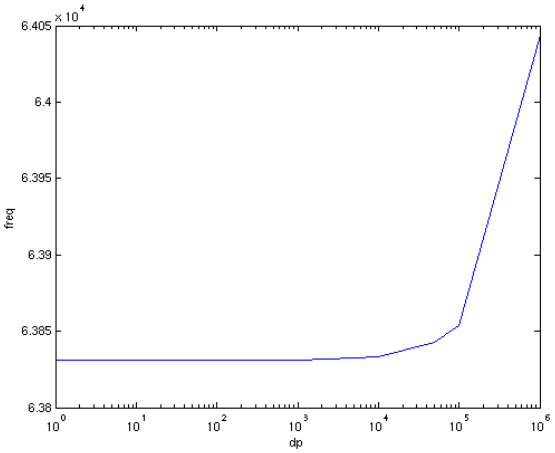


Figure 2-89. – Frequency vs. dp.

2.22. Shock Tube

Analysis Type	Nonlinear Acoustics
Element Type	Hex8
Boundary Conditions	absorbing, fixed velocity
Keywords	nonlinear acoustics, run time compiler

2.22.1. Problem Description

This is the verification test of nonlinear acoustics.

2.22.2. Verification of Solution

The SierraSD nonlinear acoustics equation is the Kuznetsov equation. In the SierraSD Verification manual, see Section 8.1 and specifically Section 8.2. Fubini's exact solution to a wave guide is used. A MATLAB script, `shocktube_exact_solution.m` generates the exact solution.

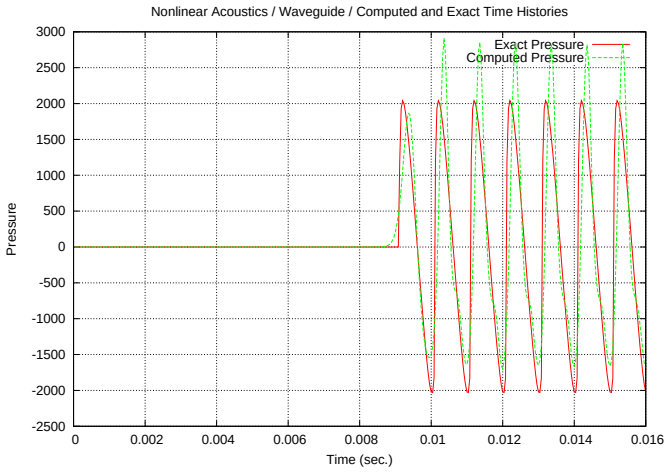


Figure 2-90. – Shock Tube.

For input deck see Appendix 9.30.

3. MATERIALS

3.1. Linear Viscoelasticity

Linear viscoelasticity is a physics whose implementation in structural dynamics code is not uncommon. The most conventional implementation is that which employs Prony series (see Theory and User's manuals.) Again, the purpose of verification is to assure that the conventional implementation is done correctly.

For this test, we consider a beam of isochoric, isotropic viscoelastic material subject to normal displacements in one direction consistent with a uniform compression. The imposed displacement is ramped up and held at a fixed value. After the material is deformed at a rate $\dot{\gamma}$ for a period Δt and then held, the resulting stress will be,

$$\sigma(t) = \dot{\gamma} E_\infty \Delta t - \dot{\gamma} \sum_n (E_G - E_\infty)\tau_n (1 - e^{\Delta t/\tau_n}) e^{-(t+\Delta t)/\tau_n} \tag{3.1.1}$$

A plot of the above exact solution and the predictions of **Sierra/SD** are presented in Figure 3-1.

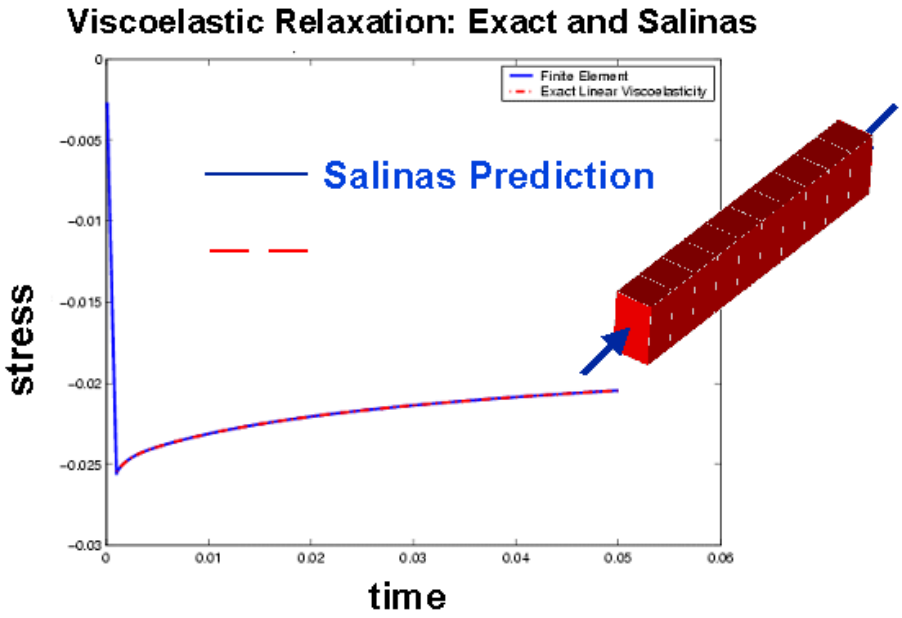


Figure 3-1. – Viscoelastic Relaxation. The **Sierra/SD** results reproduce the exact solution viscoelastic relaxation after ramp and hold deformation.

3.2. Thermally Induced Elastic Waves: Hollow Sphere

This test compares thermally induced elastic vibrations in a hollow sphere with an analytic solution from the 1965 paper *Thermal Stress-Wave Propagation in Hollow Elastic Spheres* by Tsui and Kraus. The inner surface is heated suddenly while the outer surface is held at the initial temperature, causing an elastic wave to propagate from the inner surface to the outer surface. It should be noted that the properties chosen are implausible for real materials: the elastic wave speed is unphysically low, and the thermal diffusivity is unphysically high. Thus, the time required for the temperature to reach equilibrium is on the order of the time required for the elastic wave to travel to the outer surface. This results in a more challenging dynamic test because the quasi-static approximation is not valid.

Table 3-1. – Parameter Definitions in Tsui and Kraus.

a	sphere inner radius
b	sphere outer radius
μ	shear modulus
ν	Poisson ratio
β	density
α	coefficient of thermal expansion
κ	thermal diffusivity

We begin by describing the solution found in the paper, with key parameter definitions given in Table 3-1. The temperature $T(r,t)$ solves the heat equation

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{3.2.1}$$

$$T(a,t) = T_a \tag{3.2.2}$$

$$T(b,t) = 0 \tag{3.2.3}$$

$$T(r,0) = 0, \tag{3.2.4}$$

where κ is the thermal diffusivity. The Sierra code Aria is used to compute T, but it solves an energy conservation equation which reduces to the heat equation provided that the specific heat capacity C_p, density β, and thermal conductivity k are related to the thermal diffusivity κ in the heat equation as follows:

$$\kappa = \frac{k}{\beta C_p}. \tag{3.2.5}$$

The change from zero temperature induces a thermal strain $\epsilon_{ii} = \alpha T$, which drives the elastodynamic equations.

Tsui and Kraus introduce a dimensionless "inertia" parameter

$$\gamma = \frac{\kappa}{ca}, \tag{3.2.6}$$

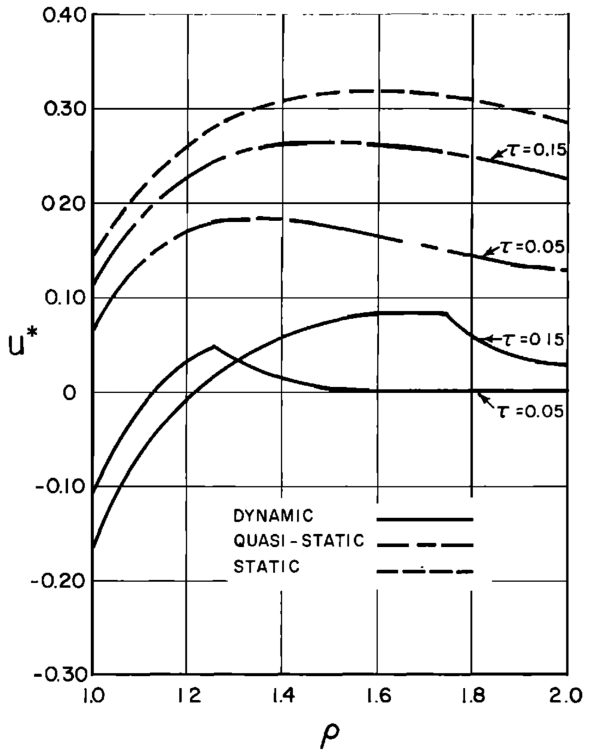


Figure 3-2. – This is Figure 1 from Tsui and Kraus. Plotted are the dimensionless radial displacement u^*, which is related to the physical displacement by $u^* = [(1-\nu)/(a\alpha T_a * (1+\nu))]u$ against the dimensionless radius $\rho = r/a$. We are interested in the dynamic case (solid line) at dimensionless times $\tau = 0.05, 0.15$, where $\tau = \kappa t/a^2$.

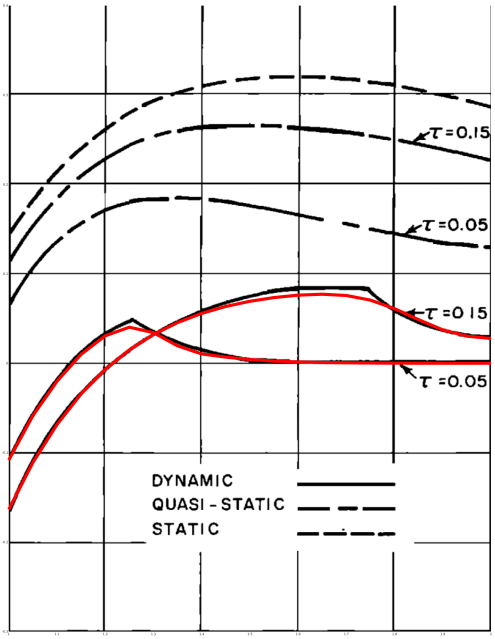


Figure 3-3. – Overlay of Sierra SD results on Figure 3-2.

where the propagation speed of elastic pressure waves is given by

$$c = \sqrt{\frac{2(1-\nu)\mu}{\beta(1-2\nu)}}. \tag{3.2.7}$$

In Figure 3-2, the analytic solution of Tsui and Kraus is plotted for $\gamma = 1/5$, $b/a = 2$, and $\nu = 1/3$. Note that values of γ are much smaller than this for real materials, e.g., approximately 10^{-8} for steel. We choose $a = 1$, $\kappa = 1$, and $c = 5$ so that $\gamma = 1/5$ as in the paper. We set $c = 5$ by choosing $\mu = 25/4$, and $\beta = 1$. The temperature parameters are chosen with $\alpha = 10^{-2}$, and $T_a = 1$. These choices imply (see the definitions in the caption of Figure 3-2) that $\rho = r$, $\tau = t$, and $u^* = 50u$.

Results using Sierra SD are shown in Figure 3-3. We do not make a direct numerical comparison for two reasons: the analytical formula in Tsui and Kraus is based on series solutions, and is very challenging to evaluate. Furthermore, they provide no table of the values computed using the analytic solution, so the best that we can do is scale the graphs by hand to line up the axes. Agreement is excellent, except for the kink at the propagating wavefront, which could presumably be better resolved with a finer mesh or finer initial timesteps.

For input deck see Appendix 9.31

3.3. Thermal Expansion

In this section we give verification examples for thermal expansion.

3.3.1. Free beam

This example consists of a free floating beam that is subjected to a uniform temperature increase of 178^o. The built-in end is such that expansion can occur without generating any stresses. In the end, the beam is stress free but undergoes a uniform expansion. The exact solution for the tip displacement is

$$\Delta L = \alpha L \Delta T = 0.0001 \times 50 \times 178 = 0.89 \tag{3.3.1}$$

where α is the coefficient of thermal expansion, and L is the length of the beam. **Sierra/SD** gives the exact answer of 0.89. This test is included in the verification test suite in the following directory

`tests/Salinas_rtest/verification/thermal/thermal_beam.xml.`

3.3.2. Free beam with linear temperature distribution

This is also a free floating beam example, except that the temperature variation is linear along the length of the beam, instead of the uniform temperature of the previous example. The exact axial displacement of the end of the beam is given by (thanks to Jason Hales for the derivation of this equation)

$$u(x) = \alpha(T_0 - T_i)x + \alpha(T_L - T_0)\frac{x^2}{2L} \tag{3.3.2}$$

where T_0 is the temperature of the beam at the fixed end, T_L is the temperature of the beam at the free end, and T_i is the initial (uniform) temperature of the beam. Plugging in the parameters for this example gives

$$u(L) = 0.0001 * 1 * 50 + 0.0001 * 1 * 25 = 0.0075 \tag{3.3.3}$$

This example is also included in the verification test suite in the following directory,

`tests/Salinas_rtest/verification/thermal/thermal_beam2.xml.`

A note about the boundary conditions for these tests may be useful. These examples simulate free expansion. The boundary conditions are applied at one end to eliminate rigid body modes which generate solution difficulty. The example with linear temperature distribution results in a free expansion solution that is concave at the constraint end. Original boundary conditions constrained that surface to be planar, and resulted in a solution that was about 1% in error. Relaxing the boundary conditions to the minimal set results in a much better solution.

3.4. Thermal/Structural Responses (TSR)

Sierra/SD is not used to compute a thermal solution. However, input temperature or energy density may be applied to the materials or to determining the thermal stress and strain We support this interaction as follows.

TSR_preload reads an initial stress and stores it on the body. An internal force response is computed. No deformation is determined, and the element matrices are not modified. It is typically followed by a static or transient dynamic response.

Thermal load may be applied to a body. The load may be specified on all nodes, on element centroids, or on element integration (or Gauss) locations.

Material Properties Temperature dependent material properties are supported. A user provided function determines the property as a function of temperature.

Energy Density may be used as a thermal input for elements. The energy density is specified on element centroids or element integration points and converted to temperature using the specific heat capacity. Energy density may not be specified as a nodal quantity.

More detailed information is available in the corresponding sections of the SierraSD Users' Guide.

Test Matrix We would like to generate effective tests that verify that these capabilities are working properly, and especially that they work together. The test matrix shown in Table 3-2 summarizes the tests. Particular emphasis is paid to combined capabilities.

Section	TSR_preload	statics	NLstatics	trans	load	Material
3.4	X					
3.4	X					X
3.4	X	X				X
3.4	X		X			X
3.4	X			X		X
3.4	X	X			X	X
3.4	X	X		X	X	X
3.4	X	X		X	X	X

Table 3-2. – Thermal/Structural Test Matrix. All tests apply temperature inputs except 3.4, which is a repeat of 3.4 applying energy inputs.

Thermal Model Definition The model is a perfect unit cube with a uniform thermal load on a single block. No other boundary conditions are applied. In the following NLStatics refers to the nonlinear statics solution method.

TSR The model, defined in Section 3.4, results in a uniform stress throughout the single hex element. For this solution case, no deformation results. Applying a Young's modulus of 30×10^6, and a thermal expansion coefficient, $\alpha = 10^{-6}$ together with the temperature change of $\Delta T = 5$,

results in a thermal strain of 5 x 10^{-6} and a stress of 150. Note that the total structural strain is zero, as the body cannot deform in this solution.

TSR with Thermal Material Results are identical to Section 3.4, but the material properties are determined using a Young's modulus which depends on temperature.

TSR, with Thermal Material followed with Statics Following the solution of Section 3.4 with linear statics equilibrium results in a solution with zero stress and a net strain of 5 x 10^{-6}. Deformations match the strain. The statics solution is only well posed when the rigid by motion is constrained.

TSR, with Thermal Material followed with NLStatics Following the solution of Section 3.4 with nonlinear static equilibrium results in a solution with zero stress and a net strain of 5 x 10^{-6}. Deformations match the strain. The statics solution is only well posed when the rigid by motion is constrained.

TSR, with Thermal Material followed with Transient Following the solution of Section 3.4 with damped transient equilibrium results in a solution that oscillates about the solution of Section 3.4, with a net strain of 5 x 10^{-6}. Deformations match the strain.

TSR, with Thermal Material followed with Loaded Statics Section 3.4 determines an unloaded equilibrium. The same thermal load may be applied with a negative scale factor, resulting in zero strain – the initial (TSR) stress is exactly balanced by an opposing stress. To better verify the code, we apply a negative thermal stress that is three times the original stress, resulting in a solution with zero stress and a net strain of -10 x 10^{-6}. Deformations match the strain. The statics solution is only well posed when the rigid by motion is constrained.

TSR, Thermal Material followed with Loaded Statics & Dynamics We follow the solution of Section 3.4 with a transient load scaled with the original force. This TSR pushes the solution out. The statics solution pushes it back in, to twice the strain of TSR. Dynamics results in a solution that oscillates about 5 x 10^{-6}.

TSR, Thermal Material, Loaded Statics & Dynamics with Energy Specific energy may be supplied as the input to the TSR and static and dynamic loading. The specific energy is converted to temperature using the specific heat. Material properties are determined from the *temperature*, not the energy in the body.

3.5. Direct Energy Deposition at Gauss Points

Energy deposited in the body (as by an X-ray event) can result in an instantaneous change in temperature. For consistency with other applications, the energy is applied as a specific energy, i.e., the energy per unit mass, $\tilde{E} = Q/(\rho V)$. Because such energy typically decays exponentially, it is important that energy be provided at the Gauss points especially for larger, higher order elements.

3.5.1. Two Element Linear Variation Hex20

The example consists of two unit Hex20 elements forming a beam of dimension 2x1x1. The specific energy varies as the long dimension of the beam, X. The geometry is shown in Figure 3-4.

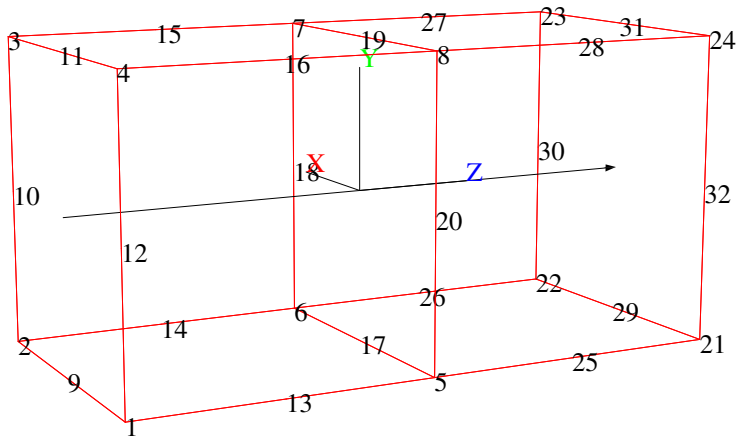


Figure 3-4. – Simple Energy Deposition Test Geometry.

We have the following:

- The specific energy is properly read into **Sierra/SD**, as verified with line sample output.
- The specific energy is properly converted to temperature using the specific heat of the material.
- The total energy input is determined properly.
- Resulting displacements meet the analytic solutions (see Figure 3-5). The numerical results are obtained by using *Ensight* to post process the displacements through the center of the body. The analytic displacement may be obtained by using the one dimensional ODE generated by the thermal stress.

$$\begin{aligned} \epsilon_{thermal} &= \frac{du}{dX} = \alpha_t T(X) \\ &= \alpha_t X / C_v \\ u(x) &= \frac{\alpha_t}{2C_v} X^2 \end{aligned}$$

The corresponding nightly test may be found in,

`tests/Salinas_rtest/verification/thermal/edep_lin.xml.`

Resulting displacements are quadratic as from equation 3.3.2, with $\alpha = 0.001$, and $T_L = 1$.

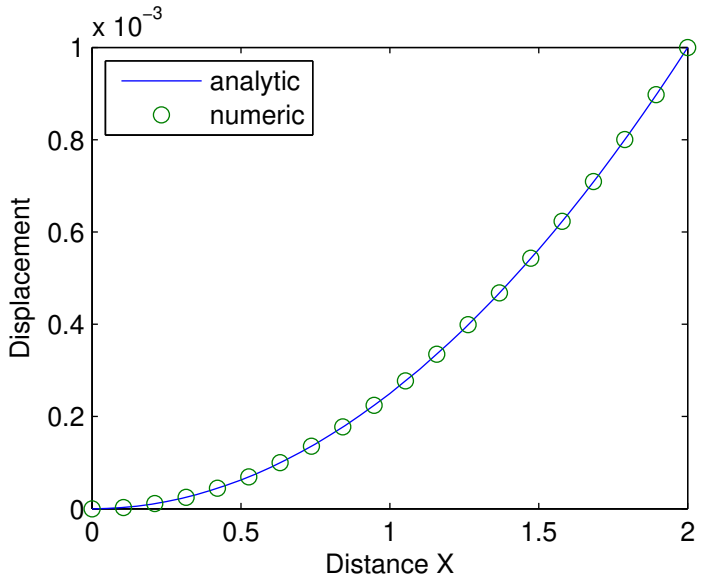


Figure 3-5. – Displacements Resulting from Linear Temperature Profile.

3.5.2. Two Element Quadratic Variation Hex20

This test uses the same geometry described in Section 3.5.1 and Figure 3-4, but with specific energy variation, $\tilde{E}(x, y, z) = x^2 + y^2 + z^2$. The example ensures the following:

1. Exact representation of the energy and temperature as shown in linedata.
2. The total energy is $\rho \int_{elem}(x^2 + y^2 + z^2)dx\,dy\,dz$, which is 3ρ, where ρ is the density.
3. Ensures numbering of the Gauss points.
4. The displacement is inexact, as the analytic solution is cubic.

3.5.3. Two Element Exponential Decay Variation Hex20

This test uses the same geometry described in Section 3.5.1, but with specific energy variation, $\tilde{E}(x, y, z) = e^{-x}$. The example ensures the following:

1. Approximate representation of the energy and its error can be extracted using line sample (`linesample`) data and is represented in Figure 3-6.
2. The total energy is $E_t = \rho(1 - e^{-2})$. The solution is approximate, because the energy is represented by a quadratic in each element, but the error is less than 10^{-5}.
3. The displacement is inexact. The one dimensional thermal strain equation provides the ODE for the solution. We use $T(x) = \tilde{E}/C_v$. Then,

$$\epsilon_{thermal} = \frac{du}{dX} = \frac{\alpha_t}{C_v}\, e^{-\gamma X} \tag{3.5.1}$$

Figure 3-7 from the `edep_expx` test compares the computed solution to the exact solution,

$$u = \frac{\alpha_t}{C_v \gamma} \left(1 - e^{-\gamma X}\right) \tag{3.5.2}$$

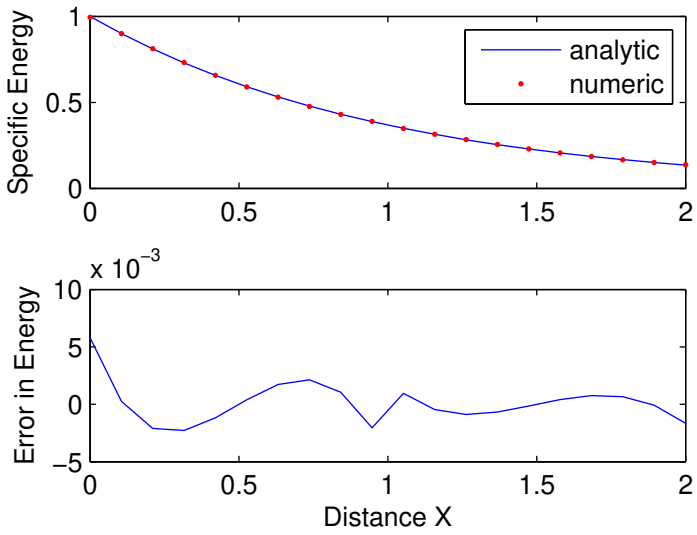


Figure 3-6. – Comparison of exact and interpolated energies from the Gauss Points.

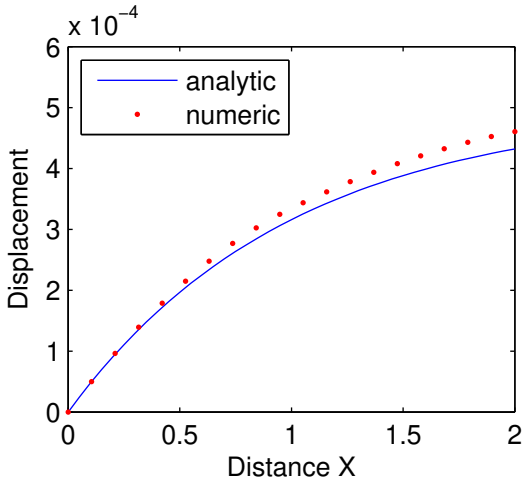


Figure 3-7. – Exponential Energy Deposition, computed Displacements. The numerical results are measured at Gauss points and interpolated within the elements. Displacements are interpolated from nodal values.

.

3.5.4. Two Element, Two Material Hex20

The same geometry is used, but with two different materials for the Hex20 elements. Temperature is specified as a linear function of X. The computed specific energy, $\tilde{E} = C_v T$, is consistent with

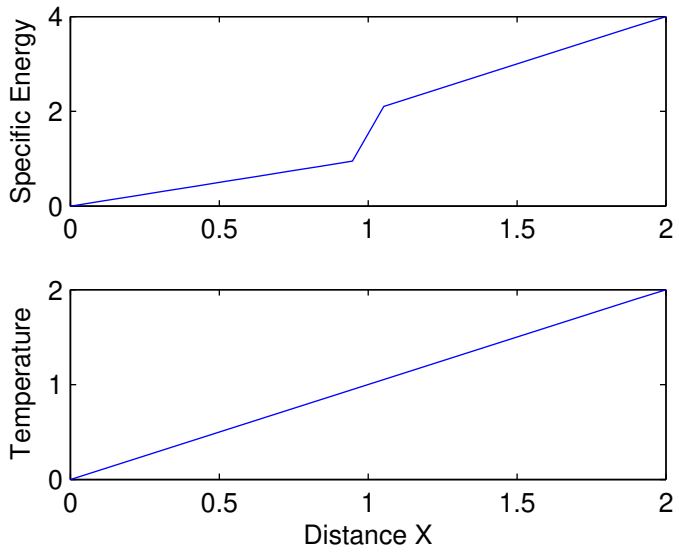


Figure 3-8. – Sampled specific energy and temperature deposition on two blocks.

the temperature. This provides a simple solution for the quadratic displacement. The specific energy is shown in Figure 3-8, as extracted from line sample (`linesample` . The resulting quadratic displacement (and corresponding analytic solutions) is shown in Figure 3-9. For these solutions, the heat capacity is 1 in the first element, and 2 in the second.

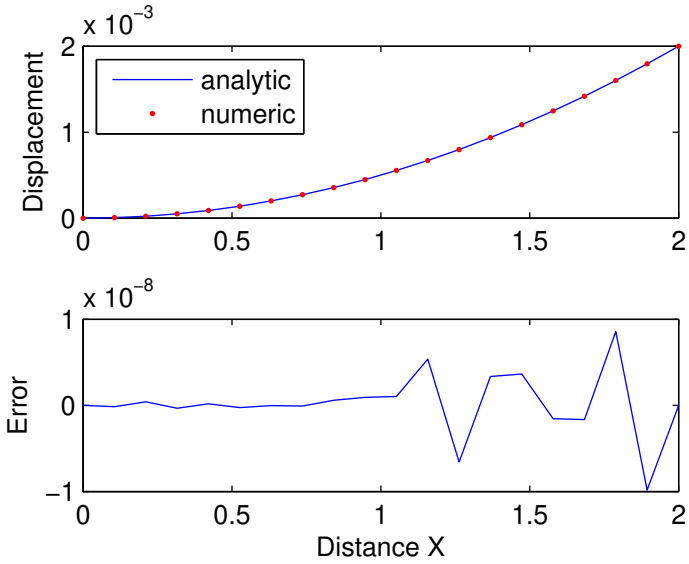


Figure 3-9. – Linear Energy Deposition. The displacement response and associated error is shown.

This page intentionally left blank.

4. ELEMENTS

4.1. Beam Elements

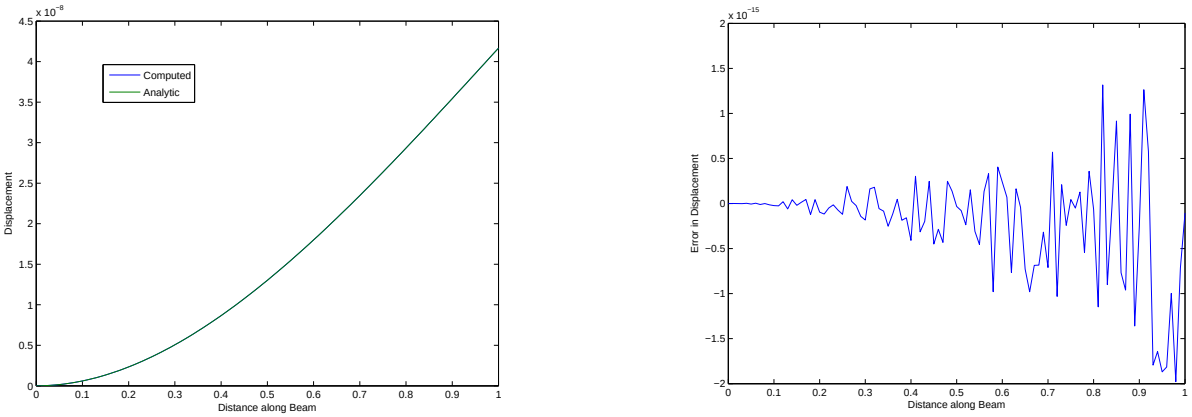


Figure 4-1. – Comparison of Beam2 Bending.

The Beam2 element is a simple Euler Beam. The beam bending equation for a point load, P, on the end of a cantilever beam of length, L, is,

$$w(x) = \frac{Px^2(3L - x)}{6EI}$$

A beam of length $L = 1$, $E = 10e6$, and bending moment $I1 = 0.2$ modeled with 100 Beam2 elements is is compared with the analytic solution in Figure 4-1. Figure 4-2 shows the convergence as a function of the number of elements in the beam. The solutions here are performed with a serial sparse direct linear solver, and with a parallel iterative linear solver with default parameters. The exact solution is a cubic, and is obtained using a one element beam. Thus, increasing the beam count is not required to improve accuracy. The example illustrates both the correctness of the solution for a low element count, and the effect of matrix condition and solver on the solution.

For input deck see Appendix 9.32.

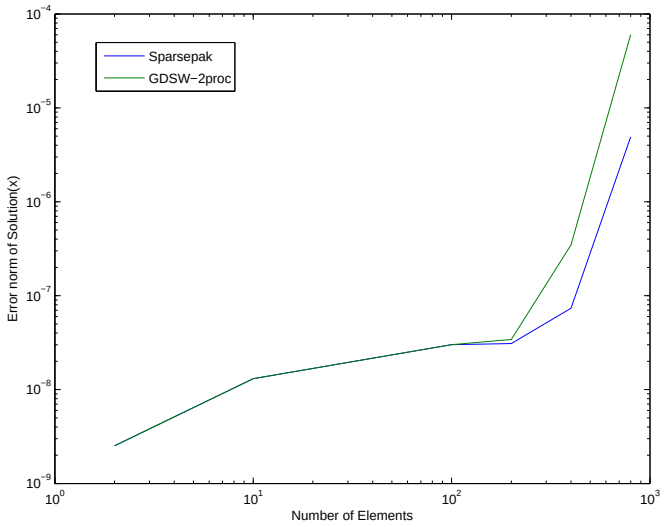


Figure 4-2. – Beam2 Bending "Convergence". The L_2 error norm divided by the L_2 displacement norm is plotted versus the number of elements. Properly convergent solutions would decrease as the number of elements increase. While this solution is accurate, it is not converging to the analytic solution as the number of elements increases.

4.1.1. Comparision with exact solution and NASTRAN

The following test verifies that Sierra SD uses the I1 and I2 properties defined in the input deck in the manner outlined by the user's documentation. The problem was analyzed using a thin long cantilevered beam with a rectangular cross-section made up of 100 Beam2 elements with the following geometry: The beam's cross-sectional properties were chosen to give a very long slender beam with a good separation between bending axes. A MATLAB script was created to calculate the modal frequencies for a single span cantilevered beam using the following formula from Blevins:

$$f_i = \frac{\lambda_i^2}{2\pi L^2}\left(\frac{EI}{m}\right)^{\frac{1}{2}}$$

Here f_i is the natural frequency and λ_i is the natural frequency parameter (Tabular Values). Thysical properties of the beam are quantified by E, I, m, A, and L (standard notation).

Figure 4-3. – Geometry of Beam.

The beam was analyzed using both Sierra-SD and NASTRAN. The NASTRAN results were used as a reference for comparison along with the analytical solution results obtained previously. It is important to note that both the analytical solution and the NASTRAN solution do not calculate twisting modes, while the Sierra-SD model did. These modes were not compared.

Table 4-1. – Beam Cross-Sectional Properties.

Width	0.1	Height	0.3
Length	100	Area	0.03
I1	0.09	I2	0.01

The natural frequencies for all 3 modes [10] are shown in Table 4-2

Natural frequencies that show N/A are twisting modes. Figure 4-4 shows the differences in calculated natural frequencies.

After testing that natural frequencies were in agreement for all three models, the displacements of the Sierra-SD model were compared to the displacements of the NASTRAN model to confirm that the orientations of I1 and I2 were correct. The following Table 4-2 shows the comparison results.

For input deck see Appendix 9.33.

Table 4-2. – Natural Frequency [Hz] results for Analytical, Sierra-SD and NASTRAN models, Displacement Axis Comparison for NASTRAN and Sierra-SD models.

Mode	Analytical	NASTRAN	Sierra-SD	NASTRAN	Sierra-SD
1	0.1022	0.1021669	0.102161	Z-Axis	Z-Axis
2	0.3065	0.3065007	0.306484	Y-Axis	Y-Axis
3	0.6403	0.640269	0.640129	Z-Axis	Z-Axis
4	1.7928	1.792772	1.79205	Z-Axis	Z-Axis
5	1.9208	1.920807	1.92039	Y-Axis	Y-Axis
6	3.5131	3.513118	3.51092	Z-Axis	Z-Axis
7	N/A	N/A	4.90285	N/A	N/A
8	5.3783	5.378316	5.37615	Y-Axis	Y-Axis
9	5.8074	5.807436	5.80229	Z-Axis	Z-Axis
10	N/A	7.905694	7.90561	Z-Axis	Z-Axis

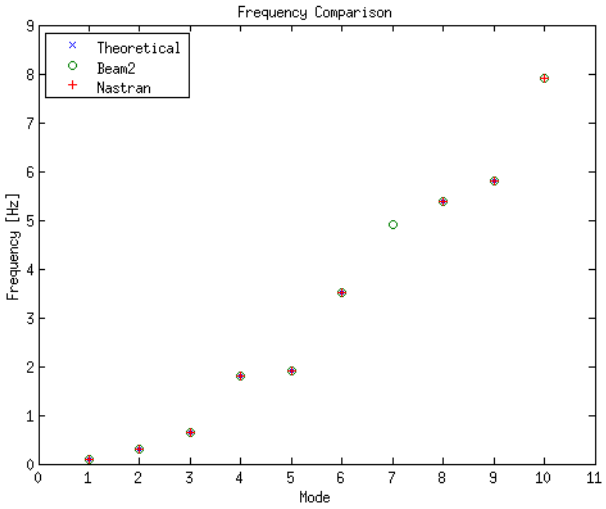


Figure 4-4. – Frequency Comparison.

4.1.2. Beam Element Provided by the Navy

As part of the Navy/CREATE program,[37,40] various elements are being introduced to **Sierra/SD**. These elements fall into two categories: specialty connector element and legacy elements pulled from NASTRAN.

The legacy elements are designed to exactly mimic elements in the NASTRAN capabilities. Typically, these come from the open literature. Because of the nature of these elements, verification is naturally a code to code comparison.

Connector elements are all two node elements provided to enhance special Navy needs. For example, connection of rafts to a hull is best defined using a nonlinear spring dashpot.

The names for all Navy provided elements begins with "N". For example, the navy beam element is the "NBeam".

The **NBeam** is both a connector and a legacy element. The **Beam2** element has most of the same functionality, but does not include offset moments (I12) or shear factors. The static tests included are detailed in Table 4-3. Table 4-4 summarizes some results of the tests. In this section of tests, the NASTRAN results are treated as the truth model. Models were translated using "Nasgen".

Test	Section	Description
btest1	rectangular	simple test of end loaded cantilever
btest2	rectangular	tests rotational invariance
btest3	rectangular	tests beam tower
btest4	channel	tests I_{12}
btest5	channel	rotational invariance of I_{12}
btest6	I-beam	end loaded offset
btest7	rectangular	one element test
btest8	C	offset, rotated C beam

Table 4-3. – Static Tests for NBeam.

Test	Maximum Error
btest1	0.02%
btest2	0.01%
btest3	0.05%
btest4	%
btest5	%
btest6	%
btest7	%
btest8	%

Table 4-4. – Results of Static Tests for NBeam. The maximum error in deflection is shown.

The tests are `Salinas_rtest/test_tool/navy/nbeam/btest1.test`,...,
`Salinas_rtest/test_tool/navy/nbeam/btest8.test`.

To verify that the moments $I1$ and $I2$ specified in an input deck are used as documented, a long thin cantilever beam with a rectangular cross-section was simulated using 100 Nbeam elements.

Figure 4-5. – Geometry of Beam.

Table 4-5. – Beam Cross-Sectional Properties.

Width	0.1	Height	0.3
Length	100	Area	0.03
I1	0.09	I2	0.01

The beam's cross-sectional properties were chosen to give a very long slender beam with a good separation between bending axes.

4.1.2.1. Analytical Solution

A MATLAB script was created to calculate the modal frequencies for a single span cantilevered beam using the following formula [10]

$$f_i = \frac{\lambda_i^2}{2\pi L^2}\left(\frac{EI}{m}\right)^{\frac{1}{2}} \tag{4.1.1}$$

f_i Natural Frequency
λ_i Natural Frequency Parameter (Tabular Values)
E, I, m, A, and L are the usual physical properties of the beam

4.1.2.2. Computational Approach

The beam was analyzed using both Sierra-SD and NASTRAN. The NASTRAN results were used as a reference for comparison along with the analytical solution results obtained previously. It is important to note that both the analytical solution and the NASTRAN solution do not calculate twisting modes, while the Sierra-SD model did. These modes were not compared.

The natural frequencies for all 3 modes are shown in Table 4-6

Table 4-6. – Natural Frequency [Hz] results for Analytical, Sierra-SD and NASTRAN models, Displacement Axis Comparison for NASTRAN and Sierra-SD models.

Mode	Analytical	NASTRAN	Sierra-SD	NASTRAN	Sierra-SD
1	0.1022	0.1021669	0.102161	Z-Axis	Z-Axis
2	0.3065	0.3065007	0.306484	Y-Axis	Y-Axis
3	0.6403	0.640269	0.640129	Z-Axis	Z-Axis
4	1.7928	1.792772	1.79205	Z-Axis	Z-Axis
5	1.9208	1.920807	1.92039	Y-Axis	Y-Axis
6	3.5131	3.513118	3.51092	Z-Axis	Z-Axis
7	N/A	N/A	4.90285	N/A	N/A
8	5.3783	5.378316	5.37615	Y-Axis	Y-Axis
9	5.8074	5.807436	5.80229	Z-Axis	Z-Axis
10	N/A	7.905694	7.90561	Z-Axis	Z-Axis

Natural frequencies marked not available are the twisting modes. Figure 4-6 shows the differences in calculated natural frequencies.

I1 and I2 Verification After testing that natural frequencies were in agreement for all three models, the displacements of the Sierra-SD model were compared to the displacements of the NASTRAN model to confirm that the orientations of I1 and I2 were correct. The following Table 4-6 shows the comparison results.

For input deck see Appendix 9.34.

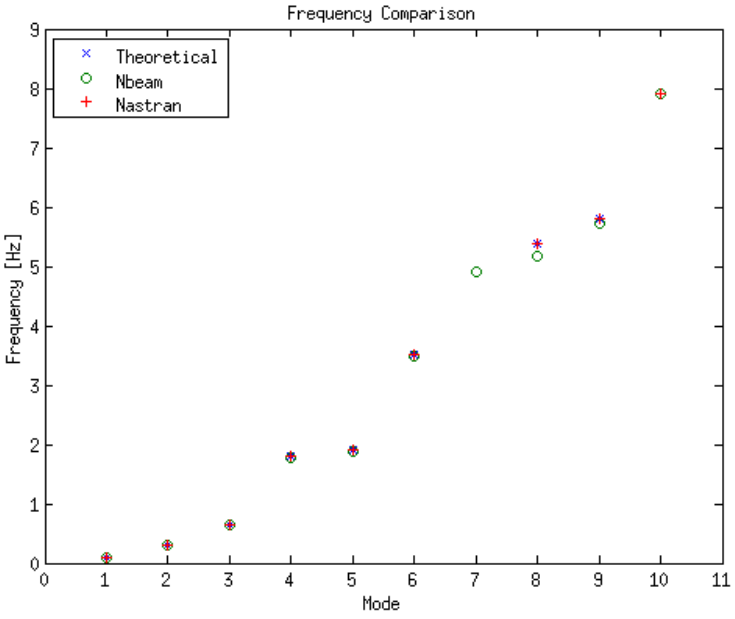


Figure 4-6. – Frequency Comparison.

4.1.3. *Preloaded Beam*

The following test was used to verify that Sierra SD accurately accounts for an axial preload on a beam. This test was verified using three different references, two different analytical solutions [46, 12] and an Abaqus benchmark problem. The problem was first analyzed with no preload using the same analytical solution and then modeled to verify that the system is functioning appropriately.

We use an Abaqus verification problem for the modes of a prestressed beam. A cantilever beam, one hundred elements, is on the x-axis. An axial force is applied in the x direction.

The beam was analyzed with and without the static preload. Figure 4-7 shows the geometry of the model.

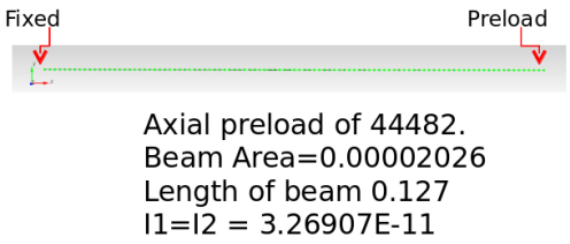


Figure 4-7. – Geometry of Beam.

An equivalent test was created and analyzed in Sierra-SD. The test had three solution cases static, tangent, and eigen analysis. The cantilever beam is partitioned into one hundred beam elements. The frequencies were compared between Abaqus and Sierra-SD and shown in Table 4-7.

Table 4-7. – Results Abaqus vs. Sierra-SD (beam elements).

	Abaqus	Sierra-SD
Without Preload		
Mode 1	212.4	212.818
Mode 2	1330.8	1333.49
Mode 3	3727.2	3733.11
With Preload		
Mode 1	1137.9	1136.8
Mode 2	3624.4	3616.07
Mode 3	6694.1	6667.12

The results are consistent for the benchmark problem. For input deck see Appendix 9.35.

4.1.4. *Prescribed displacement*

The following test was used to verify that Sierra SD accurately accounts for an axial preload on a beam. This test was verified using three different references, two different analytical solutions [46, 12] and an Abaqus benchmark problem. The problem was first analyzed with no preload using the same analytical solution and then modeled to verify that the system is functioning appropriately.

First the solution is compared to the exact solution for Euler beams. All parameters were incorporated using SI units. The beam parameters are:

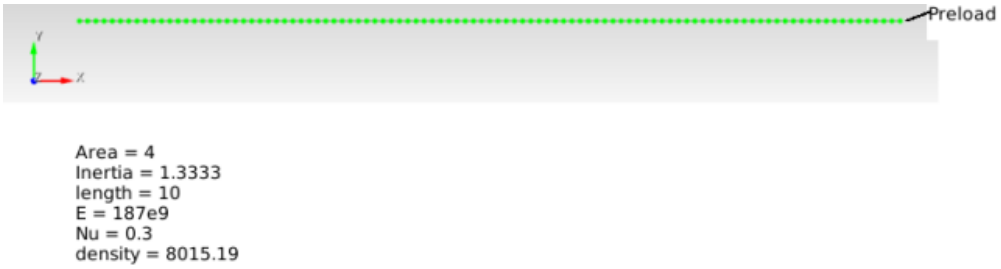


Figure 4-8. – Geometry.

The beam is pinned on both ends (pinned-pinned), with an axial preload in the x direction. This test was analyzed using a tensile and compressive preload.

An axial preload has limited verification due to lack of closed form solutions, however; in the paper [12] an analytical solution can be used. Assuming pinned-pinned constraints on the beam the natural frequencies are:

$$V_n = \left(\frac{n\pi}{L}\right)^2 \left(\frac{EI}{\rho A}\right)^{\frac{1}{2}} \left[1 - \frac{PL^2}{EIn^2\pi^2}\right]^{\frac{1}{2}}$$

n represents the mode number;
P is the axial load;
E, I, ρ, A, and L are the usual physical properties of the beam

A similar analytic solution for non dimensional natural frequency of a pinned pinned beam under axial preload can be found at (Shaker, 1975). Also, a MATLAB file is in the test repository under beam preload verification that solves the two analytical solutions.

4.1.4.1. Computational Approach

This test case was modeled using Sierra SD.The frequencies of a beam under an axial preload require a multicase solution set including static, tangent, and eigen. The static case applies the preload. The tangent case is used following the linear solution step, where the stiffness matrix is

recomputed based on the current value of displacement. Finally, the eigen case is used to output frequencies. The beam was partitioned into one hundred elements.

There are several parameters for this test. To model the beam with pinned-pinned constraints and an axial displacement due to preload the beam was treated with pin roller constraints (where y=0) first. The preload was applied in the x direction at the roller and the max displacement was found. This Max displacement was then used in the pin-pin model as a boundary condition of x at the location of the pin and preload.This extra step needs to be done for a pin-pin case because an axial preload is being applied at the pin where x=0 as a boundary condition, so the beam has zero displacement in the x direction. Also, the length of the beam had the following constraints: z = 0, rotx = 0, and roty = 0. These constraints are used to ensure that the appropriate bending modes are analyzed.

A summary of the results where compared and shown in Table 4-8.

Table 4-8. – Natural Frequency results for Analytical and Sierra SD solution.

	Sierra SD				Analytical			
#	P=N/A	P = 1e3	P = 1e7	P = 1e10	P=N/A	P = 1e3	P = 1e7	P = 1e10
1	43.8041	43.8041	43.8041	51.3605	43.8048	43.805	43.805	51.948
2	175.207	175.207	175.207	181.575	175.219	175.220	175.220	183.905
3	394.18	394.18	394.18	397.775	394.244	394.244	394.244	403.046
4	700.677	700.677	700.677	700.268	700.878	700.878	700.878	709.723
5	1094.63	1094.63	1094.63	1089.04	1095.122	1095.122	1095.122	1103.987
6	1575.96	1575.96	1575.96	1564.04	1576.976	1576.976	1576.976	1585.852
7	2144.55	2144.55	2144.55	2125.15	2146.439	2146.44	2146.44	2155.322
8	2800.29	2800.29	2800.29	2772.26	2803.512	2803.513	2803.513	2812.399
9	3543.03	3543.03	3543.03	3505.24	3548.196	3548.196	3548.196	3557.085
10	4372.62	4372.62	4372.62	4323.94	4380.489	4380.489	4380.489	4389.381

All modes are within 1.5 percent error between the analytical solution and Sierra SD. For input deck see Appendix 9.35.1.

4.2. Membrane Elements

4.2.1. Membrane Quad

A verification test was created for membrane elements in Sierra-SD. The geometry of this test is shown in Figure 4-9.

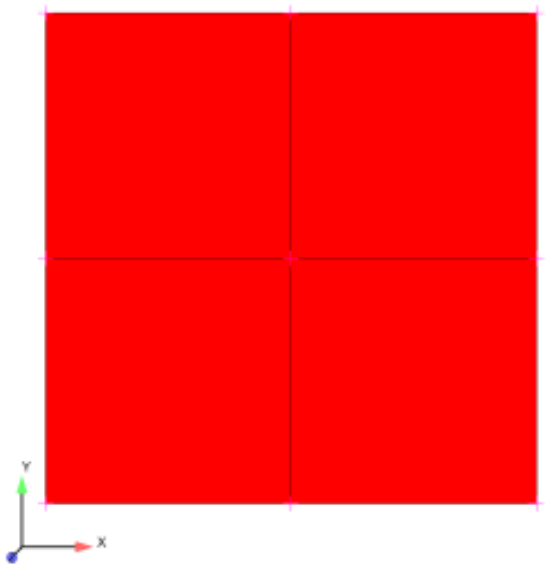


Figure 4-9. – membraneGeometry.

There are four membrane elements in the model with the following boundary conditions. The three bottom and top nodes are fixed in the x and y direction. Th eigenvalue problem is to compute fourteen modes. For verification the test in Sierra-SD was compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 4-9. All modes are compared. There are nine rigid body modes in the model.

For input deck see Appendix 9.36.

Table 4-9. – Sierra-SD and Abaqus Eigenvalue Comparison.

Mode Number	Sierra-SD	Abaqus
1	-6.70788E-09	0.0
2	-6.70788E-09	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	3.7945E-08
7	6.70788E-09	3.7945E-08
8	9.48637E-09	8.8049E-05
9	1.16184E-08	1.1743E-04
10	2607.7	2607.7
11	4237.42	4237.4
12	4723.49	4723.5
13	4723.49	4723.5
14	5164.01	5164.0

4.2.2. QuadM membrane Patch

A patch test that was created for a SierraSD membrane element. The geometry of this test is shown in Figure 4-10. There are five boundary conditions constraining the model. First, all nodes

Figure 4-10. – Patch Test Geometry.

are fixed in the z direction, which is the direction normal to the plane of the model. Second, the top left corner node is fixed in all directions. Third, the nodes on the left side of the geometry are constrained in the x direction. Fourth, the nodes on the top of the geometry are constrained in the y direction. Finally, the nodes on the far right side of the geometry have a prescribed displacement of 0.1 in the positive x direction. The test was analyzed by verifying constant strain throughout the geometry. The results from this test can be seen in Table 4-10.

Table 4-10. – Strain for Membrane Elements.

Node Numbers	Strain
1,4,6,8	Fixed = 0
2,3,5,7,9-13	0.0250

4.2.3. Eigen

The model was also tested using an eigen solution. In this case only the out-of-plane boundary conditions were applied, resulting in a model that should have three rigid body modes. The number of rigid body modes was to be verified in accordance with the boundary conditions. The test case outputs three rigid body modes as expected.

4.2.4. Rotated Patch Test

Further verification was performed using the same patch test by rotating the test out of the XY plane, shown in Figure 4-11.

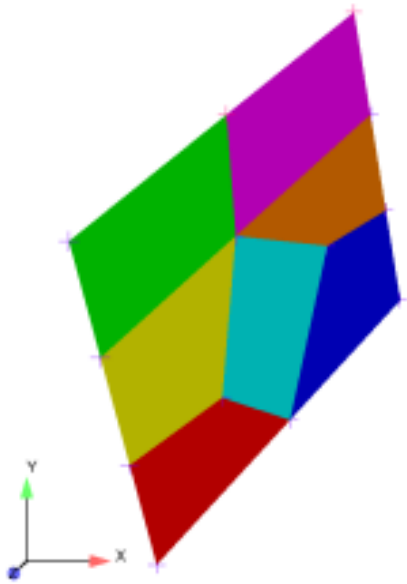


Figure 4-11. – Test Geometry

The model is constrained by MPC's to impose exactly the same boundary conditions as were described in the previous section, except that they were defined with respect to the rotated coordinate system. With these boundary conditions the model has no rigid body modes. The first 10 modes for the rotated test are compared to the in plane patch test. The Eigenvalue results are shown in Table 4-11. As expected, the modes are the same in both cases and are invariant with respect to the rotation of the model.

Table 4-11. – Rotated Patch Test

Mode Number	No-rotation	Rotated
1	627.172	627.172
2	818.997	818.997
3	924.864	924.864
4	1471.59	1471.59
5	1869.91	1869.91
6	2187.29	2187.29
7	2429.53	2429.53
8	2574.91	2574.91
9	2931.04	2931.04
10	3073.42	3073.42

4.2.5. Hex Elements

For verification, the model was also created using the default hex8 elements. The same geometry was used as the membrane element, but the surface was extruded with a thickness of 1. The same

boundary conditions were used as well. The results can be seen in Table 4-12. The strain is constant for every node through out the model, therefore, verifying the patch test is working.

Table 4-12. – Strain for Hex Elements

Node Numbers	Strain
1,2,5,6,10,12-14	Fixed = 0
3,4,,7-9,11	0.0250
15-26	0.0250

4.2.6. Orthotropic Material Properties

In this test, we consider a 2×2 mesh of an orthotropic membrane model where the material elasticity tensor only provides stiffness in the x direction, with zero stiffness in the remaining directions. In addition, we constrain the out-of-plane motion to be zero. With these conditions, we expect 12 rigid body modes, since each of the nodes in the mesh is free to move in the y direction with no resistance. This test involves a coupled Sierra-SM and Sierra-SD analysis, where Sierra-SM produces an output exodus file that contains the necessary material properties. Sierra-SD uses this output exodus file and performs a modal analysis. For verification, the first 18 modes are compared to the Abaqus finite element code. The eigenvalue results are shown in Table 4-13. There are 12 rigid body modes in the model, and the remaining modes show an acceptable comparison of the two codes.

Table 4-13. – Orthotropic Material Patch Test

Mode Number	Abaqus	Sierra-SD
1	0.0000	-3.63305E-03
2	0.0000	-2.86194E-03
3	2.18886E-03	-2.33876E-03
4	4.74120E-02	-9.21049E-04
5	6.70089E-02	9.91374E-05
6	6.70388E-02	5.23966E-04
7	6.70477E-02	9.29529E-04
8	6.70864E-02	1.14456E-03
9	6.71252E-02	1.45159E-03
10	8.20846E-02	1.71789E-03
11	8.20859E-02	2.19313E-03
12	9.47649E-02	2.70663E-03
13	1.08203E+05	1.08184E+05
14	1.53022E+05	1.52995+05
15	1.53022E+05	1.52995+05
16	1.87413E+05	1.87379+05
17	2.16406E+05	2.16367+05
18	2.65042E+05	2.64994+05

The direction of the fibers in the material properties were also changed from the y direction to the x direction. The modes were verified to match exactly and were independent of the fiber direction as expected. For input deck see Appendix 9.37.

4.2.7. Membrane Geometrical Stiffness

We wish to evaluate the geometric stiffness for a simple unit square, with pre-stress in the Y direction. As described in the theory manual, the geometric stiffness is given by,

$$E_g = t \int_A \sigma_{lm} \left[\left(\frac{\partial \delta \boldsymbol{u}}{\partial x_m} \right)^T \frac{\partial \boldsymbol{u}}{\partial x_l} - \frac{1}{2} \sum_{\gamma=1}^{2} \left(\boldsymbol{e}_\gamma \frac{\partial \delta \boldsymbol{u}}{\partial x_l} + \boldsymbol{e}_l \frac{\partial \boldsymbol{u}}{\partial x_\gamma} \right) \left(\boldsymbol{e}_\gamma \frac{\partial \delta \boldsymbol{u}}{\partial x_m} + \boldsymbol{e}_m \frac{\partial \boldsymbol{u}}{\partial x_\gamma} \right) \right] dA \tag{4.2.1}$$

Development Let nodes 1, 2, 3 and 4 have coordinates (0,0), (1,0), (0,1), and (1,1). The shape functions for the nodes are given by

$$\begin{aligned} N_1 &= (1-x)(1-y) \\ N_2 &= x(1-y) \\ N_3 &= (1-x)y \\ N_4 &= xy. \end{aligned}$$

The shape function derivatives are then

$$\begin{aligned} N_{1,x} &= y - 1 \\ N_{1,y} &= x - 1 \\ N_{2,x} &= 1 - y \\ N_{2,y} &= -x \\ N_{3,x} &= -y \\ N_{3,y} &= 1 - x \\ N_{4,x} &= y \\ N_{4,y} &= x \end{aligned}$$

We have

$$\boldsymbol{u} = \sum_{i=1}^{3} (u_{1,i} N_1 + u_{2,i} N_2 + u_{3,i} N_3 + u_{4,i} N_4) \boldsymbol{e}_i, \tag{4.2.2}$$

where $\boldsymbol{e}_i$ is a unit vector in global direction i. We then obtain

$$\begin{aligned} \boldsymbol{u}_{,x} &= \sum_{i=1}^{3} (u_{1,i} N_{1,x} + u_{2,i} N_{2,x} + u_{3,i} N_{3,x} + u_{4,i} N_{4,x}) \boldsymbol{e}_i \\ \boldsymbol{u}_{,y} &= \sum_{i=1}^{3} (u_{1,i} N_{1,y} + u_{2,i} N_{2,y} + u_{3,i} N_{3,y} + u_{4,i} N_{4,y}) \boldsymbol{e}_i \end{aligned}$$

When $\sigma = \sigma_{22}$, and all other components are zero, we can write,

$$E_g = t \int_A \sigma_{22} \boldsymbol{u}_{,y}^T \boldsymbol{u}_{,y} +$$

$$-\frac{1}{2}t\int_A \sigma_{22}\left[(\boldsymbol{e}_1\boldsymbol{u}_{,y}+\boldsymbol{e}_2\boldsymbol{u}_{,x})(\boldsymbol{e}_1\boldsymbol{u}_{,y}+\boldsymbol{e}_2\boldsymbol{u}_{,x})+(\boldsymbol{e}_2\boldsymbol{u}_{,y}+\boldsymbol{e}_2\boldsymbol{u}_{,y})(\boldsymbol{e}_2\boldsymbol{u}_{,y}+\boldsymbol{e}_2\boldsymbol{u}_{,y})\right]dA \qquad (4.2.3)$$

or,

$$\begin{aligned}\frac{E_g}{t\sigma_{22}} &= \int_A \boldsymbol{u}_{,y}^T\boldsymbol{u}_{,y}dA\\ &-\frac{1}{2}\int_A(\boldsymbol{e}_1\boldsymbol{u}_{,y})^2dA\\ &-\int_A(\boldsymbol{e}_1\boldsymbol{u}_{,y})(\boldsymbol{e}_2\boldsymbol{u}_{,x})dA\\ &-\frac{1}{2}\int_A(\boldsymbol{e}_2\boldsymbol{u}_{,x})^2dA\\ &-2\int_A(\boldsymbol{e}_2\boldsymbol{u}_{,y})^2dA\end{aligned}$$

4.2.7.1. Upper left entry

We will examine the 1,1 entry of the stiffness matrix first. This can be found by setting $\boldsymbol{u}_{j,i}=0$ unless $i=j=1$, and $\boldsymbol{u}_{1,1}=1$. This is often called "probing". Then,

$$\begin{aligned}\boldsymbol{u}_{,x} &= N_{1,x}\boldsymbol{e}_1\\ \boldsymbol{u}_{,y} &= N_{1,y}\boldsymbol{e}_1\end{aligned}$$

Then,

$$\begin{aligned}\frac{E_g}{t\sigma_{22}} &= \int_A \frac{N_{1,y}^2}{2}dA\\ &= \int_A \frac{(x-1)^2}{2}dx\,dy\\ &= \left.\frac{(x-1)^3}{6}\right|_0^1\\ &= \frac{1}{6}\end{aligned}$$

4.2.7.2. Other Entries

Computing the remaining terms in the matrix is tedious, but straightforward. A Maple script can be used to accomplish this. From that script, we determine the following.

$$\begin{aligned}K_{11} &= t\sigma_{22}/6\\ Kg_{22} &= -t\sigma_{22}/2\\ Kg_{33} &= t\sigma_{22}/3\\ Kg_{12} &= t\sigma_{22}/8\end{aligned}$$

The maple script is available.

4.2.7.3. Rotations

The test in this directory runs only on a unit square in the xy plane. However, a related verification test that is not included in the code verification manual rotates that structure generally, and compares eigen responses for that rotation with an unrotated square. Having identical eigenvalues assures us that rotations are an issue. For input deck see Appendix 9.38.

4.2.8. QuadS_GY Shear Membrane Shell

The existing Salinas membrane element used for vibration/linear analysis is a quad with three extensional degrees of freedom: u, v, and w. The new shell finite element draws on the Reissner-Mindlin plate theory, as described in Chapter 5 of Ref.[31]. This element has six degrees of freedom per node; three infinitesimal displacements: u, v, w; and three infinitesimal rotations: θ_x, θ_y, and θ_z. Selective reduced integration is used in this bilinear element. Bending and membrane strains are integrated with the 2-by-2 Gauss rule. Shear deformation is integrated with the 1-by-1 Gauss rule. Under integration avoids locking attributed to the shear interpolation. Uncoupled drilling stiffness is added to curb in-plane rotation θ_z. This stiffness is set internally and prevents the solution from containing meaningless null eigenvalues.

Two verification procedures are applied: a) The existing Salinas element QuadT is used to generate reference data; b) Analytical solutions are used. Note that whereas the element QuadT captures only bending, the new QuadS_GY captures bending and shear deformations, in addition to membrane modes. The shell used for verification has dimensions of 1 m by 1 m, the modulus of elasticity is $E = 30\,\text{MPa}$, the Poisson ration is 0.3, and density is 0.288 kg/m^3.

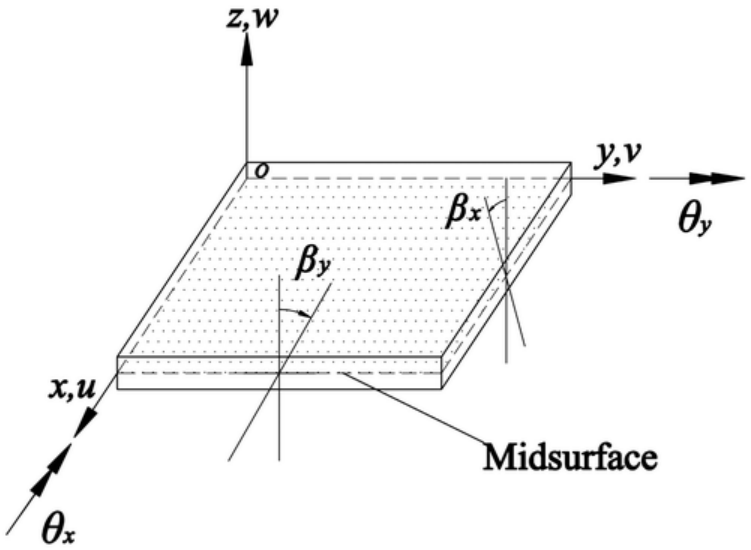


Figure 4-12. – Transverse shear strains β_x and β_y allow cross-sections to not remain on a plate perpendicular to fiber direction. This relaxation of the Kirchhoff hypothesis enables accurate study of thick plates and shells.

4.2.8.1. Eigenvalues of an Isotropic Flat Shell

In this subsection, the behavior of Quad_T (bend. + memb.) and QuadS_GY (bend. + memb. + shear) are compared to bending analytical results (Kirchhoff-Love). A general formula for obtaining the natural frequencies of a flat plate for various boundary conditions is as follows

$$f_{ij} = \frac{\lambda_{ij}^2}{2\pi a^2}\left[\frac{Eh^3}{12\gamma(1-\nu^2)}\right]^{\frac{1}{2}}, \tag{4.2.4}$$

where λ is a parameter that depends on the shell dimensions and its boundary conditions, a is the first dimension of the rectangular shell, E is the isotropic modulus of elasticity, h is the thickness, γ is the mass per unit area of the shell, and ν is the Poisson ration. The λ_{ij} values for specific boundary conditions, relative dimensions, and mode number are given in the literature (see Ref. [10]).

4.2.8.1.1. Fixed-Fixed-Fixed-Fixed (FFFF) The bending eigenfrequencies of the plate for two different thickness values are reported in Tables 4-14 and 4-15. The shear-deformable shell element (QuadS_GY) results naturally diverge from bending theory for increasingly thicker sections.

4.2.8.1.2. Free-Free-Free-Free(FrFrFrFr) Tables 4-16 and 4-17 show natural frequency results of the same plate with the four edges free. Rigid body motion has been disregarded. Only deformation modes are reported in this subsection.

Table 4-14. – Frequencies for FFFF flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
1st mode	347.620	347.466	347.669
	(Ref.)	(0.04)	(0.01)
2nd mode	709.052	708.562	709.363
	(Ref.)	(0.07)	(0.04)
3rd mode	709.052	708.579	709.406
	(Ref.)	(0.07)	(0.05)
4th mode	1046.048	1044.239	1045.507
	(Ref.)	(0.17)	(0.05)
5th mode	1271.098	1270.185	1272.846
	(Ref.)	(0.07)	(0.17)
6th mode	1276.893	1276.245	1278.894
	(Ref.)	(0.05)	(0.15)

Table 4-15. – Frequencies for FFFF flat shell of thickness 0.01 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
hline 1st mode	3476.203	3474.659	3463.921
	(Ref.)	(0.04)	(0.35)
2nd mode	7090.527	7085.620	7048.431
	(Ref.)	(0.07)	(0.60)
3rd mode	7090.527	7085.790	7048.851
	(Ref.)	(0.07)	(0.59)
4th mode	10460.48	10442.393	10361.58
	(Ref.)	(0.17)	(0.94)
5th mode	12710.98	12701.847	12598.886
	(Ref.)	(0.07)	(0.88)
6th mode	12768.93	12762.453	12661.539
	(Ref.)	(0.05)	(0.84)

Table 4-16. – Frequencies for FrFrFrFr flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
1st mode	130.297	129.818	129.919
	(Ref.)	(0.37)	(0.29)
2nd mode	191.147	188.996	189.086
	(Ref.)	(1.12)	(1.08)
3rd mode	235.964	233.438	234.240
	(Ref.)	(1.07)	(0.73)
4th mode	338.251	333.017	335.625
	(Ref.)	(1.54)	(0.78)
5th mode	338.251	335.954	335.756
	(Ref.)	(0.68)	(0.74)
6th mode	594.306	582.394	589.133
	(Ref.)	(2.00)	(0.87)

4.2.8.1.3. Simply supported-Free-Free-Free (SFrFrFr) The natural frequencies associated with the lowest-frequency deformation modes are shown in Tables 4-18 and 4-19.

Table 4-17. – Frequencies for FrFrFrFr flat shell of thickness 0.01 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
1st mode	1302.97	1316.80	1263.69
	(Ref.)	(1.06)	(3.01)
2nd mode	1911.48	2167.89	1938.90
	(Ref.)	(13.41)	(1.43)
3rd mode	2359.65	2353.98	2632.25
	(Ref.)	(0.24)	(11.55)
4th mode	3382.51	3359.54	3331.60
	(Ref.)	(0.68)	(1.50)
5th mode	3382.51	4489.73	3331.81
	(Ref.)	(32.73)	(1.50)
6th mode	5943.06	5891.27	5873.92
	(Ref.)	(0.87)	(1.16)

Table 4-18. – Frequencies for SFrFrFr flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
1st mode	64.212	64.152	64.177
	(Ref.)	(0.09)	(0.05)
2nd mode	145.075	143.874	143.905
	(Ref.)	(0.83)	(0.81)
3rd mode	246.203	244.989	244.650
	(Ref.)	(0.49)	(0.63)
4th mode	252.384	250.912	249.830
	(Ref.)	(0.58)	(1.01)
5th mode	470.480	467.576	467.594
	(Ref.)	(0.62)	(0.61)
6th mode	491.150	488.143	487.013
	(Ref.)	(0.61)	(0.84)

Table 4-19. – Frequencies for SFrFrFr flat shell of thickness 0.01 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
1st mode	642.117	641.523	635.558
	(Ref.)	(0.09)	(1.02)
2nd mode	1450.752	1438.741	1437.167
	(Ref.)	(0.83)	(0.94)
3rd mode	2462.029	2449.891	2426.925
	(Ref.)	(0.49)	(1.42)
4th mode	2523.845	2509.117	2486.897
	(Ref.)	(0.58)	(1.46)
5th mode	4704.803	4675.760	4639.690
	(Ref.)	(0.62)	(1.38)
6th mode	4911.501	4881.430	4841.552
	(Ref.)	(0.61)	(1.42)

4.2.8.2. Eigenvalues of an Orthotropic Shell

For an orthotropic material model, we use a clamped-clamped shell with the following arbitrary orthotropic properties: $E_x = 30\,\text{MPa}$, $E_y = 0.5\,\text{MPa}$, $\nu_{xy} = 0.3$, $G_{xy} = 0.5\,\text{MPa}$, $\rho = 7.46\ \text{g/m}^3$, and thickness is 1 mm. It is assumed that the fiber is aligned with the element frame of reference, i.e., fiber angle $\alpha = 0\,\text{deg}$. Both analytical and QuadT results disregard shear dynamics, whereas shear is present in the computations of the QuadS_GY. The effect of transverse shear tends to be negligible for small relative thickness values. Analytical results are obtained by applying a similar expression to 4.2.4, also provided in Ref. [10]. Results are summarized in Table 4-20. A graphical comparison of the (32) mode for two SD elements is shown in Fig. 4-14.

Table 4-20. – Frequencies for clamped-clamped orthotropic flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytical	QuadT	QuadS_GY
11 mode	209.022	210.144	210.365
	(Ref.)	(0.54)	(0.64)
12 mode	226.154	226.862	227.138
	(Ref.)	(0.31)	(0.43)
13 mode	266.218	266.395	266.738
	(Ref.)	(0.06)	(0.19)
21 mode	572.750	571.523	572.802
	(Ref.)	(0.21)	(0.01)
22 mode	585.382	583.755	585.204
	(Ref.)	(0.28)	(0.03)
23 mode	611.422	609.315	611.004
	(Ref.)	(0.34)	(0.07)
31 mode	1118.82	1115.867	1120.096
	(Ref.)	(0.26)	(0.11)
32 mode	1130.410	1126.535	1131.111
	(Ref.)	(0.34)	(0.06)
33 mode	1152.056	1147.003	1152.097
	(Ref.)	(0.43)	(0.00)

For input deck see Appendix 9.39.

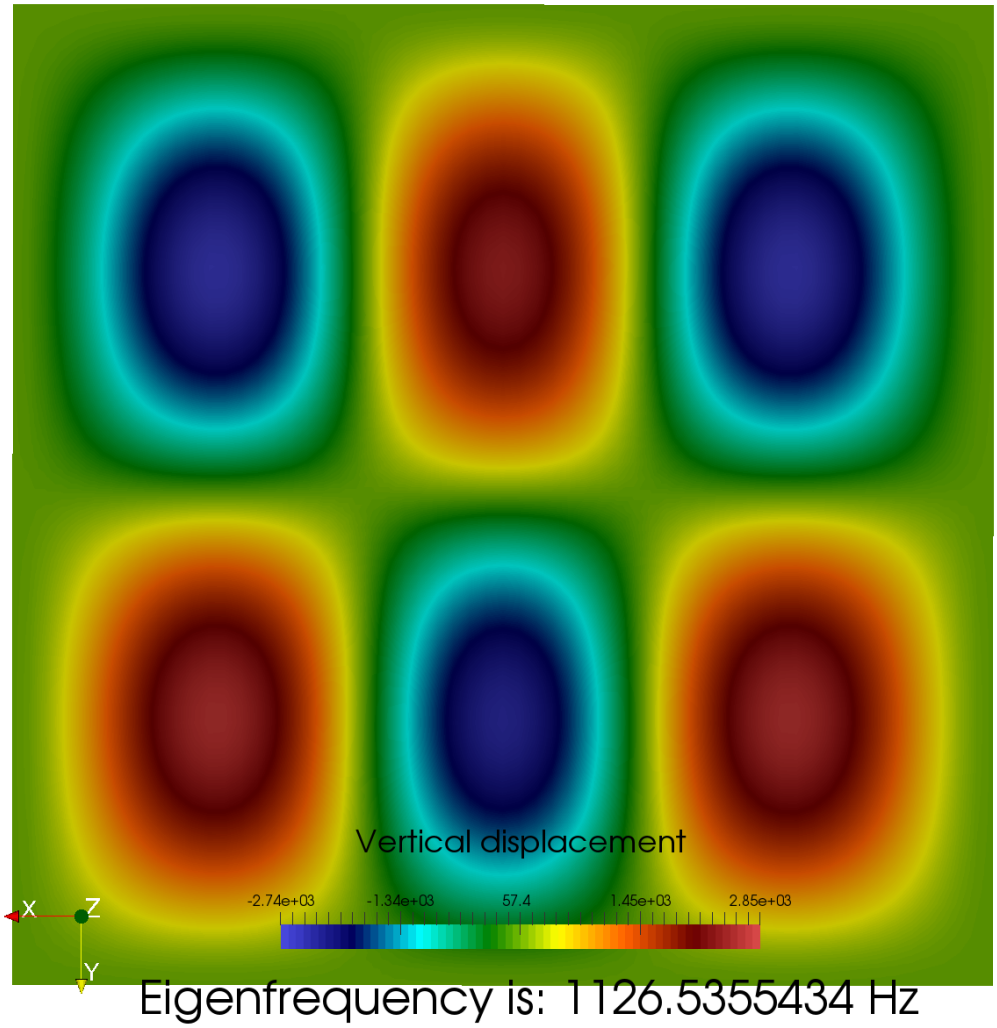


Figure 4-13. – Orthotropic constitutive law QuadT (memb+ bend).

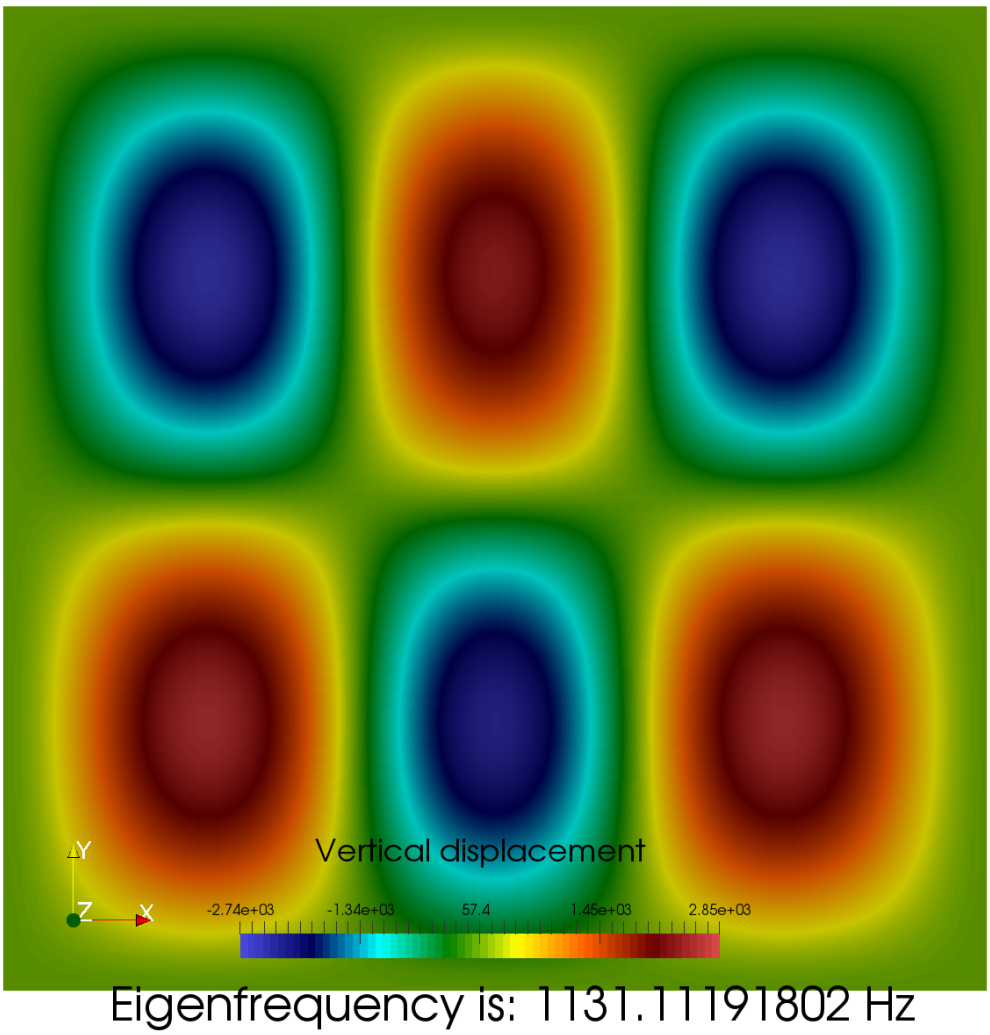


Figure 4-14. – QuadS_GY (memb.+ bend. + shear) Comparison of (32) modes resulting from orthotropic material model (see Table 4-20).

4.2.9. QuadS_GY Shear Membrane Shell - Geometric Stiffness and Preload

A cantilever beam modeled by shear-deformable shell elements used to test other Sierra-SD shell elements is used here too. One end of the beam is clamped. An axial pressure is applied to the other end. The beam is 0.127 m (length) by 0.0044504 m (width) by 0.0044504 m (thickness). The modulus of elasticity is 187 GPa, $\nu = 0.3$, and $\rho = 8015.19\,\text{kg/m}^3$. A linear pressure of -2245852908.28 N/m is applied to the free end, which yields an axial displacement of 1.5656243 mm. The effect of an axial load stiffens the system thus increasing the beam's natural frequencies. The following table summarizes the behavior of the new element:

Table 4-21. – First three natural frequencies of a beam with applied axial pressure.

	Abaqus	SD shell	QuadS_GY	Difference (%)
Without Preload				
Mode 1	212.4	212.793	215.574	1.49
Mode 2	1330.8	1327.73	1345.831	1.12
Mode 3	3727.2	3689.86	3740.46	0.36
With Preload				
Mode 1	1137.9	1141.66	1111.647	2.31
Mode 2	3624.4	3621.86	3536.431	2.42
Mode 3	6694.1	6636.30	6507.385	2.79

Two methods are used to obtain the eigenfrequencies reported in Table 4-21:

- **SD shell**. In Sierra-SD, the pressure load is applied to the shelled beam and, with the resulting displacements, the system stiffness is updated. After that, eigenvalue analysis on the beam is performed considering the updated stiffness.
- **QuadS_GY**. The eigenfrequencies of the preloaded system is computed in a two-step process. First, we applied a prescribed displacement in Sierra-SM to achieve a beam stress state analogous to the SD shell. Then we write those stress to an Exodus output file. This file is used in Sierra-SD to read the geometry of the system and its stresses, which are then used to compute the natural frequencies of the preloaded beam.

This difference in methodology is justified by the way tire modal analysis is performed: First a complex nonlinear system is solved in Sierra-SM. With the resulting stresses, a geometric stress stiffness matrix is built to account for the preloaded state of the tire. Finally eigenvalue analysis is performed in Sierra-SD. Note that the process used for the QuadS_GY shell involves some approximation: Only one integration point is used to carry stresses from Sierra-SM to Sierra-SD, whereas membrane and bending deformation is spatially integrated on a 2-by-2 grid – this may be the reason for the slight discrepancies reported in Table 4-21.

4.2.9.1. Comparison with Sierra-SM

This section compares small deformation results between Sierra-SM and Sierra-SD. For the GY fiber shell, several fiber angles are chosen to verify that element frames of reference and

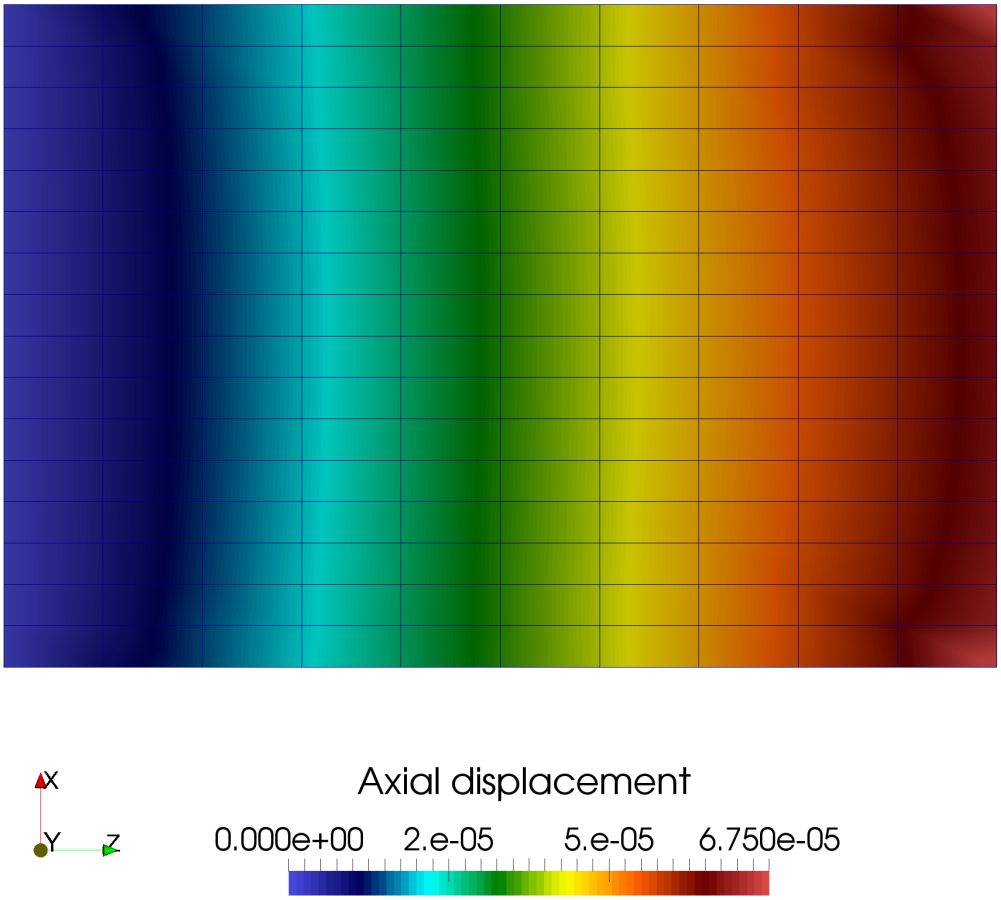


Figure 4-15. – Axial displacement in Sierra-SM

orientation match. A clamped shell on one edge, of dimensions 150 mm by 100 mm is used to compare the displacement results of Sierra-SM and Sierra-SD for small deformation. The shell thickness is 0.4409 m, its modulus of elasticity is 187 MPa, and its Poisson ration, 0.3. One of the short edges is fully clamped and a force of 200 N/node is applied on the other short edge. The same shell is defined in both, the quasistatic nonlinear code Sierra-SM and the linear solver Sierra-SD. Results in terms of axial and lateral displacements may be observed in Figs. 4-16 and 4-18. The axial displacement on the solicited edge center for Sierra-SM is $5.9924 \cdot 10^{-5}$ mm, whereas for Sierra-SD is $5.9908 \cdot 10^{-5}$ mm. Similarly, for lateral displacement, the values are $1.0332 \cdot 10^{-6}$ mm for Sierra-SM, and $1.0409 \cdot 10^{-6}$ mm for Sierra-SD.

For input deck see Appendix 9.40.

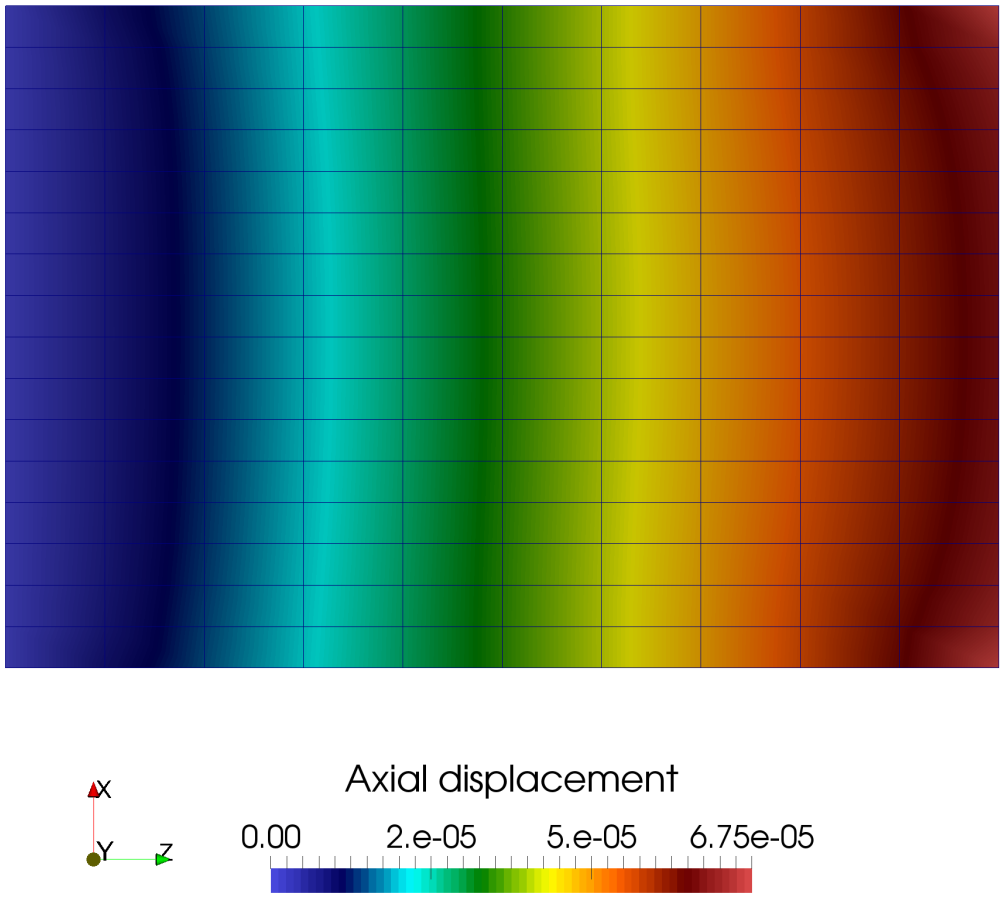


Figure 4-16. – Comparison of axial displacement.

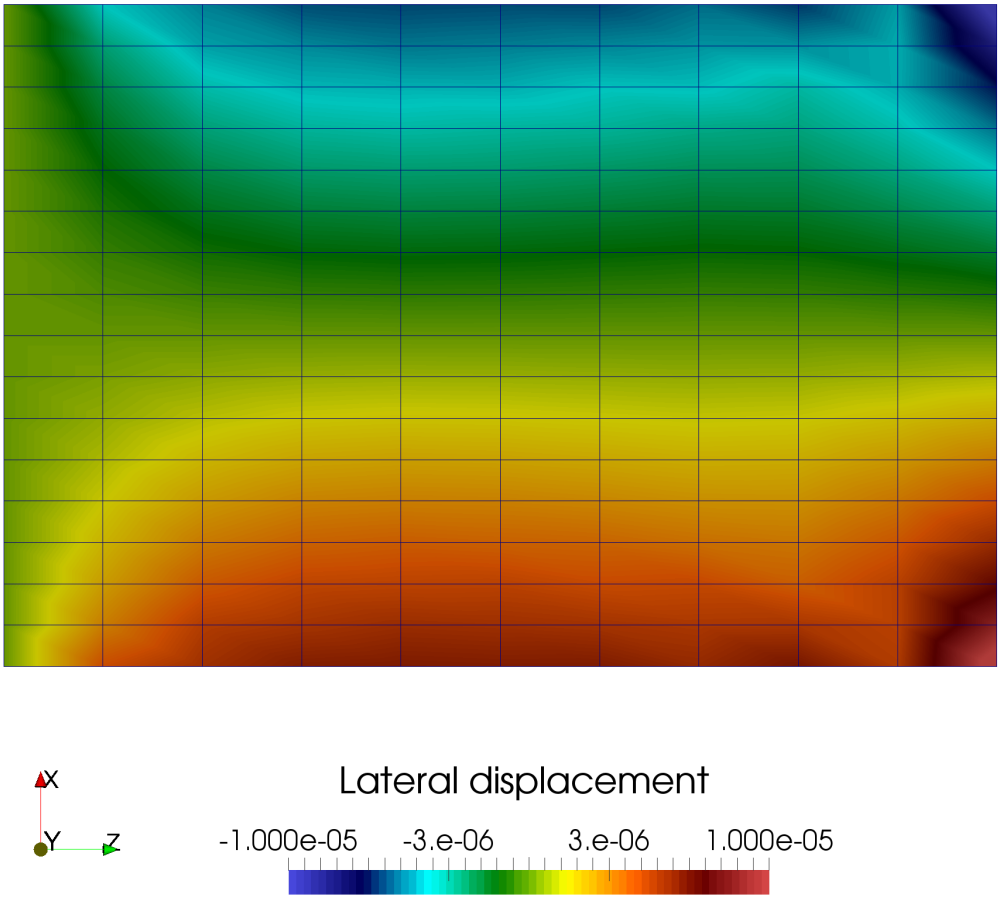


Figure 4-17. – Lateral displacement in Sierra-SM

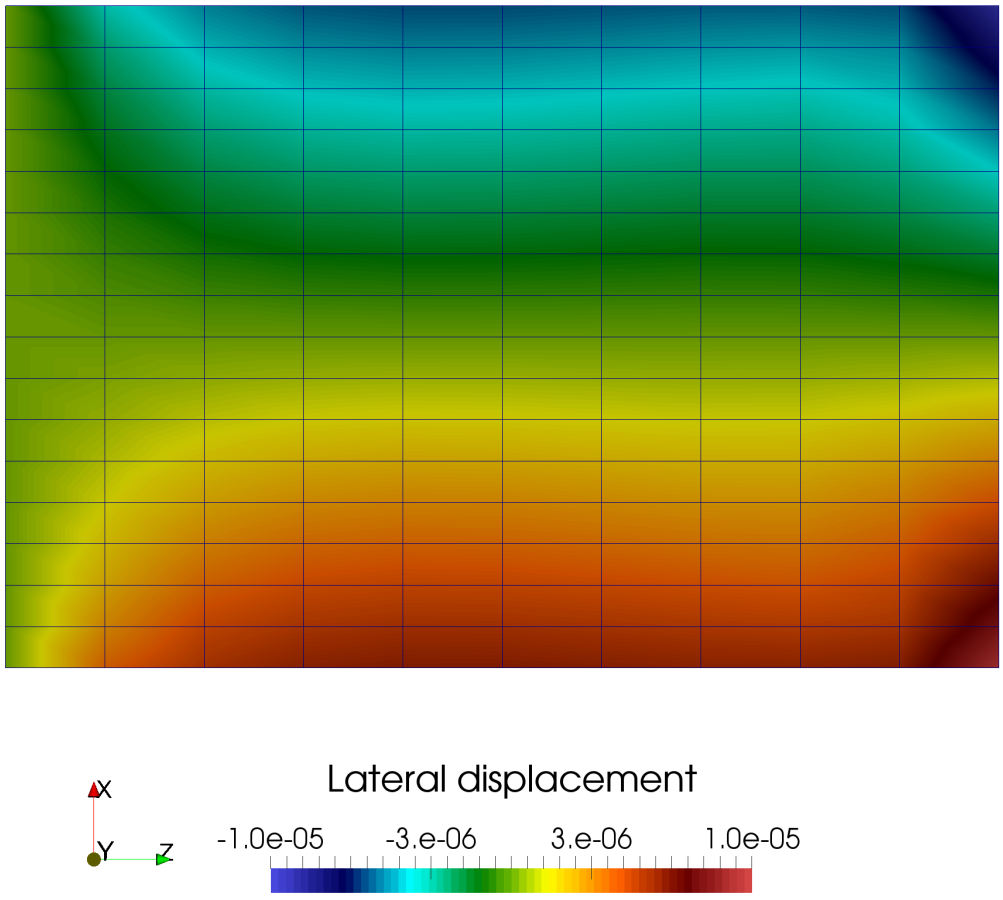


Figure 4-18. – Comparison of lateral displacement.

4.3. Other Shell Elements

4.3.1. Partial Cylinder Patch

This verification example checks the strain output on shell elements. The model is a partial cylinder under axial stretch, with a radius $r = 2.0$, height $h = 1.0$ and thickness $t = 0.01$, shown in Figure 4-19. The material has a Young's modulus of $E = 10^6$ and a Poisson's ratio of $\nu = 0.3$. An axial displacement of $d_{axial} = 0.01$ is applied to the cylinder.

The analytical axial strain and hoop strains are:

$$\epsilon_{axial} = 0.010 \tag{4.3.1}$$

$$\epsilon_{hoop} = 0.003. \tag{4.3.2}$$

The analytical axial stress and hoop stress are:

$$\sigma_{axial} = \epsilon_{axial} * E = 10^4 \tag{4.3.3}$$

$$\sigma_{hoop} = 0.0. \tag{4.3.4}$$

The analytical strain energy density and total strain energy are:

$$SE_{density} = 0.5 * \sigma_{axial}\epsilon_{axial} = 50 \tag{4.3.5}$$

$$SE = SD_{density} * \frac{2ht\pi * r}{4} = 1.570754. \tag{4.3.6}$$

Post processing scripts are used to transform the shell strain results to the hoop and axial directions. Special care has been taken to ensure that the mesh is general, and to verify strain output for arbitrary shape elements. Figure 4-20 shows the axial strain for each element type. Figure 4-21 shows the strain energy density for each element type. Figure 4-22 shows the axial stress for each element type. For input deck see Appendix 9.41.

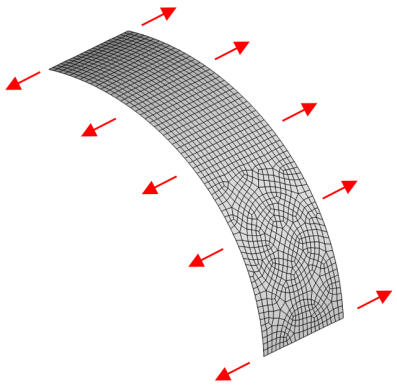


Figure 4-19. – Partial Cylinder under Axial Stretch.

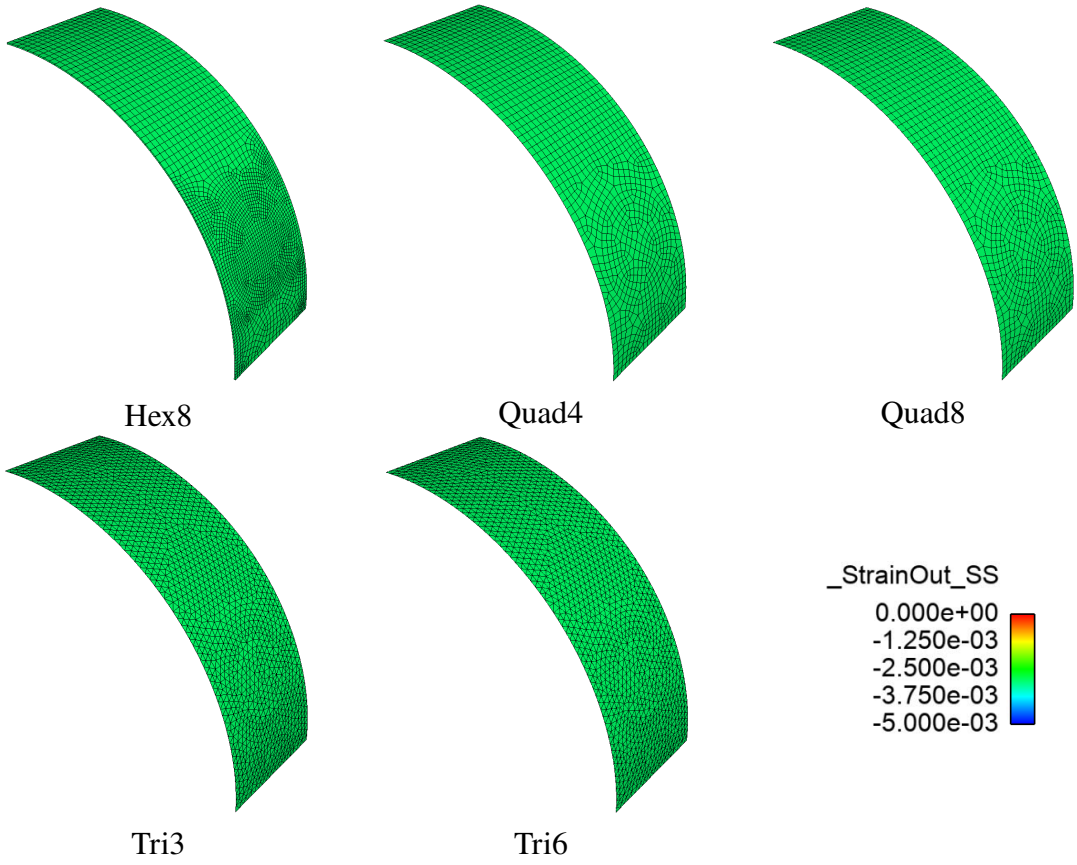


Figure 4-20. – Axial Strain for Partial Cylinder.

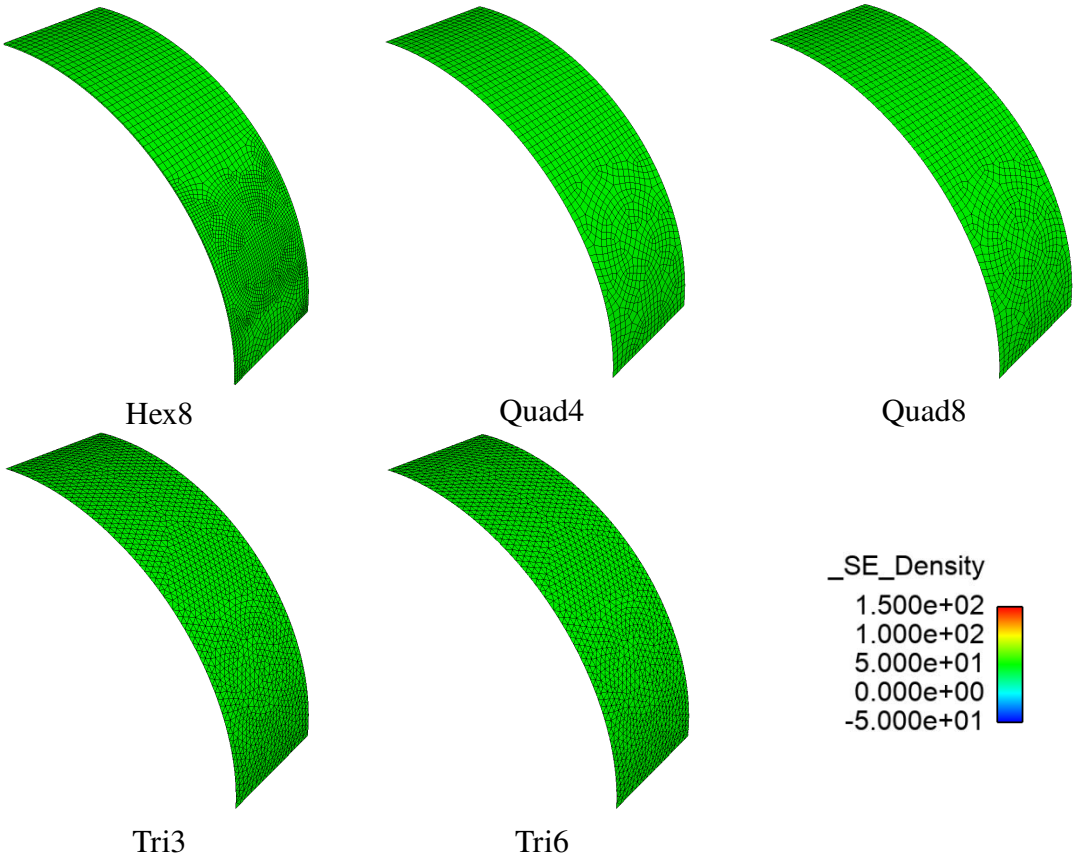


Figure 4-21. – Strain Energy Density for Partial Cylinder.

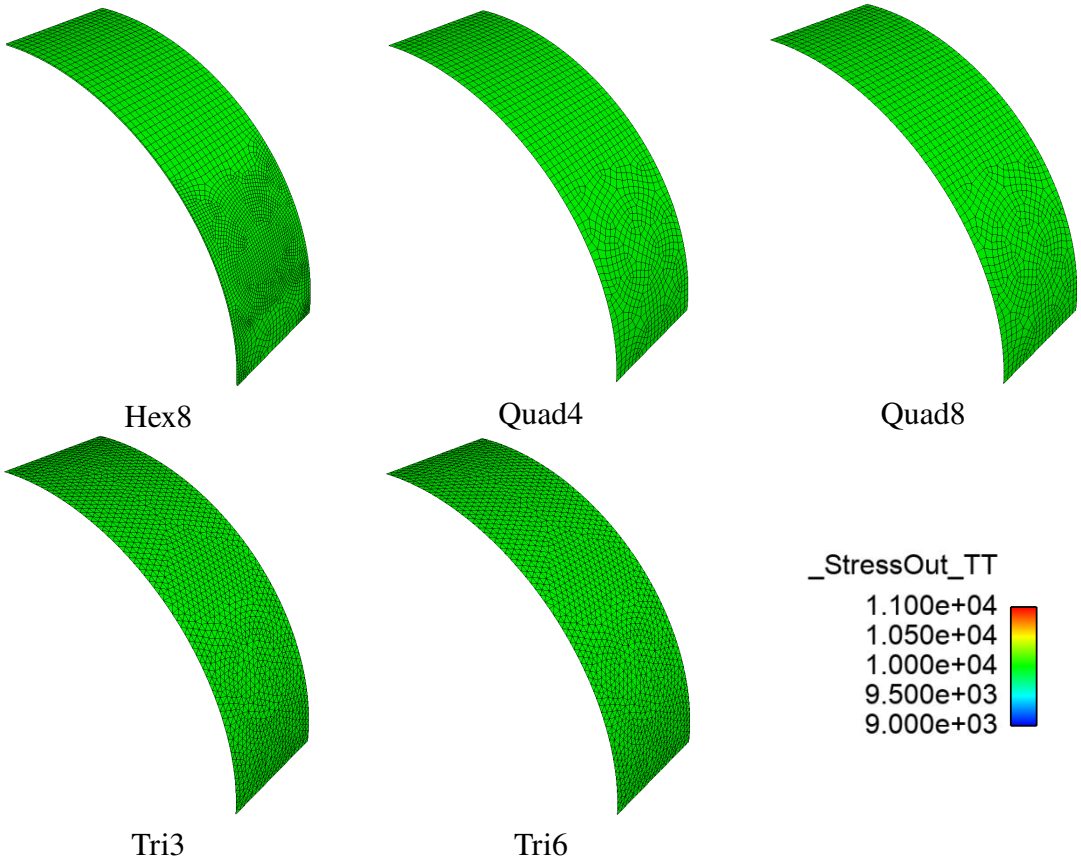


Figure 4-22. – Axial Stress for Partial Cylinder.

4.3.2. *Thin Plate Bending*

The model, shown in Figure 4-23, is a flat rectangular plate of dimension 48 by 24 by 0.5. The normal is in the Z coordinate direction. A uniform pressure is applied to the plate. The thin plate approximation of the maximum displacement is available [53], and should be valid here. The edges are clamped (no rotations for translations).

Table 4-22 compares the solutions from various methods and elements for this example.

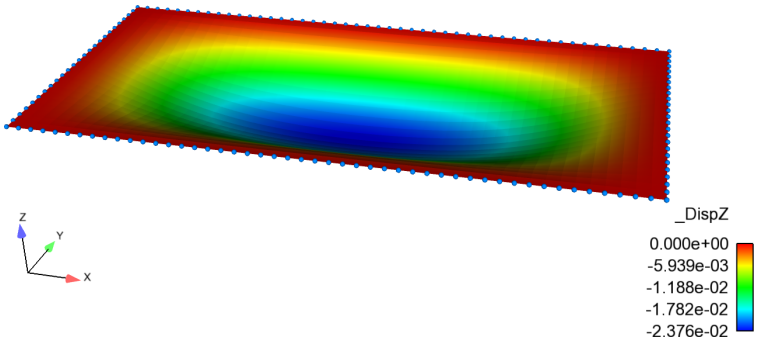


Figure 4-23. – Thin Plate Bending. Geometry and Deformation.

Roark	NASTRAN	%error	NQuad	%error	QuadT	%error
0.02451	0.02459	-0.33	0.02376	3.05	0.024497	0.05

Table 4-22. – Thin Plate Bending Center Point Solutions.

For input see Appendix 9.42

4.3.3. Two Layered Hexshell

This example demonstrates that the automatic verification documentation is viable. Static analyses of a sequence of layered plates problems are solving using the hexshell element

Analysis Type	Statics
Element Type	Hexshell
Dimensions	$[-1/2, 1/2] \times [-1/2, 1/2] \times [-5/2, 5/2]$
Keywords	layered

The example is a step in a study of deflection versus layer thickness in a $[-1/2, 1/2] \times [-1/2, 1/2] \times [-5/2, 5/2]$ brick. Results have been compared to documented results [23],[22] in the past. For input deck see Appendix 9.43.

4.4. Hex Membrane Sandwich

4.4.1. Isotropic Material

A simple plate model was constructed and analyzed using hex and membrane elements, shown in Figure 4-24.

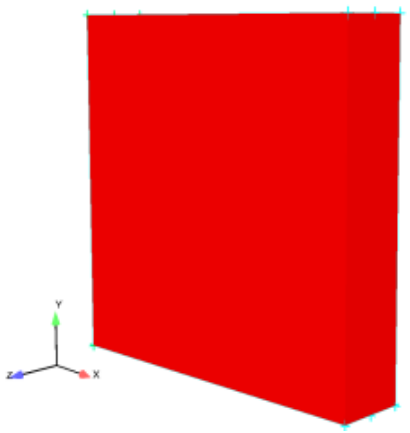


Figure 4-24. – Test Geometry.

The first test using this plate model had no preload. It consisted of isotropic membrane elements sandwiched in between hex elements. The model is fixed on one end and constrained in the Y and Z direction on the other end. The Eigenvalue results are shown in Table 4-23.

Table 4-23. – Isotropic-Nopreload.

Mode Number	Abaqus	Sierra-SD
1	1472.5	1472.46
2	1994.5	1994.48
3	5231.2	5231.19
4	6787.4	6787.39
5	8958.0	8957.96
6	11674.0	11674.2

For a preloaded model, this test was stretched with large deformations in Sierra-SM and a representative Exodus file was outputted. This Exodus file was used in Sierra-SD for a subsequent eigen analysis. For verification, all modes were compared to the Abaqus finite element code. As in the first case, the plate is fixed on one end and is constrained in the Y and Z direction on the other end. The Eigenvalue results are shown in Table 4-24.

Table 4-24. – Isotropic-Preload.

Mode Number	Abaqus	Sierra-SD
1	1420.8	1410.79
2	1798.3	1808.77
3	5212.8	5208.10
4	6765.5	6765.63
5	8914.0	8911.89
6	11638	11636.50

4.4.2. Orthotropic Material

An orthotropic material test was performed using the same plate model. The material elasticity tensor only provides stiffness in the x direction, with zero stiffness in the remaining directions. Initial tests had no preload. The modal results are shown in Table 4-25.

Table 4-25. – Orthotropic-Nopreload.

Mode Number	Abaqus	Sierra-SD
1	4776.10	4772.99
2	5231.20	5231.19
3	8152.20	8149.91
4	8958.00	8957.96
5	10998	10970.90

For the second test, the same model was used, and an uniaxial preload in the x-direction was applied using **Sierra/SM**. An output Exodus file was then passed to Sierra-SD for the modal analysis. For verification, all modes were compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 4-26.

Table 4-26. – Orthotropic-Preload.

Mode Number	Abaqus	Sierra-SD
1	4600.30	4451.72
2	5212.80	5208.10
3	7821.60	7919.50
4	8914.00	8911.89
5	9878.40	9227.89

For input deck see Appendix 9.44.

4.5. Higher Order Hex Acoustic Element Convergence

For the phex elements of order up to 4, the convergence rates demonstrate the expected values in the limit of small enough element size.

The geometry of the model is shown in Figure 4-25. It consists of an acoustic waveguide of length $L = 10.0(m)$, and cross-sectional dimensions of $1.0(m)$. The walls were assigned as rigid around the boundaries of the waveguide, including the endcaps. The speed of sound was given as $c = 332.0\frac{m}{s}$. With these parameters, the exact frequencies of vibration of the air in the waveguide are given as

$$f_n = \frac{nc}{2L} = 16.6, 33.2, \ldots \tag{4.5.1}$$

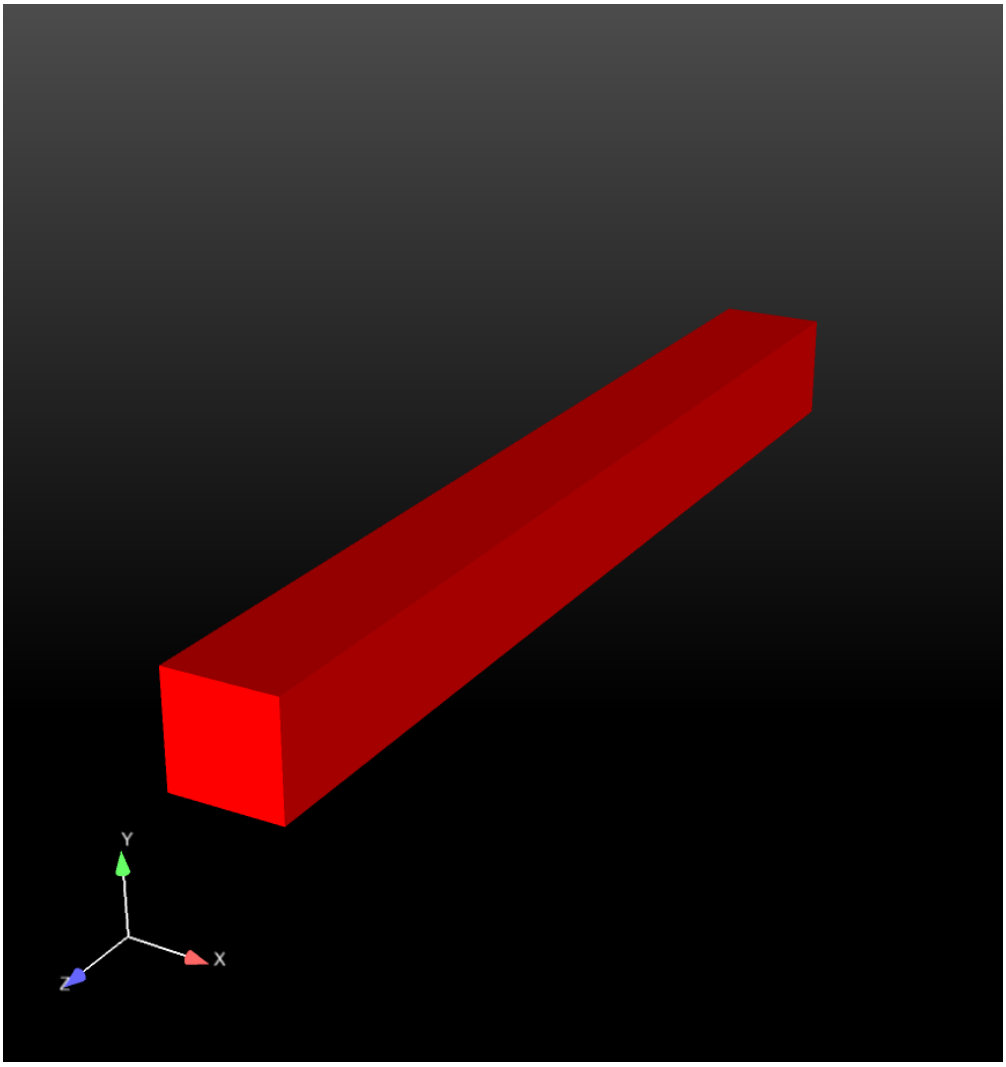


Figure 4-25. – Waveguide Model for Convergence Study of P-hex elements.

Figure 4-26 shows the convergence plot for the hex element for orders 2 – 4. The theory predicts that the modal frequencies should converge at a rate of h^{2p}, where h is the element size, and p is

the order. Thus, on a log-log plot, the slopes of the convergence lines should be 4, 6, and 8, respectively. In Figure 4-26 we show the relative errors in the 10^{th} modal frequency. Similar results were obtained for the other modes, and so we only show the 10^{th} modal frequency for brevity. In addition to the errors, we show lines that have slopes of 4, 6, and 8, respectively for comparison with the error curves. As seen, for each order, the correct slope is obtained in the limit of small h, (or large $\frac{1}{h}$).

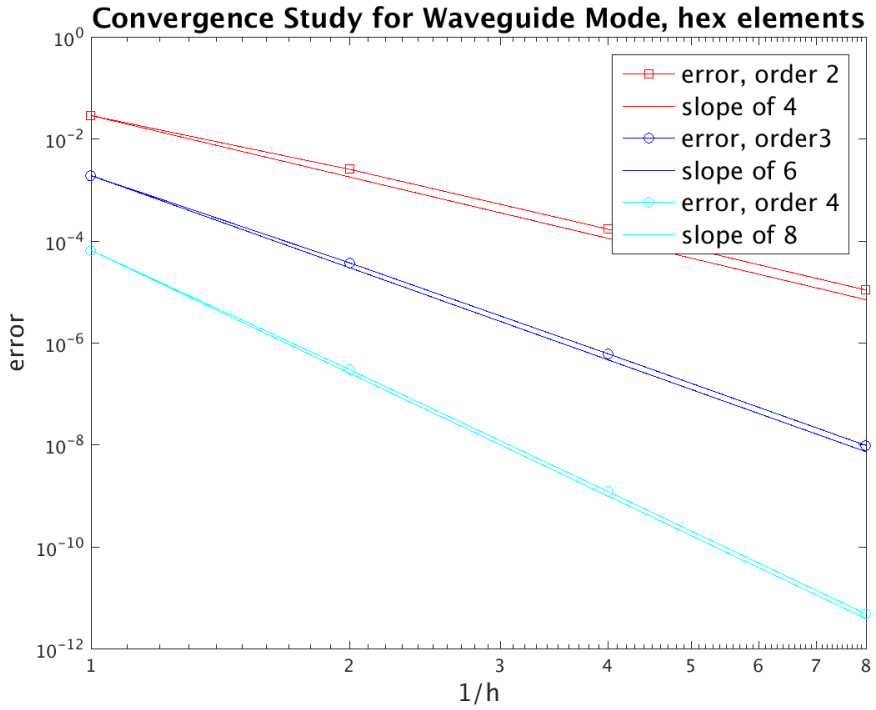


Figure 4-26. – Convergence Study of P-hex elements.

For input deck see Appendix 9.45.

4.6. Higher Order Tet Acoustic Element Convergence

This section demonstrates a convergence study for the ptet element, up to order 4. We verify that the convergence rates approach the theoretically predicted ones in the limit of small enough element size.

The geometry of the model is shown in Figure 4-27. It consists of an acoustic waveguide of length $L = 10.0(m)$, and cross-sectional dimensions of $1.0(m)$. The walls were assigned as rigid around the boundaries of the waveguide, including the end caps. The speed of sound was given as $c = 332.0\frac{m}{s}$. With these parameters, the exact frequencies of vibration of the air in the waveguide are,

$$f_n = \frac{nc}{2L} = 16.6, 33.2, \ldots \tag{4.6.1}$$

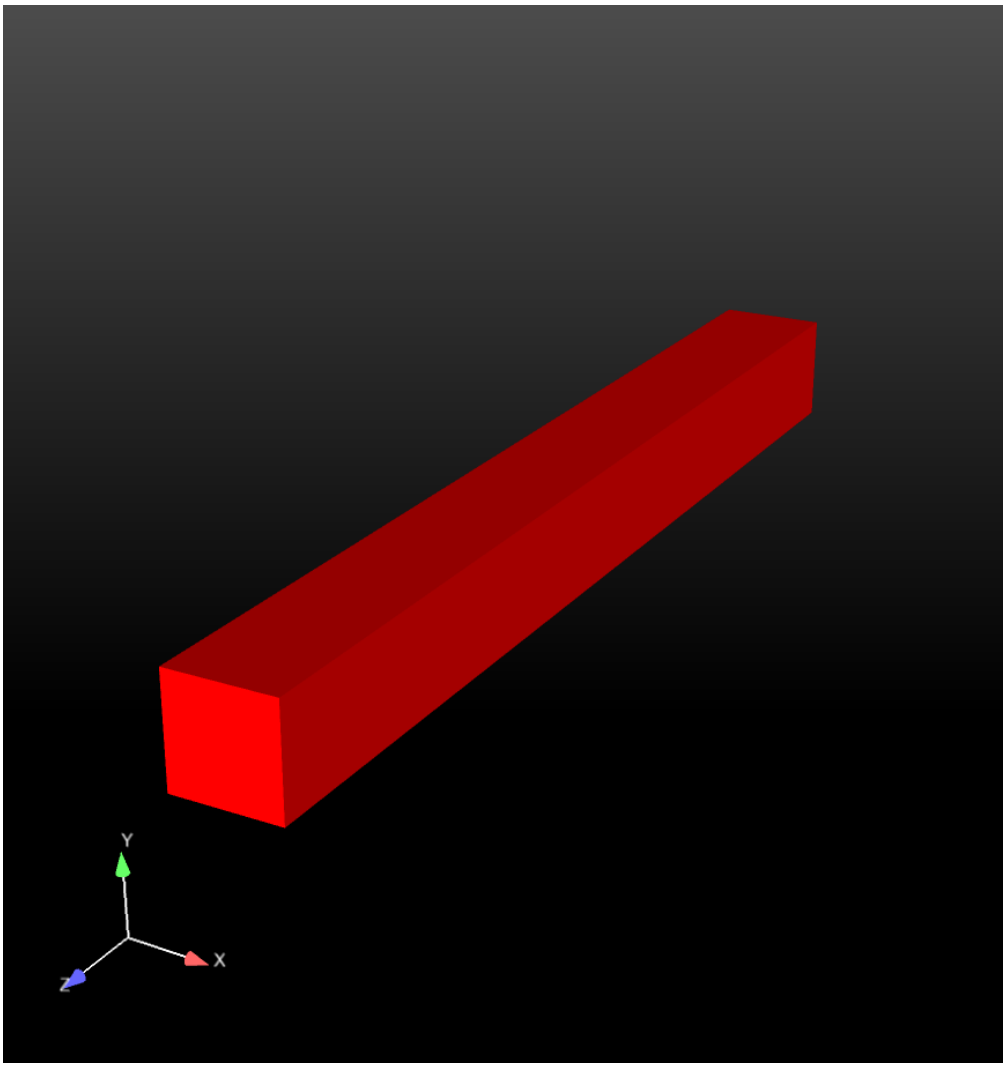


Figure 4-27. – Waveguide Model for Convergence Study of P-tet elements.

Figure 4-28 shows the convergence plot for the tet element for orders 2 – 4. The theory predicts that the modal frequencies should converge at a rate of h^{2p}, where h is the element size, and p is

the order. Thus, on a log-log plot, the slopes of the convergence lines should be 4, 6, and 8, respectively. In Figure 4-28 we show the relative errors in the 10^{th} modal frequency. Similar results were obtained for the other modes, and so we only show the 10^{th} modal frequency for brevity. In addition to the errors, we show lines that have slopes of 4, 6, and 8, respectively for comparison with the error curves. As seen, for each order, the correct slope is obtained in the limit of small h, (or large $\frac{1}{h}$).

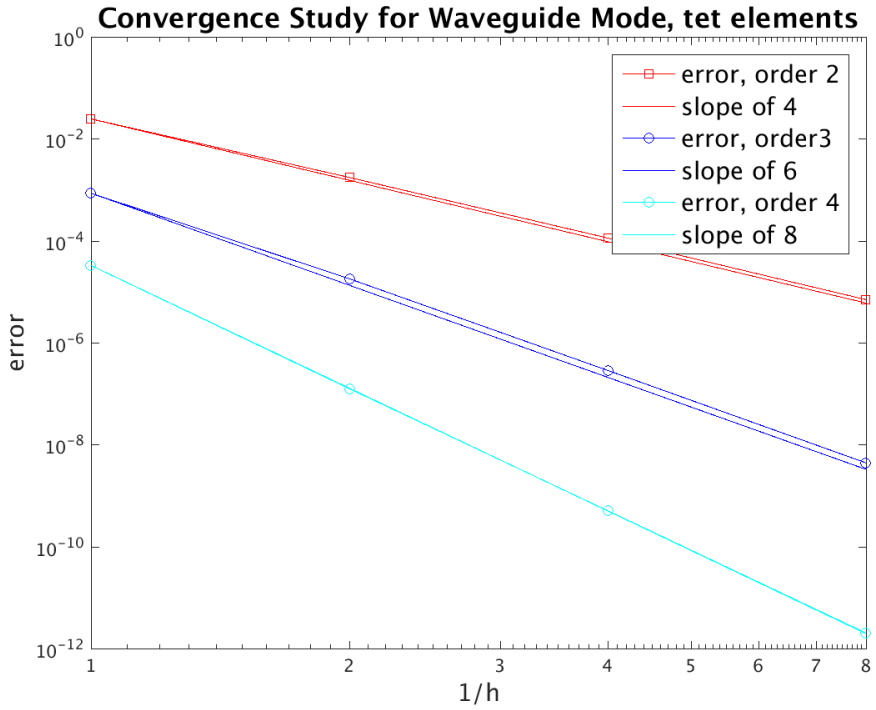


Figure 4-28. – Convergence Study of P-tet elements.

For input deck see Appendix 9.46.

4.7. Superelements

4.7.1. Damping

A superelement can have block proportional damping in Sierra SD. [1] A model was created consisting of two steel blocks acting as a cantilever beam. To incorporate block proportional damping into a system two parameters may be used, `blkalpha` and `blkbeta`. Blkalpha is mass proportional damping and blkbeta is stiffness proportional damping. For this model stiffness damping has the largest impact on the system. The damping parameters are set low enough for energy to enter block two, but high enough to absorb energy. A pressure load is applied on the top surface of block 1. A transient analysis is run with and without superelements and compared. Block 2 is reduced to a superelement and contains block proportional damping. The damping parameters for the CBR solution are set in the block section of the input deck. Figure 4-30

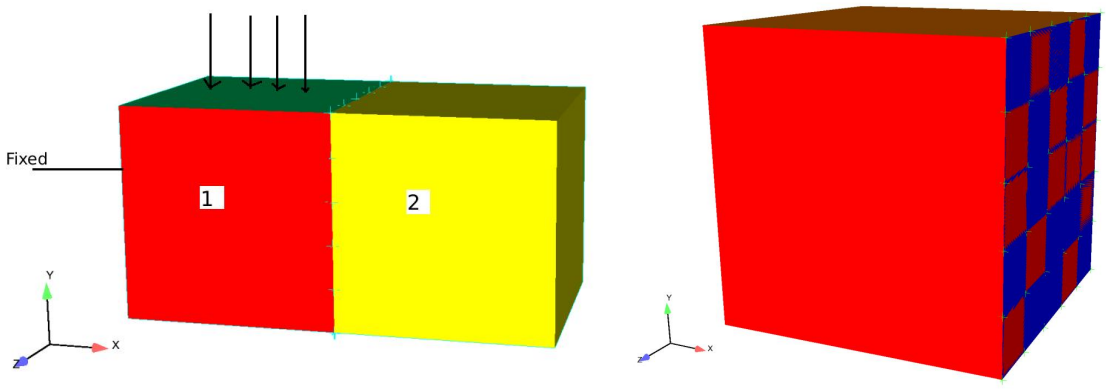


Figure 4-29. – Initial model and model with superelement.

consists of three curves including the undamped full system solution, the damped solution with no superelements, and the damped solution with superelements. The damped model with superelements traces the damped model without superelements well. A full convergence study was not preformed as the two damped models will not match perfectly due to model truncation. For input deck see Appendix 9.47.

[1] System proportional damping does not create a damping matrix and cannot be used to generate a reduced order damping matrix.

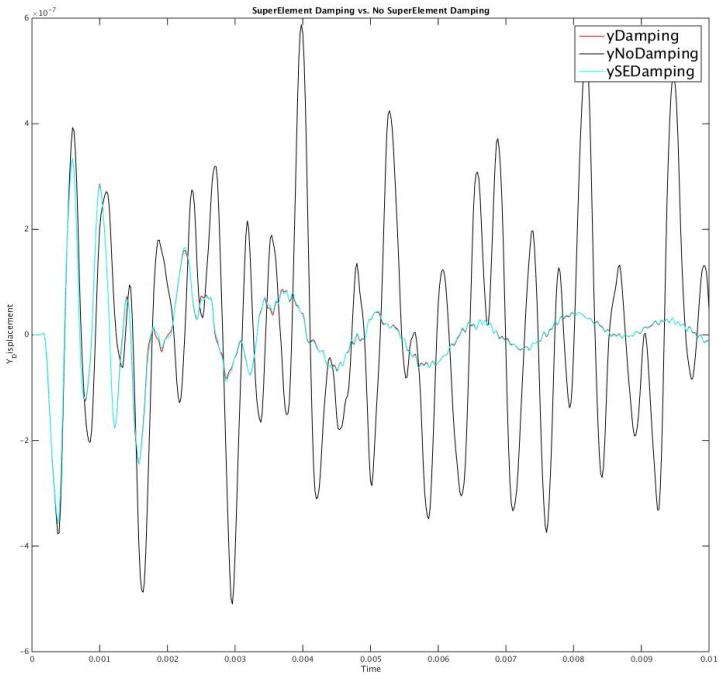


Figure 4-30. – Superelement Damping Results. Damped and undamped response of full system models compared with damped model of the reduced order model.

4.7.2. Sensitivity Analysis

The goal of the test is to verify both generation of a Craig-Bampton (CB) reduction and associated matrix sensitivities as well as the Taylor series expansion of the resulting matrices to generate a point evaluation of a parameter. In this example, a more complex model is evaluated with two parameters. While the geometry of the model is more complex the structure still is linear in the parameters of interest.

4.7.2.1. Blade Model

The model is shown in Figure 4-31. The full model (including superelement and residual structure) is shown on the left. The next cut away shows only the residual structure in gray. A portion of that model is provided only for visualization. On the right is the model of the superelement which consists of quadrilateral and triangular shells. The interface nodes are in red. Analysis is performed in two stages. First, the CB reduction is performed and sensitivity matrices dKr/dp and dMr/dp are generated. The reduction is performed in two ways: by constant vector, and by finite difference approaches. Following sensitivity analysis and model reduction, a system analysis is performed where those matrices are used in a Taylor series expansion.

For this analysis, we use the material density and Young's modulus as the sensitivity parameters. There are no repeated frequencies, which avoids any issue of mode mixing for finite difference sensitivity.

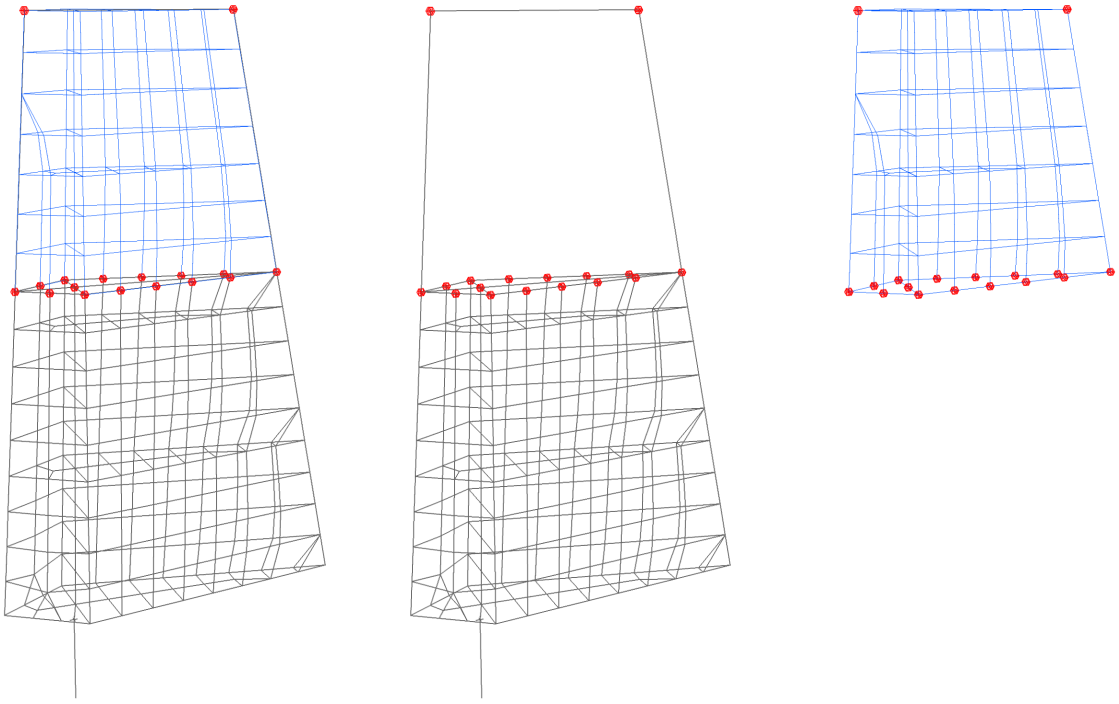


Figure 4-31. – Blade superelement model.

4.7.2.2. Exact and Numerical Results

For a constant vector sensitivity analysis, the reduced order matrices are given by,

$$\tilde{k}_o = T_o^T K(p_o) T_o \tag{4.7.1}$$

$$\tilde{k}_1 = T_o^T K(p_o + \Delta p) T_o \tag{4.7.2}$$

$$\frac{d\tilde{k}}{dp} \approx \frac{\tilde{k}_1 - \tilde{k}_o}{\Delta p} \tag{4.7.3}$$

Here,

T_o	is the transformation matrix evaluated at p_o,
p_o	is the nominal value of the sensitivity parameter,
Δp	is the change of the sensitivity parameter,
$\tilde{k}$	is the reduced stiffness matrix, and
$K()$	is the unreduced stiffness matrix.

Identical relations exist for the mass matrix.

For a **finite difference** sensitivity analysis, the relations are somewhat different.

$$\tilde{k}_o = T_o^T K(p_o) T_o \tag{4.7.4}$$

$$\hat{k}_1 = T_1^T K(p_o + \Delta p) T_1 \tag{4.7.5}$$

$$\frac{d\tilde{k}}{dp} \approx \frac{\hat{k}_1 - \tilde{k}_o}{\Delta p} \tag{4.7.6}$$

Here,

T_1	is the transformation matrix evaluated at $p_o + \Delta p$,

Because T_1 depends on the density and Young's modulus, the reduced stiffness matrix is affected by the transformation.

Numerical Results. Figure 4-32 shows a comparison of the system level solutions as a function of design parameter. We vary the density and Young's modulus together. Three curves are shown. The exact solution shows results obtained by rebuilding the superelement using the parameter, and without sensitivities. The other two curves evaluate dk/dp at the nominal value, and estimate the superelement contribution using a Taylor series expansion. Results are shown for mode 3. A comparison of the error is shown in Figure 4-33

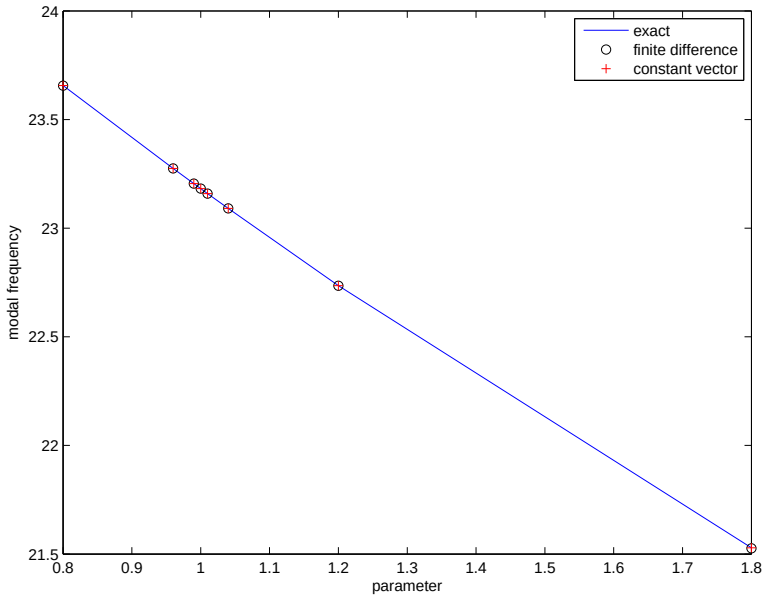


Figure 4-32. – Modal Frequency Variation with Density.

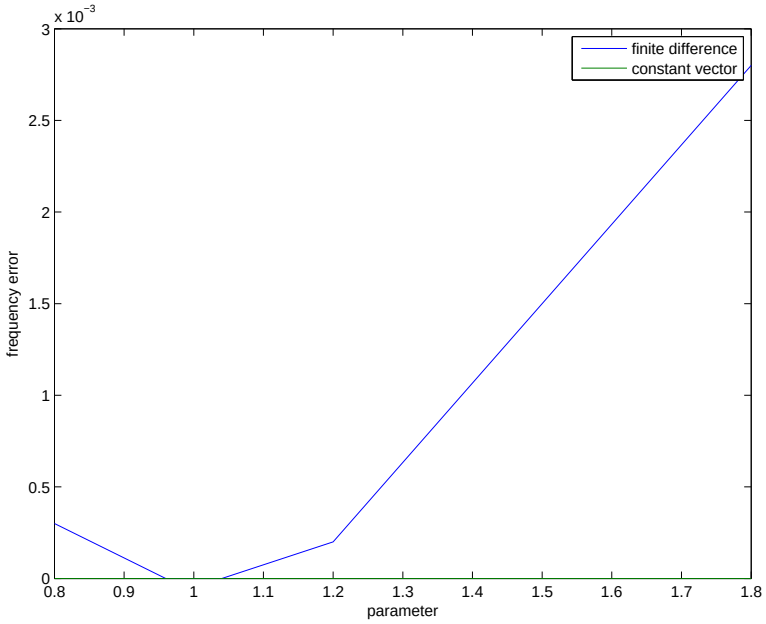


Figure 4-33. – Modal Frequency Error with Density.

4.7.2.3. Summary

These analyses compare results for application of sensitivity matrices to superelement analysis. In this example, for which the superelement matrices vary linearly with the parameter, the constant vector method works extremely well. While not shown here, variations of a single parameter by itself returns very similar results.

One point of interest is that for large variations of the parameter, the finite difference method of computing sensitivities resulted in indefinite matrices that caused the eigenvalue algorithm to fail. For input deck see Appendix 9.48.

4.7.3. Superposition

A four truss, 1-D problem provides a simple verification of Craig-Bampton Reduction (CBR) and superposition based reconstruction. As illustrated in Figure 4-34, the model is clamped on the left end, and constrained to admit only translations in the X direction on the remaining four nodes. A transient load acts in the X direction for this problem, so the model is fully one-dimensional [2]. The verification proceeds as follows.

1. Compute the full system (4 element) static load due to a point load on the center node. This is used as the truth model.
2. Split the model into two pieces, each composed of two elements each. The CBR model is floating in the X direction, where load is applied.
3. Approximate solution uses CBR methods to reduce the last two elements (3 nodes) to two dofs.
4. The "residual solution" computes the system statics solution based on the left-hand side (unreduced) model connected to the CB reduced right-hand side system. Results in the residual are compared with step 1.
5. One output of the system transient solution is `endtruss-out.ncf`. This file contains the modal amplitudes and the interface amplitudes for the superelement. These amplitudes, together with the modal bases computed in step 2 above, provide the information necessary to compute the physical degrees of freedom in the portion of the structure on the right. The model is generated using the "superposition" solution method. This model is then compared with the results from the right-hand portion of the truth model.

Figure 4-35 provides a comparison of the solutions using the full model, and the individual components.

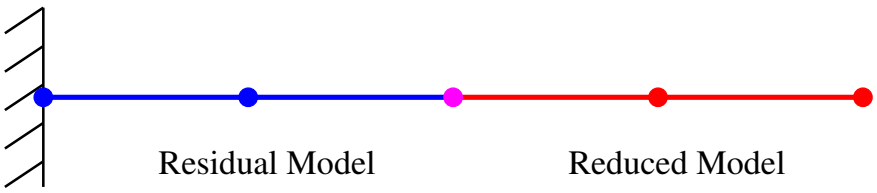


Figure 4-34. – Four Truss Geometry.

Theory. A CB model generates a transformation matrix consisting of a combined set of fixed interface and constraint modes. These modes may be stored in an exodus file. We call this "`se-base.exo`". A netcdf file, "`se.ncf`" is also created at this time. Subsequently, this reduced model is inserted into a residual model for superelement analysis, say a transient analysis. That analysis outputs the standard exodus results, "`resid-out.exo`" and results on the netcdf file, "`se-out.ncf`". The point is to recover the response on the original interior degrees of freedom of the superelement.

[2] The CBR reduction must use lumped masses for consistency with the statics solutions.

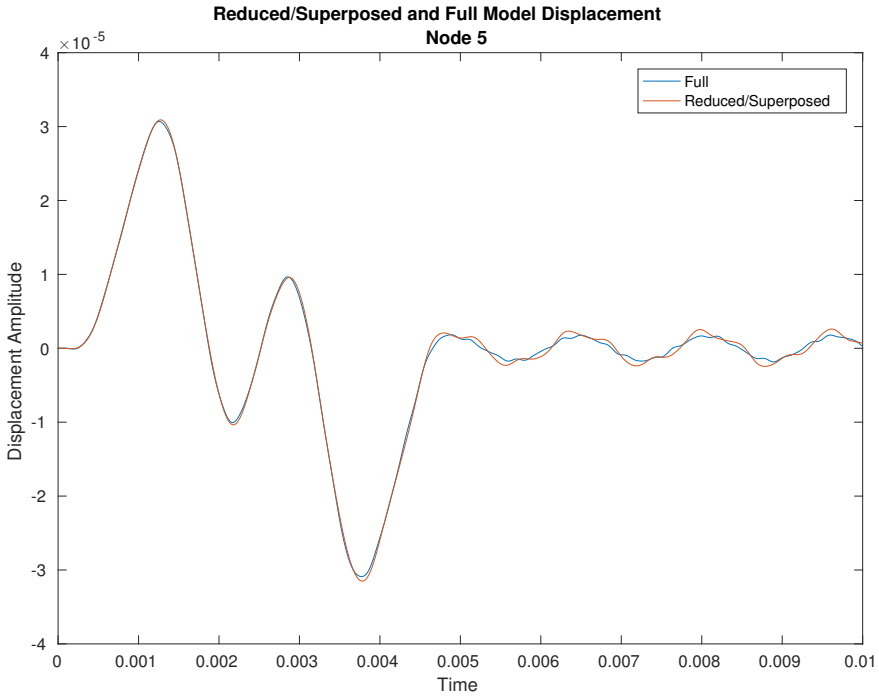


Figure 4-35. – Results of Superposition Problem.

The transient response on the interior degrees of freedom is,

$$u_k(t_n) = \sum_{i}^{nmodes} q_i(t_n)\phi_{ik} + \sum_{j}^{nconstraint} w_j(t_n)\psi_{jk} \tag{4.7.7}$$

where,

$$\begin{aligned} u_k(t_n) &= \text{is the displacement at interior dof } k \\ t_n &= \text{is the time step} \\ q_i &= \text{is the amplitude of a generalized dof for mode } i \\ \phi_{ik} &= \text{is the fixed interface mode } i \text{ at dof } k \\ w_j &= \text{is the amplitude of interface dof } j \\ \psi_{jk} &= \text{is the constraint mode } j \text{ at dof } k \end{aligned}$$

The amplitudes q_i and w_j are found in "`se-out.ncf`", while the mode shapes, ϕ_{ik} and ψ_{jk} are found in "`se-base.exo`". Super_superp simply combines these results and writes a new output file containing the results.

For input see Appendix 9.49

4.7.4. Inertia Tensor

The inertia tensor provides a means of applying initial conditions to the interior dofs of a superelement. General boundary conditions are not supported, but initial conditions that include linear combinations of rigid body motion can be readily managed. As these are the most common boundary conditions, there is great utility in computing the inertia tensor as part of the Craig-Bampton (CB) reduction process.

A CB reduction and the rigid body deformation define two inertia matrices. In terms of the CB reduction matrix T, a six column rigid body vector in the physical space R, and the mass matrix in the physical system, M, the inertia tensor,

$$I_v = T^T R,$$

and the mass inertia matrix,

$$I_m = T^T M R.$$

I_v is used to establish initial velocity, and I_m is used to apply gravity or other body loads.

The development of the inertia tensor was used for use in LS-Dyna. LS-Dyna also has the reduction process. Verification involves comparison of the output of the two codes. The LS-Dyna output is in DMIG format. We compare with a previous MATLAB output from Sierra/SD which was compared by hand with the LS-Dyna results. Also, Sierra/SD outputs the fixed interface modes first, while LS-Dyna puts them last. The model is shown in Figure 4-36.

The overall comparison of the values is very good with a relative L_2 norm about 6‰. Figure 4-37 compares the values of the matrix. Corresponding to each of the three translations, there are 3 rigid body modes. There are 10 fixed interface modes and 12 constraint modes, for 22 columns in the inertia tensor. The significant difference in mode 10 is because it is the last mode retained, and it is very near in frequency to the next highest mode.

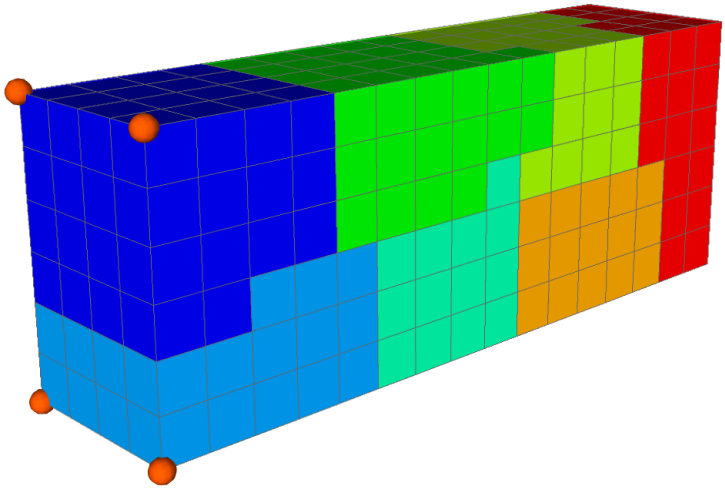


Figure 4-36. – LS-Dyna and Sierra/SD Inertia Tensor Model. The model is colored by the parallel decomposition.

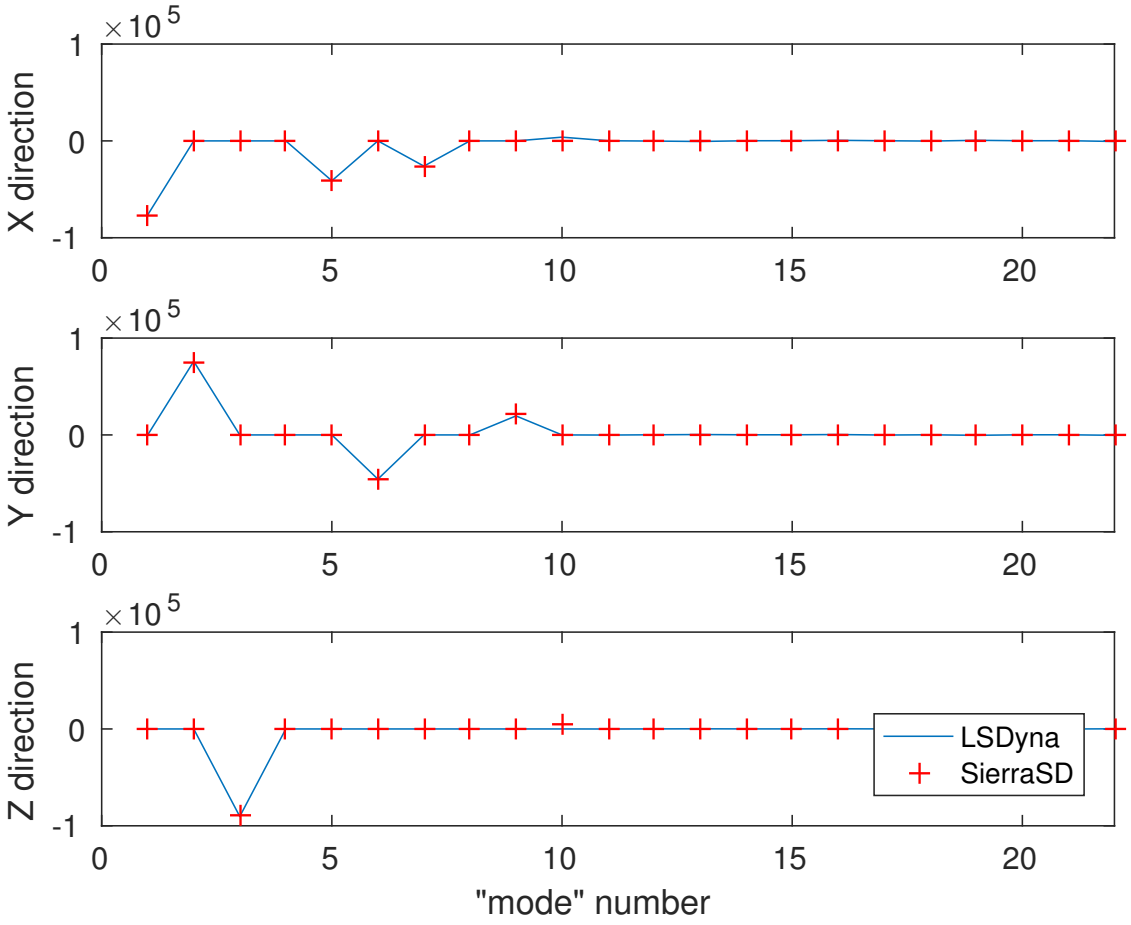


Figure 4-37. – LS-Dyna and Sierra/SD Inertia Tensor Terms.

4.7.5. Mass Inertia Matrix

The Mass Inertia matrix, $I_m = T^T M R$, is determined by a comparison with an independent MATLAB calculation, using the following steps.

1. Use the single processor input, and enable "mfile" output.
2. Run Sierra/SD to reduce the model and generate the mass inertia matrix.
3. Read in the fixed interface modes, ϕ, and constraint modes, ψ, from Sierra output.
4. Form the transformation matrix.
$$T = \begin{pmatrix} \phi & \psi \\ 0 & I \end{pmatrix}$$
5. Read the partitioned components of the mass matrix (M_{vv}, M_{cc}, and M_{cv}) from Sierra output. Generate a mass matrix that includes all dofs of interest.
$$M = \begin{pmatrix} M_{vv} & M_{vc} \\ M_{cv} & M_{cc} \end{pmatrix}$$
6. Compute and compare the reduced mass matrix computed by the two methods. $\hat{M} = T^T M T$.
7. Compute the Nx3 rigid body matrix. Only translational components are included.
8. Compare the Sierra computed Inertia Tensor, $I_v = T^T R$, with the LS-Dyna stored values. This is a code-to-code comparison. This is also compared with a Matlab solution.
9. Compute the Mass Inertia matrix, $Iv = T^T M R$, and compare results with those output from Sierra. A comparison of the results is shown in Figure 4-38.
10. Results are compared in serial and in parallel.

These steps found in the Matlab script, `massInertiaTensorCompare.m`.

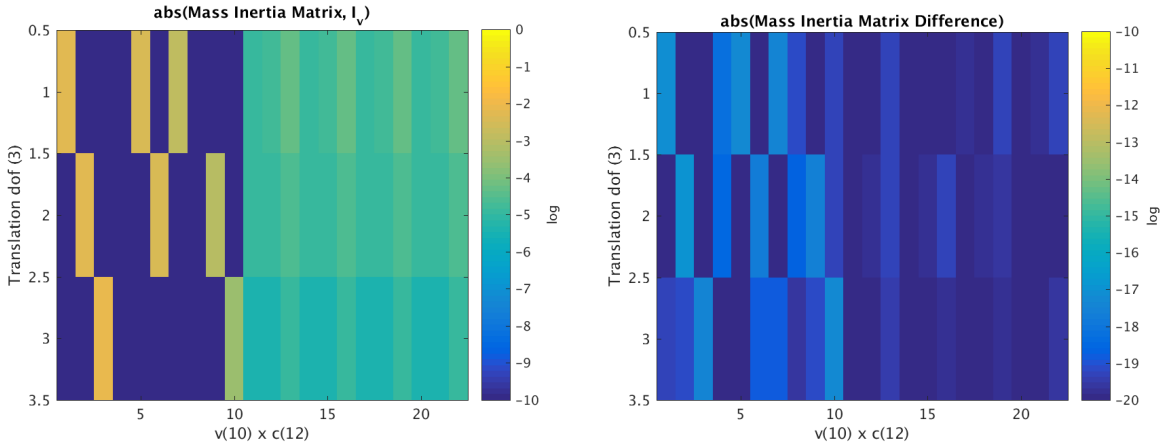


Figure 4-38. – Mass Inertia Matrix. Values (left) and Differences (right).

For input see Appendix 9.50.

4.7.6. *NASTRAN*/Sierra/SD *Interoperability*

There is often a need to exchange data with external collaborators. Most often these collaborators use commercial products for finite element analysis. One of the varieties of NASTRAN is the most commonly used exchange format. Sierra/SD has been designed to interface to these formats through its superelement capability. Export through a NASTRAN superelement may be done directly in Sierra/SD as part of the CBR method, or it may be accomplished through the "ncfout" application which translates the model into either DMIG or output4 format. In addition, Sierra/SD may import certain DMIG formatted models using NASGEN.

Such export/import capabilities provide the basis of interaction with collaborators, and it is important that the process be simple and accurate. However, while significant effort has been put into these tools, testing has been rather limited because of challenges in running NASTRAN in the Sierra test harness. Without regular testing, capabilities can not be trusted for crucial collaborations. The intent of this verification is to provide a well-defined testing strategy to ensure persistent capability. These tests may need to be run manually, but the tests should ensure capability.

This test does not regularly run NASTRAN. Section 4.7.9 contains instructions for running NASTRAN by hand to fully verify current analysis. The nightly test runs Sierra, and compares results carefully with previously completed analyses which had been compared with NASTRAN.

Scope of Evaluation. The focus of these tests is evaluation of the CBR exchange capability. In particular, we focus on the following.

1. Compatibility of the data format for exchange of reduced order stiffness and mass matrices.
2. Bi-directional capability, i.e., output of superelements from Sierra/SD in DMIG format, and input through NASGEN.
3. A clear, well-defined process for generating and using these reduced order models (or ROM).
4. Support for damping matrices, and output transfer matrices (OTM).
5. Support for inertia mass matrix export. The inertia mass matrix is not currently supported for boundary conditions in Sierra/SD. As such, it cannot be tested for import.

To keep the focus, we explicitly limit the following.

- No element comparisons. NASTRAN element formulations clearly differ from Sierra/SD capabilities. That is expected, and not tested here. Convergence of these elements to proper solutions is performed elsewhere.
- NASGEN translation of most data. There are extensive tests for translation of the model. Except for the superelement capabilities, these lie outside the scope of this set of testing.

4.7.6.1. Model Evaluation

The model must be evaluated for suitability for comparison. In particular, the solutions of the unreduced models (NASTRAN and Sierra), must be close enough to allow code to code comparison of reduced models.

The model is illustrated in Figure 4-39. There are three primary areas of consideration.

Base The support at the base provides the fixed boundary condition and the attachment location for the two tuning tines. It is part of the residual.

Load Tine The leftmost tine (red) is also part of the residual. Force/Pressure boundary conditions may be applied to this tine.

ROM Tine The rightmost tine (yellow) is the portion of the model to be reduced. The interface to the residual is the element at the base of the tine. There is a single point on the end of the tine that serves as a location for OTM evaluation.

All sections of the model use the same material properties (aluminum), and all use Hex20 elements, as these are expected to be very similar between the two applications. We evaluate the model for lowest eigenvalues and for a modal frequency response function (FRF) to an impulse on the side of the loading tine. The FRF provides a useful comparison, even when the time history data would suffer from phase errors introduced by small differences in the element formulations.

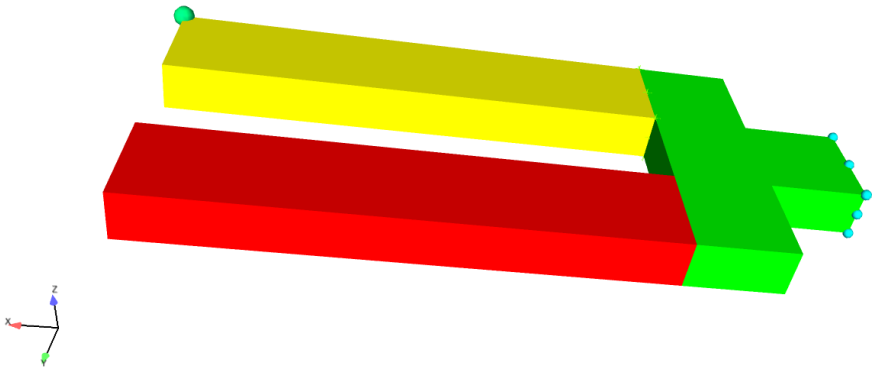


Figure 4-39. – Tuning Fork Model.

Table 4-27 provides a comparison of the frequencies for vibration of the structure.

#	Description	Sierra/SD	NASTRAN	Diff %
1	base bending	532.07	527.84	0.8%
2	symmetric bending	937.07	926.53	1.1%
3	asymmetric bending	2956.4	2891.84	2.2%
4	symmetric 2nd bending	4733.4	4630.10	2.2%

Table 4-27. – Vibrational Frequency Comparison.

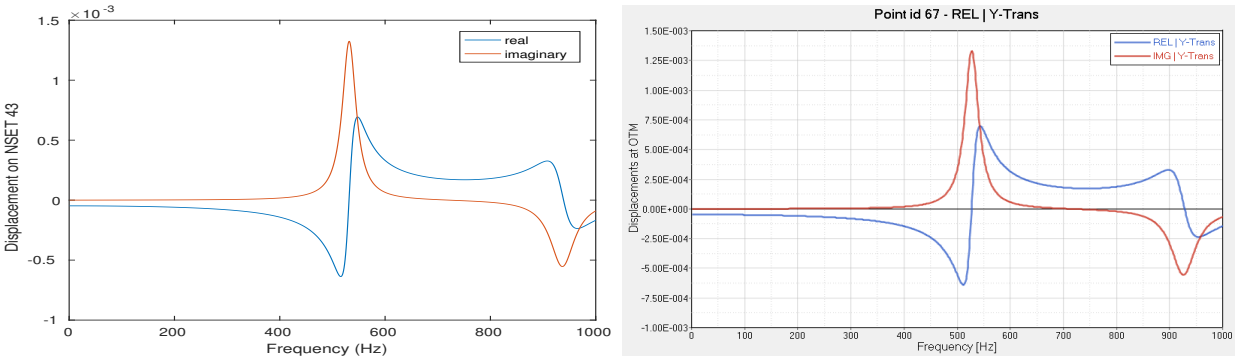


Figure 4-40. – FRF Solutions with 3% damping. Sierra/SD and NASTRAN.

Figure 4-40 compares the modal FRF solutions for the Sierra and NASTRAN solutions. The model is considered suitable for evaluation.

4.7.6.2. Superelement Reduction and Insertion

In Sierra/SD, the following steps are followed to compute the system response by superelement reduction techniques.

1. The ROM section of the exodus model is pulled out separately. This can be accomplished using grepos.
2. The CB reduction input is generated. This is similar to the full system model, with additions of a CB section.
3. Sierra/SD is run on the CBR input. This generates a netcdf output.
4. The residual model is generated. Like step 1, we use grepos and delete the block associated with the ROM.
5. A "socket" is created for the superelement, using "mksuper".
6. A residual input is created. This is very similar to the original full system model, but now contains entries for the new superelement block.
7. Sierra/SD is run on the residual input.

Commands for some of these operations are shown in Figure 4-41. A comparison of the eigenvalues with the full system eigenvalues is shown in Table 4-28. With no internal modes, significant errors are introduced. Four modes in the ROM represents the system well.

Mode	Full Model	4-Mode ROM	0-Mode ROM
1	532.065	532.066	551.163
2	937.066	937.066	1107.19
3	2956.37	2956.87	3758.39
4	4733.4	4734.76	6022.09

Table 4-28. – SierraSD full model eigenvalues compared with ROM.

1. *The ROM section of the exodus model is pulled out separately.*

```
grepos tuningforkx.exo rom.exo « EOF
   delete block 11
   delete block 31
EOF
```

2. *The CB reduction input is generated. The solution and cbmodel sections look like the following.*

```
SOLUTION
   CBR
     nmodes=4
END
cbmodel
   nodeset 41
   format=netcdf
   file=rom.ncf
   inertia_matrix=yes
end
```

3. *Sierra/SD is run on the CBR input. This generates a netcdf output.*

4. *The residual model is generated. This is identical to step 1, but deletes block 21.*

5. *A "socket" is created for the superelement, using "mksuper".*

```
mksuper tmp.exo « EOF
   add nodeset
   41
   write residual.exo
   quit
EOF
```

6. *A residual input is created. Copy full model input to residual.inp. Comment out block definition for block 21, and add definition for block 32.*

7. *Sierra/SD is run on the residual input, and compared with original model.*

Figure 4-41. – Sierra/SD solution with Superelement.

4.7.6.3. NASTRAN Model reduction and Insertion

In MSC or NX NASTRAN, one approach to compute the system response by superelement reduction techniques is described in the following steps.

1. The ROM section of the NASTRAN mesh file is pulled out separately. This was accomplished using the Altair HyperMesh preprocessor. The residual structure's node and element definition are saved as a separate bulk data file `residual_struct.bulk`.
2. The CB reduction input is generated in `cbr.bdf`. This requires using the `EXTSEOUT` card in the case control section. The definition of a `BSET` card containing the interface nodes (a-set dofs) to be constrained during the dynamic reduction step is also required. A `QSET` card is used to define the generalized dofs (q-set) to be used for the reduction. Lastly, a `SPOINT` card is necessary to define scalar points for the generalized dofs. Note that the number of generalized dofs requested should not be excessive – otherwise, the reduced matrices will have null columns for unused q-set dofs and may result in a performance degradation.
3. NASTRAN solves the eigenvalue problem (`SOL 103`). The `EXTSEOUT` card in the case control section has many options for the type and format of superelement information generated. In this example, the `EXTSEOUT` card was specified to request a punch (.pch) file `cbr.pch` that contains the reduced stiffness and mass DMIG matrices. Additional superelement information (e.g., DMI matrices and DTI tables that are associated with the OTM) which may not be necessary for subsequent use is also generated by default.
4. The resulting punch file `cbr.pch` is then cleaned up by removing all the information within it except the stiffness and mass DMIG matrices. The names of the DMIG matrices were also renamed to something more convenient. This updated punch file can be saved as `cbr_dmig.pch`.
5. The residual (residual structure with the superelement attached) input is created. This is very similar to the original full system model, but contains additional cards that insert the superelement via DMIG input. The stiffness and mass DMIG matrices are called in using the K2GG and M2GG cards, and the `SPOINT` card must be included to define the generalized dofs.
6. NASTRAN is run on the residual input.

Additional details of NASTRAN's superelement functionality can be found in Reference [47] (MSC NASTRAN 2017 Superelements User's Guide). Eigenvalues of the full model and the residual model with superelement are shown in Table 4-29. The results are practically identical.

Figure 4-42 compares the input displacement of the Sierra/SD and MSC/NASTRAN ROM on a Sierra/SD residual. Data on the output (ROM) tine is not available with these methods because the basis vectors of the ROM are available only internal to NASTRAN.

	MSC NASTRAN 2016 (full Model)	MSC NASTRAN 2016 (NASTRAN based DMIG)	
Mode	Natural Frequency [Hz]	Natural Frequency [Hz]	Difference [%]
1	528	528	0.00
2	927	927	0.00
3	2,892	2,892	0.00
4	4,630	4,630	0.00
5	6,078	6,078	0.00
6	6,446	6,446	0.00
7	8,118	8,119	0.01
8	12,863	12,864	0.01
9	14,426	14,427	0.01
10	17,672	17,681	0.05

Table 4-29. – MSC NASTRAN Full Model Eigenvalues Compared with ROM.

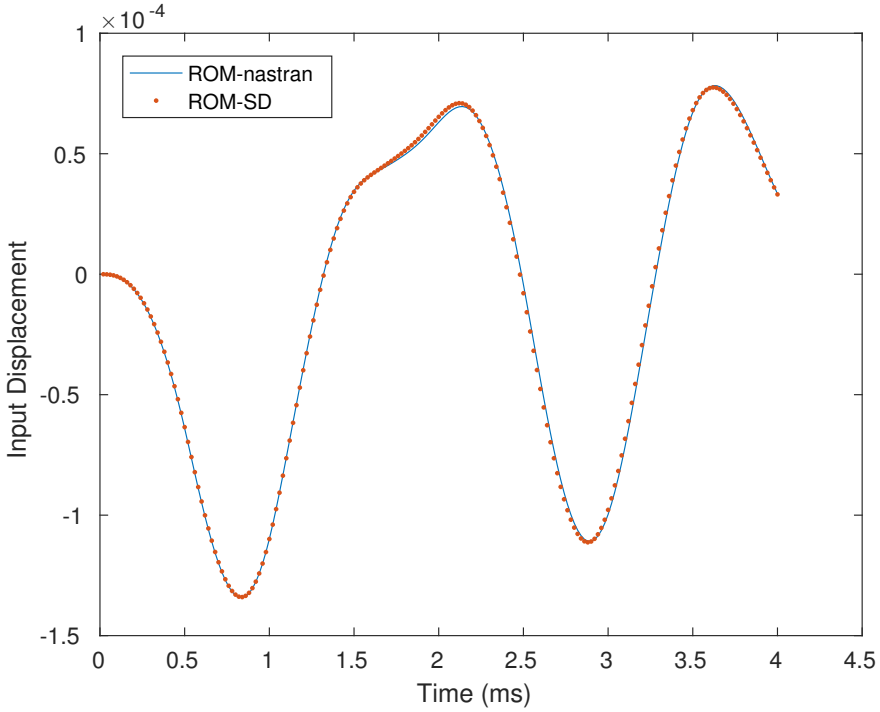


Figure 4-42. – Modal Transient Comparison. The input displacement of the Sierra/SD and MSC/NASTRAN ROM on a Sierra/SD residual.

4.7.6.4. Using Sierra/SD Superelements in NASTRAN

It is also informative to compare the eigenvalues to assess the equivalence of the DMIG matrices generated by Sierra/SD and NASTRAN. In one case, DMIG matrices are exported by Sierra/SD and then used within NASTRAN to attach to the residual structure for an eigenvalue problem. In the second case, DMIG matrices are generated entirely within NASTRAN. These results, shown in Table 4-30, indicate that within practical frequencies of interest, Sierra/SD produces very similar reduced matrices to NASTRAN. Results of a modal frequency response analysis for the full NASTRAN model and the residual model with superelement are shown in Figure 4-43. The output is located at node 14, which lies at the boundary between the residual mesh and the superelement. The results are practically identical [3].

Mode	MSC NASTRAN 2016 (Sierra/SD based DMIG) Natural Frequency [Hz]	MSC NASTRAN 2016 (NASTRAN based DMIG) Natural Frequency [Hz]	Difference [%]
1	528	528	-0.04
2	931	927	-0.52
3	2,916	2,892	0.84
4	4,675	4,630	0.95
5	6,144	6,078	1.07
6	6,499	6,446	0.83
7	8,292	8,119	2.09
8	13,209	12,864	2.62
9	14,972	14,427	3.64
10	17,796	17,681	0.65

Table 4-30. – Eigenvalue Comparison – Sierra/SD -generated DMIG and NASTRAN-generated DMIG. Residual and Superelement are employed in each analysis.

Sierra/SD computes a superelement using a Craig-Bampton reduction. That reduced order model may be written in several formats. For use in Sierra/SD, we write this as a netcdf/exodus file. It may alternatively be written as a DMIG [4] compatible with NASTRAN. More flexibly, we can convert the netcdf/exodus file to several formats (including DMIG and Output4) using the `ncfout` application.

For application of a DMIG to a NASTRAN model, the interface node numbers must be consistent. Figure 4-44 illustrates the nodes on an interface, together with the first few lines of the DMIG, which define a portion of the reduced stiffness matrix. Each row and column is indicated by the GRID/CID pair.

The original BDF file must be modified as follows.

1. Copy original, and remove the five elements in the ROM region.

[3] Sierra/SD has recently added a higher precision DMIG output. This uses 16 character "long" format NASTRAN fields, and is selected with the "FMT=dmig*" option.

[4] Direct Matrix Input at Grid points

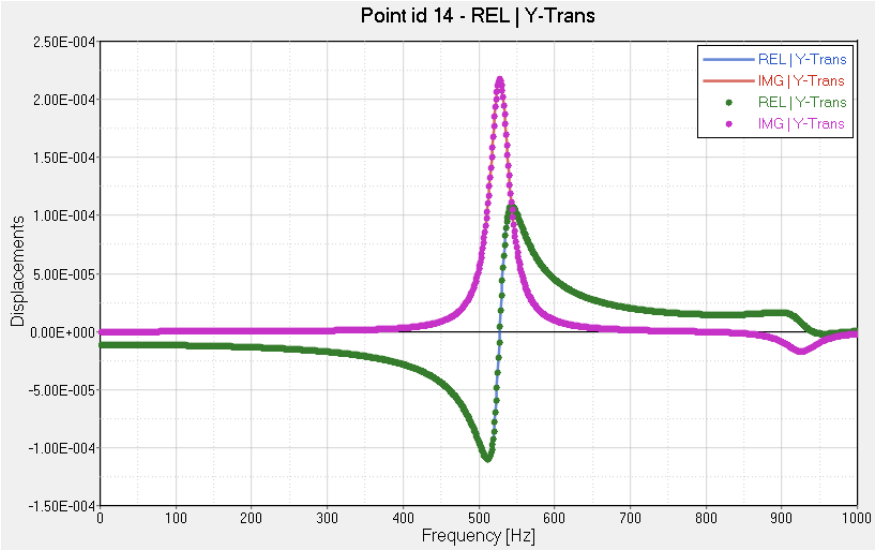


Figure 4-43. – Modal FRF Comparison: Full Model (line) and Residual Model with Superelement (markers).

```
DMIG    K2GG            6     1                6     1  1.6896+6
                        6     2  -197648.      14     1   726510.
                       14     2   36078.7      47     1   899966.
                       47     2   63681.1      47     3   90702.2
                       55     1   750185.      55     2   58875.3
                       55     3  -67421.3      97     1  -2.141+6
                       97     2  -115178.      97     3  -275194.
                      111     1  -454831.     111     2   45753.7
                      113     1  -1.028+6     113     2   156507.
                      113     3  -587.767     116     1  -442113.
                      116     2  -48070.2     116     3  -202927.
DMIG    K2GG            6     2                6     2  2.0542+6
                       14     1  -36078.7      14     2  1.0384+6
```

Figure 4-44. – DMIG example. On the left, the interface nodes and orientation from the model is shown. The extract from the DMIG on the right illustrates the first row of the stiffness matrix. The index to each value is the GRID and CID pair for that column.

2. Add SPOINTS corresponding to the DMIG.
3. Include the new DMIG data.
4. Add commands to include K2GG and M2GG in the case control.

Table 4-31 compares the eigenvalues of the full NASTRAN model to the eigenvalues of the reduced order model from **Sierra/SD**. The solution with four fixed interface modes provides good accuracy. [5]

Mode	Full Model	4-Mode ROM	0-Mode ROM
1	527.842	528.046	546.461
2	926.535	931.379	1098.845
3	2891.837	2916.451	3681.061
4	4630.102	4674.563	5980.433

Table 4-31. – Eigenvalue Comparison - NASTRAN full model and with **Sierra/SD** ROMs.

4.7.7. Using NASTRAN Superelements in Sierra/SD

The NASTRAN superelement model is translated using `NASGEN`. This tool translates the model and superelement simultaneously, with the superelement written to a netcdf file. NASTRAN uses a different element formulation, and orders the modes differently from Sierra/SD, so we may not reasonably directly compare the matrices output in the translation. It is possible to simply run the translated analysis using Sierra/SD. The compared eigenvalues are shown in Table 4-32. The results are very reasonable.

Mode	Sierra/SD (Hz)	NASTRAN (Hz)
1	530.594	527.8421
2	932.069	926.5357
3	2930.28	2891.865
4	4692.38	4630.148

Table 4-32. – Comparison of NASTRAN and Sierra/SD Eigenvalues using NASTRAN Superelement.

4.7.8. Superposition Methods for Output of Internal Data

The Craig-Bampton method necessarily removes internal physical degrees of freedom from the superelement. Sometimes results on those internal dofs are required. The displacements, accelerations and velocities on these locations may be readily obtained through post-processing using the `super_superp` tool.

[5] The default data width for a DMIG is 8 characters. There may be a significant loss of accuracy in truncating data to this size. We have recently added the option to output 16 character DMIG using the DMIG* format.

Figure 4-45 compares the output of the sample on nodeset 41, at the tip of the unloaded tine, from the full model with the results obtained using the reduced model. Both models are run in Sierra/SD for consistency. The left tine is loaded with an impulse. Figure 4-46 illustrates the deformation of the full model, compared with the residual and superimposed superelement.

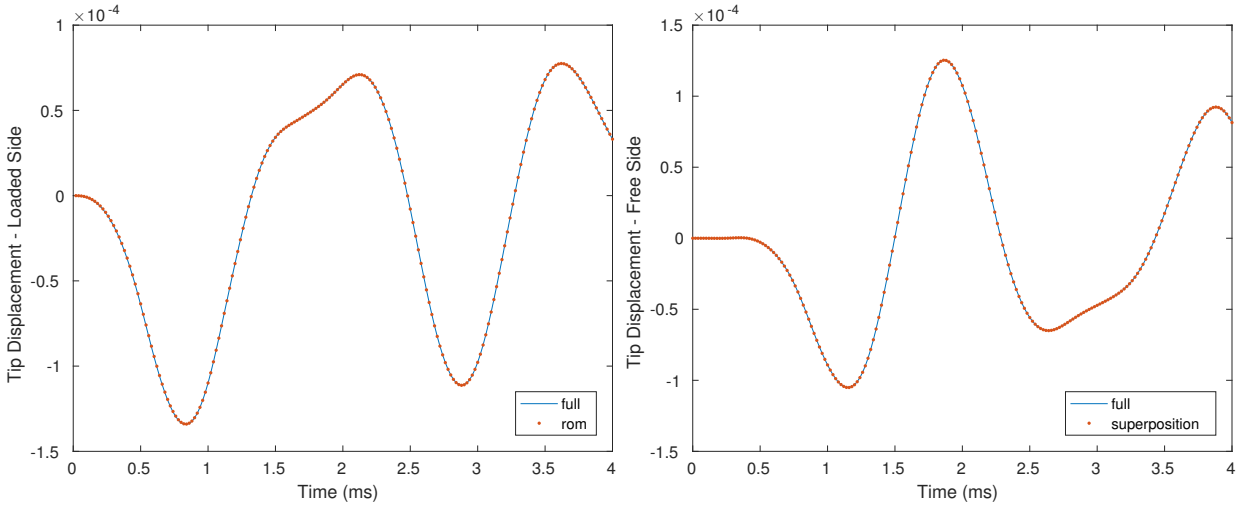


Figure 4-45. – Comparison of Output Displacements. The plot on the left compares displacements of the full and reduced order models at the input location. The plot on the right compares displacements on the unloaded tine after the `selem_superp` tool is used to extract the displacement from the reduced model.

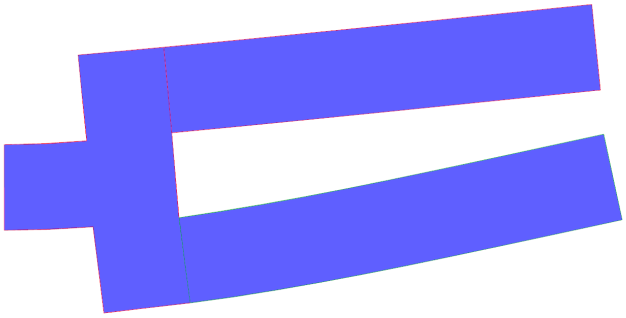


Figure 4-46. – Superposition Solution and Full Deformation, $t = 2\,ms$. The full model (in blue background) is compared with the residual and the post-processed superelement.

4.7.9. Related NASTRAN Analyses Required for Verification

The NASTRAN inputs for these analyses are included in the test repository, but are not run as part of the nightly test process. To evaluate these models, the following steps may be followed.

4.7.9.1. Eigenvalue Problem

NASTRAN approximates the eigenvalues of the entire model by running:

```
workstation> nastran tuningfork.bdf
```

The resulting output in `tuningfork.f06`, may be evaluated for the appropriate normal mode frequencies.

4.7.9.2. Modal FRF

The analysis may be modified to run a modal frequency response. Most modifications are in the case control section. Analyze with,

```
workstation> nastran tuningforkfrf.bdf
```

Output analysis is a relatively easy using NASTRAN aware tools, or the PCH file may be mined to garner the data.

4.7.9.3. Insertion of a ROM from Sierra/SD

There are relatively few changes required to the original BDF file required to include a DMIG from Sierra/SD. See the example in se.bdf, which includes the DMIG for the rightmost tine.

```
workstation> nastran se.bdf
```

Output of this analysis is the normal modes solution (as in section 4.7.9.1), but with the ROM of the right tine. Comparison of the modal frequencies provides validation of the analysis.

4.7.9.4. Insertion of a ROM from NASTRAN

The eigen_se.bdf file provides the input for NASTRAN analysis using the NASTRAN generated superelement. The superelement (in DMIG format) is read using an 'include' command. Analysis is performed using this command.

```
workstation> nastran eigen_se.bdf
```

The eigenvalues are found in the `f06` output file and may be compared with the **Sierra/SD** results of Section 4.7.7.

For input see Appendix 9.51.

4.7.10. Sierra/SD *Superelement File Formats*

In this section, we consider the tuning fork model shown in Figure 4-39 and described in Section 4.7.6. We modify the workflow shown in Figure 4-41 to compare the full model (without CB reduction) against the superelement results using both `format=netcdf` and `format=dmig*`: for each file format, the superelement is written to disk and read back in to be used in the analysis. The results (c.f. Table 4-27) are shown in Table 4-33.

#	Description	Full Model	Rel. Diff. (netcdf)	Rel. Diff. (DMIG)
1	base bending	532.07	1.69649e-08	1.60495e-08
2	symmetric bending	937.07	4.47213e-09	4.36792e-09
3	asymmetric bending	2956.4	4.08084e-06	4.08064e-06
4	symmetric 2nd bending	4733.4	6.50282e-06	6.50278e-06

Table 4-33. – Eigenvalue Comparison.

For input see Appendix 9.52.

4.8. Legacy Element Verification Results

The purpose of this section is to report the verification calculations that have been performed on the **Sierra/SD** software. Test models and calculations were performed to ensure that **Sierra/SD** performs as required. Element patch tests are described, convergence studies for the elements are performed, and code to code comparisons are made to ensure that the software meets the requirements for analysis of hypersonic vehicles used in *Sandia National Labs*'s nuclear weapons program.

Verification tests can never cover the full aspects of the software. Analysis shows that there are too many paths through the software to ever adequately cover all such paths (see *Beizer*[6] or *Myers*[38]). However, these tests are essential to provide confidence that with proper input, solutions to the fundamental equations of mechanics are solved properly.

Note that verification tests address mesh discretization indirectly.

4.8.1. Patch Tests

The element patch tests in this study are derived from MacNeal's monograph.[34] These tests are designed to ensure that the element formulations are independent of element orientation, and that the elements are capable of solving exactly the equations on which they are based. As a minimum, elements should be able to represent a constant strain field exactly since the linear shape functions of the elements are the minimum required to do this exactly.

All the 2D and 3D elements in the **Sierra/SD** element library are tested. The 2D elements are: QuadT, Tria3, TriaShell, and Tria6. The 3D elements are Hex8b, Hex8, Hex20, Wedge6, Tet4, and Tet10. The 2D elements are tested using a membrane patch test and a bending plate patch test. The 3D elements are tested using the solid patch test. These patch tests are defined in *MacNeal*.[34]

All the 2D elements pass the membrane and bending patch tests. All the 3D elements pass the solid patch test. These patch test problems are located in the Salinas_test repository in the Salinas_test/patch_tests subdirectory. The results for the patch tests are shown in Table 4-34.

Table 4-34. – Patch Test Results.

Element Type	Patch Test		
	Membrane	Bending	Solid
QuadT	Passed	Passed	N/A
Tria3	Passed	Passed	N/A
TriaShell	Passed	Passed	N/A
Tria6	Passed	Passed	N/A
Hex8	N/A	N/A	Passed
Hex8b	N/A	N/A	Passed
Hex20	N/A	N/A	Passed
Wedge6	N/A	N/A	Passed
Tet4	N/A	N/A	Passed
Tet10	N/A	N/A	Passed

4.8.2. Accuracy Tests

Accuracy tests are designed to stress test elements. These are not convergence tests. The purpose of the test is to provide information about how badly the element performs in common (but under meshed) environments. It can be noted in the results below that `Tet4` elements are way too stiff in almost all loadings. This is expected, and the test results are provided to help analysts determine the applicability of this element for their analysis. Below are test results for the accuracy tests (Tables 8 through 15 of *MacNeal* [34]). All tabulated results are the ratio of the numerical solution to the exact solution, i.e., a value of 1.00 is a perfect result. The test problems are described and illustrated in the reference, Figures 4 through 10.

The first test from *MacNeal* is a straight beam with a length of 6.0, an in-plane cross-sectional dimension of 0.2 and an out of plane cross-sectional dimension of 0.1. There is a single element at any given point along the length of the beam and total of 6 elements along the length of the beam. The Young's Modulus, $E = 10^7$, the Poisson ratio, $\nu = 0.30$, and the loading is a unit force at the free end of the beam. Reported table values refer to displacement at the loaded tip of the beam. Tables 4-35, 4-36 and 4-37 show results for rectangular, trapezoidal, and parallelogram shaped elements, respectively.

In the tables Hex8 denotes the Hex8U element.

Table 4-38 below shows results for a curved beam, also with a 6 by 1 element mesh. The inner radius is 4.12, the outer radius 4.32, the arc 90 degrees, and the thickness 0.1. The Young's Modulus is $E = 1e7$, the Poisson ratio is 0.25. The tip load is of unit magnitude.

Table 4-39 shows results for a cantilever beam that twists 90 degrees along the length of the beam. The beam length is 12.0, the in-plane cross-sectional dimension 0.32 and the out of plane cross-sectional dimension is 1.1. The Young's Modulus is 29.0e6 and the Poisson ratio 0.22. The tip load is of unit magnitude.

Table 4-35. – Straight Beam – *Rectangular* Elements.

Element Type	Extension	In Plane	Out of Plane	Twist
Hex20	0.994	0.970	0.961	0.904
Hex8b	0.988	0.978	0.973	0.892
Hex8	0.986	9.22	2.50	89.2
Tet10	0.998	0.960	0.959	0.910
Tet4	0.979	0.0219	0.0119	0.00264
Wedge6	0.991	0.0326	0.0882	0.0257
QuadT	0.839	1.05	0.979	0.704
Tria6	0.999	1.00	0.988	0.716
Tria3	1.01	1.06	0.978	0.704
TriaShell	0.966	0.224	.0978	0.720

Table 4-36. – Straight Beam – *Trapezoidal* Elements.

Element Type	Extension	In Plane	Out of Plane	Twist
Hex20	0.977	0.731	0.714	0.863
Hex8	0.988	0.734	0.307	51.4
Hex8b	1.009	0.0475	0.03	0.623
Tet10	0.999	0.277	0.208	0.667
Tet4	0.978	0.0144	0.00691	0.00755
Wedge6	0.992	0.0187	0.0302	0.0546
QuadT	1.00	0.559	0.980	0.0226
Tria6	0.999	1.00	0.988	0.716
Tria3	0.999	0.733	0.980	0.705
TriaShell	0.996	0.208	0.979	0.721

Tables 4-40 through 4-43 show results for a rectangular plate with either simply supported or clamped boundary conditions and either a point load of $4x10^4$ at the center of the plate or a uniform pressure of $1x10^4$ over the plate. The plate height is 4.0, and the plate width-to-height aspect ratio is either 1.0 or 5.0. The plate thickness is 0.01 for solid elements (Hex20, Hex8, Hex8b, Tet10, Tet4, and Wedge6) and 0.0001 for shell elements (QuadT, Tria6, Tria3, and TriaShell). Young's Modulus is $1.7472x10^7$ and Poisson's ratio is 0.3. The quantity N in these tables denotes the number of node spaces on half the edge of the plate. If the element has mid-side nodes, e.g., the Hex20, Tet10, or Tria6, then the number of elements along this portion of the edge of the plate is half the value of N. These tests are unsuitable for the Tet elements (Tet10 and Tet4) as the aspect ratios of the elements is large due to the small thickness. NASTRAN's Tet10 performs similar to **Sierra/SD**'s Tet10 on the remaining problems in this section.

Table 4-44 shows the results for the Scordelis-Lo Roof test of a curved plate. An 80 degree arc has radius of curvature is 25.0. The length of the plate is 50.0 and the thickness 0.25. The straight edges of the plate are free and the curved edges are constrained to not to move in the plane in

Table 4-37. – Straight Beam Tests – *Parallelogram* Elements.

Element Type	Extension	In Plane	Out of Plane	Twist
Hex20	1.01	0.404	0.280	0.758
Hex8	0.983	1.60	0.943	38.68
Hex8b	0.977	0.623	0.528	1.27
Tet10	0.998	0.289	0.213	0.744
Tet4	0.981	0.0122	0.00708	0.00779
Wedge6	0.991	0.0148	0.0558	0.154
QuadT	0.985	0.407	0.981	0.141
Tria6	0.998	0.816	0.988	0.716
Tria3	1.00	0.535	0.978	0.702
TriaShell	0.996	0.190	0.978	0.720

Table 4-38. – Curved Beam Tests.

Element Type	In Plane	Out of Plane
Hex20	0.874	0.937
Hex8	7.06	22.8
Hex8b	0.879	0.952
Tet10	0.839	0.776
Tet4	0.0174	0.00738
Wedge6	0.0255	0.0557
QuadT	1.09	0.867
Tria6	.167	0.276
Tria3	1.07	0.864
TriaShell	0.185	0.895

which the curved edge is contained. A traction of magnitude 90.0 per unit area is applied in the z-direction on the face of the plate. Young's Modulus is 4.32e8 and the Poisson ratio 0.0. The quantity *N* still represents the number of node spaces along half of one of the edges of the plate.

Table 4-45 gives the results for the spherical shell tests. This is a semi-spherical shell with a hole cut out of the top. The angular size of the hole is 36 degrees. The radius is 10.0. The thickness is 0.04. The Young's Modulus is 6.825e7. The Poisson ratio is 0.3. The loading is made up of four equally spaced radial point loads of magnitude 2.0 at the equator. Two of these point loads are radial inward and two are radially outward. The quantity *N* represents the number of node spaces along a quarter of one of the edges of the shell.

The next table (Table 4-46) shows the results for the thick walled cylinder tests. This is a donut shaped, thick plate of thickness 1.0, inner radius 3.0, and outer radius 9.0. The Young's Modulus is 1000, and the Poisson ratio is either 0.49, 0.499, or 0.4999. The loading is a unit radial pressure on the inner radius. The mesh has five elements along the radius at 10 degree intervals and one element through the thickness, for a total of 180 elements.

Table 4-39. – Twisted Beam Tests.

Element Type	In Plane	Out of Plane
Hex20	.996	0.987
Hex8	14.3	11.0
Hex8b	0.744	0.740
Tet10	1.01	1.01
Tet4	0.0949	0.162
Wedge6	0.0846	0.243
QuadT	.998	1.01
Tria6	19.7	15.5
Tria3	30.9	24.6
TriaShell	11.4	8.99

Table 4-40. – Rectangular Plate with Simple Supports and Uniform Pressure Load, Aspect Ratio 1.0

Element Type	N=2	N=4	N=6	N=8
Hex20	0.0167	0.691	0.831	0.976
Hex8	0.220	0.904	2.02	3.11
Hex8b	0.04	0.412	0.782	0.92
Tet10	0.00116	0.00331	0.00752	0.015
Tet4	4.42e7	8.00e6	4.10e5	1.29e4
Wedge6	0.228	0.0824	0.0568	0.0543
QuadT	0.966	0.922	0.997	0.998
Tria6	1.01	0.974	0.987	0.992
Tria3	0.978	0.992	0.997	0.998
TriaShell	0.958	0.987	0.994	0.997

Table 4-41. – Rectangular Plate with Simple Supports and Uniform Pressure Load, Aspect Ratio 5.0

Element Type	N=2	N=4	N=6	N=8
Hex20	0.503	0.649	1.04	1.02
Hex8	0.130	0.515	19.21	2.03
Hex8b	0.024	0.302	1.10	0.917
Tet10	0.000702	0.00181	0.00424	0.00852
Tet4	1.57e7	2.52e6	1.28e5	4.05e5
Wedge6	0.179	0.0977	0.0474	0.0470
QuadT	0.978	0.993	0.994	0.999
Tria6	0.658	1.02	1.01	1.00
Tria3	0.945	0.991	0.997	0.999
TriaShell	0.960	0.995	0.999	0.999

Table 4-42. – Rectangular Plate with Clamped Supports and Concentrated Load, Aspect Ratio 1.0

Element Type	N=2	N=4	N=6	N=8
Hex20	0.00106	0.072	0.553	0.822
Hex8	0.120	0.578	1.33	2.36
Hex8b	0.0195	0.246	0.614	0.824
Tet10	0.00110	0.00329	0.00624	0.0109
Tet4	1.46e6	2.31e5	1.15e4	3.52e4
Wedge6	0.0037	0.0186	0.0373	0.0561
QuadT	1.08	1.03	1.02	1.01
Tria6	1.06	1.17	1.01	1.01
Tria3	0.778	1.03	1.02	1.01
TriaShell	0.860	1.02	1.01	1.01

Table 4-43. – Rectangular Plate with Clamped Supports and Concentrated Load, Aspect Ratio 5.0

Element Type	N=2	N=4	N=6	N=8
Hex20	8.51e4	0.0396	0.220	0.374
Hex8	0.0362	0.138	0.551	0.992
Hex8b	0.00585	0.083	0.247	0.415
Tet10	3.39e4	0.00141	0.00282	0.00475
Tet4	2.26e7	3.60e6	1.80e5	5.61e5
Wedge6	0.00320	0.0181	0.0241	0.0297
QuadT	0.613	0.919	1.00	1.01
Tria6	0.606	0.910	0.998	1.01
Tria3	0.603	0.915	1.00	1.01
TriaShell	0.666	0.945	1.01	1.02

Table 4-44. – Scordelis-Lo Roof Tests.

Element Type	N=2	N=4	N=6	N=8	N=10
Hex20	0.0583	0.276	0.645	0.870	0.956
Hex8	.563	1.43	2.17	2.73	3.16
Hex8b	0.125	0.574	0.889	0.967	0.981
Tet10	0.0198	0.0526	0.0770	0.101	0.149
Tet4	0.00599	0.0108	0.0196	0.0333	0.0472
Wedge6	0.017	0.0289	0.0642	0.08	0.093
QuadT	1.58	1.13	1.06	1.02	1.00
Tria6	1.45	1.13	1.06	1.02	1.00
Tria3	1.45	1.13	1.06	1.02	1.00
TriaShell	1.35	1.04	1.01	0.995	0.984

Table 4-45. – Spherical Shell Tests.

Element Type	N=2	N=4	N=6	N=8	N=10	N=12
Hex20	–	0.00129	0.00662	0.0209	0.0500	0.0974
Hex8	0.00573	0.0547	0.133	0.238	0.371	0.531
Hex8b	.000303	0.0104	0.056	0.162	0.319	0.491
Tet10	–	2.21e4	3.83e4	6.73e4	0.00107	0.00167
Tet4	2.22e5	3.18e5	3.78e5	4.46e5	5.62e5	6.94e5
Wedge6	0.0153	0.00447	0.00645	0.00660	0.00708	0.00781
QuadT	0.0423	0.0834	0.263	0.502	0.697	0.820
Tria6	0.0194	0.0879	0.263	0.502	0.697	0.819
Tria3	0.0445	0.0891	0.266	0.499	0.693	0.816
TriaShell	0.436	0.199	0.226	0.378	0.560	0.708

Table 4-46. – ThickWalled Cylinder Tests.

Element Type	ν = .4900	ν = .4990	ν = .4999
Hex20	1.03	1.04	1.04
Hex8	0.445	0.437	0.406
Hex8b	0.437	0.437	0.437
Tet10	0.444	0.442	0.442
Tet4	0.393	0.356	0.349
Wedge6	0.408	0.399	0.398
QuadT	0.416	0.414	0.413
Tria6	0.438	0.436	0.436
Tria3	0.419	0.417	0.417
TriaShell	0.425	0.423	0.423

4.8.3. Convergence Tests

Mesh convergence studies establish confidence that the accuracy of the solution increases as the mesh is refined. They also establish the rate of convergence of the solution. They may be performed with or without a known analytical solution for the problem. Fortunately, for many structural dynamics problems, analytic solutions are available.

In structural dynamics, unstructured grids are necessarily used. While standard Richardson extrapolation[41] is not directly applicable to unstructured meshes, related methods can be used to determine truncation error (see *Alvin*[3] for example).

Convergence testing is used either to explore the properties of newly designed elements or to assure the adequacy of a candidate mesh. Use of it to verify the correct implementation of an element is not universally done; instead the patch test and the accuracy tests are considered sufficient. Convergence testing is performed as part of this verification suite to provide consistency with verification efforts in other *Sandia National Labs* codes.

In its simplest form, convergence analysis involves performing an analysis with at least three levels of mesh fineness and assessing the rate at which the error goes to zero. For the elements under consideration, convergence is known to be geometric: quadratic for the low order elements and quartic for the high order elements once the elements are small enough.

The convergence tests for the Hex8 elements was the static deformation of a cantilevered beam. The meshes employed are shown in Figure 4-47 and the appropriate plot of convergence error is show in Figure 4-48. It was seen that the convergence slope increased in magnitude as the meshes were refined and that for both the fully integrated and the selectively integrated element, the slopes found through this numerical experiment approximate the theoretical value of -2. Fine meshes are required to achieve this geometric convergence. Requiring convergence at a single point was a mistake.

Element convergence for Hex20 and Tet10 elements was preformed focusing on the calculated first eigenvalues. The resulting convergence plot for the Hex20 is shown in Figure 4-49. Here we see that the convergence rate is -3.8, close to the theoretical value.

Refining a general mesh through sectioning to create new elements all of approximately the same size increases the aspect ratios compared to the coarser mesh. This prevents standard convergence tests of Tet elements. A BCC mesh can be uniformly refined, but it was too difficult to implement. Instead several independent meshes were created. The resulting slope of the log-log error plot (shown in Figure 4-50 is the theoretical value, -4.

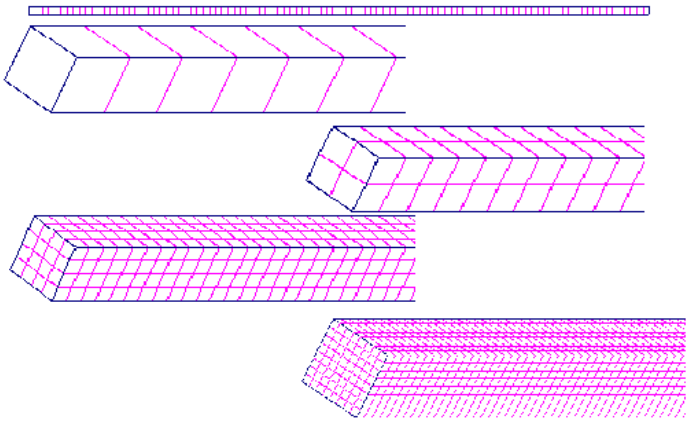


Figure 4-47. – Meshes for convergence test for Hex8 elements.

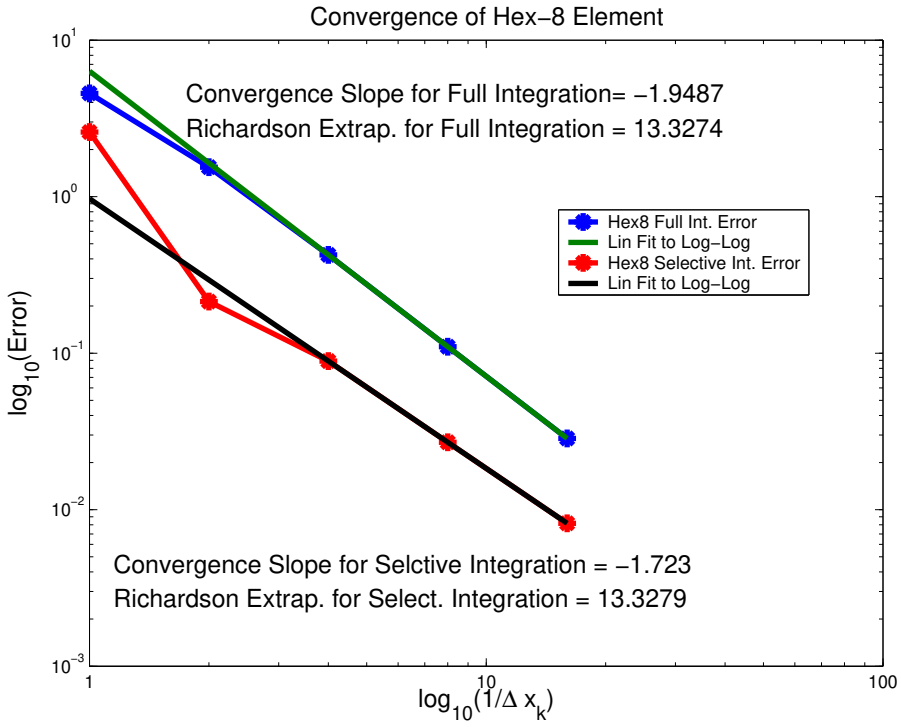


Figure 4-48. – As the meshes are progressively refined, the slope of the log-log plot of the error approaches -2, as predicted by theory.

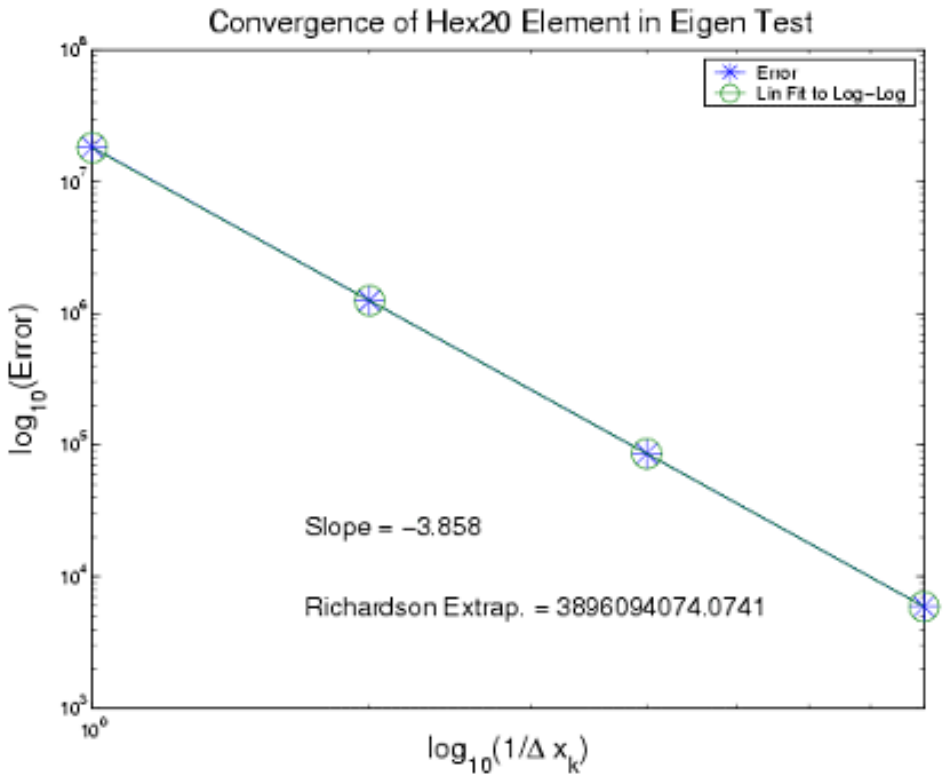


Figure 4-49. – The convergence plot of the Hex20 element for the first eigenvalue shows a slope close to the theoretical value of -4.

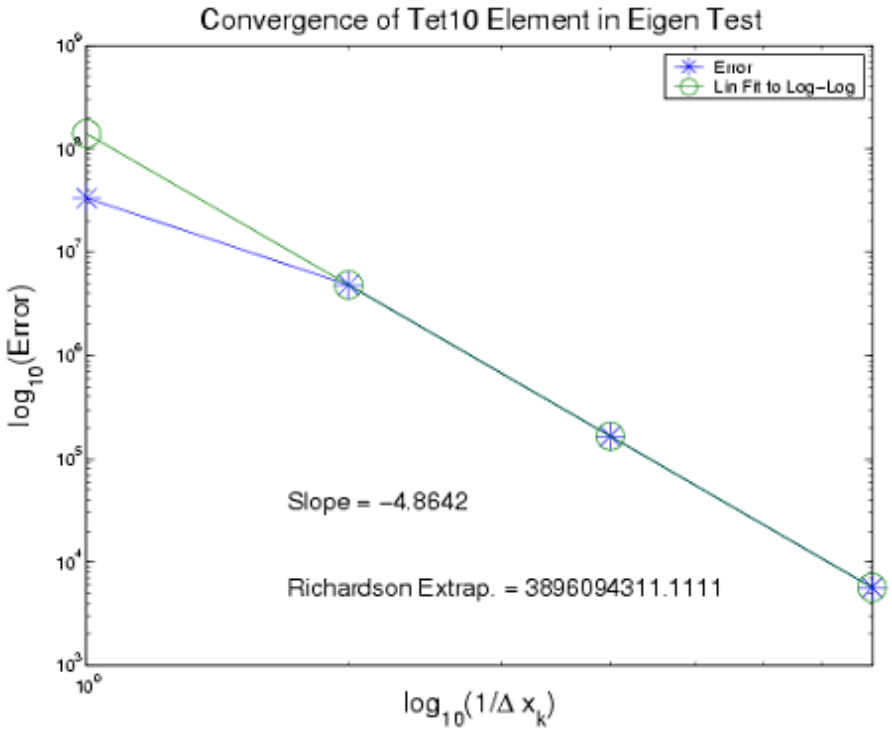


Figure 4-50. – The convergence plot of the Tet10 element for the first eigenvalue shows a slope close to the theoretical value of -4

The convergence rates of the various elements are listed in Table 4-47.

Table 4-47. – Element Convergence Rates.

Element Type	Ideal Rate	Measured Rate	Comments
Hex8	2	2	Beta=1.0, Alpha=$1-\sqrt{1-2\nu}$
Tria3	2		
TriaShell	2		
QuadT	2		derived from Tria3
Quad8T	2		derived from Tria3. First order.
Beam2	2		
Tet4	2		
Hex20	4	4	using eigen analysis
Tet10	4	4	using eigen analysis
Tria6	2		derived from Tria3. First order.

4.8.4. RBE3 - comparison with NASTRAN

Verification of the RBE3 pseudo-element necessarily requires comparison with NASTRAN, because no physical model exists. The RBE3 is designed to function like the NASTRAN pseudo element. A simple model was constructed for evaluation of an RBE3 link. The structure consisted of a cube placed on the end of a beam. The beam terminates in the center of the cube, and is connected to the eight corners of the cube with an RBE3 as illustrated in Figure 4-51. The model is named `BoxOnBarRBE3.inp`. The test is `Salinas_rtest/test_tool/fast_regression_tests/mpc/BoxOnBarRBE3.test`.

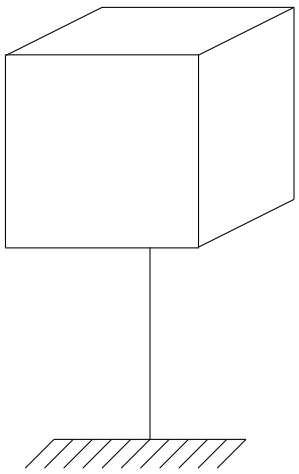


Figure 4-51. – Box on a Bar test object.

There are slight differences in the beam models used by NASTRAN and by **Sierra/SD**. A

summary of the modes is included in the table. As can be seen in the table, the agreement is good. All the modes of the structure are preserved by the RBE3.

#	NASTRAN Frequency	**Sierra/SD** Frequency	Description
1	2354.8	2354.4	1st bending
2	2354.8	2354.4	1st bending
3	6833	6832.7	Pogo Stick, axial mode
4	9942	9939.4	2nd bending
5	9942	9939.4	2nd bending
6	13697	13335	torsion
7	22367	22365	hex deformations
	$> 20,000$	$> 20,000$	hex deformations

4.8.5. Hexshells

The hexshell verification examples were taken from Professor Carlos Felippa, the developer of the element, (see reference[23]). The goal here was to reproduce the results obtained in that report.

4.8.5.1. Example 1

This example corresponds to Section 9.5 in the report,[23] and consists of a circular ring subjected to equal and opposite forces acting along the vertical direction. The exact solution for this problem is given in both reference[53] and reference[49] as

$$\frac{\pi^2 - 8}{4\pi} \frac{PR^3}{EI} \tag{4.8.1}$$

We note that this solution is the total change in diameter for the ring.

By symmetry a quarter ring with appropriate boundary conditions suffices. We note three details for comparing the results to the exact solution. First, the exact solution as given is for the total change in diameter for the ring. For a quarter ring, this result is halved. Second, since the ring is cut at the top surface and we are applying a point load on the symmetry plane, the applied load P will produce twice the deflection in a quarter ring as in the full ring. This is explained in more detail in reference.[49] However, since there is a need to both divide by two and multiply by two, these factors effectively cancel one another out, and thus equation 4.8.1 is the solution for comparison in the case of a quarter ring.

The results obtain by **Sierra/SD** are compared with those of Dr. Felippa in Table 4-48.

For this example, Dr. Felippa also reported results for a two-ply case. Since we do not have an analytical solution to compare with, and since the reported results are normalized by the exact solution, we have no reference point and thus we did not run the two-ply case. We did, however, run a two-ply example where the modulus and Poisson's ratio were the same in both plies. The

Table 4-48. – Normalized Deflections for the Pinched Composite Ring.

N_e	$\frac{R}{h}$ = 20 Felippa	$\frac{R}{h}$ = 20 **Sierra/SD**	$\frac{R}{h}$ = 100 Felippa	$\frac{R}{h}$ = 100 **Sierra/SD**
4	.5746	.5771	.0062	.062
6			.4322	.4376
8	.9582	.9631	.7813	.7971
16	.9896	.9947	.9659	.9886
32	.9955	1.00072	.9753	.9981

results were the same as running a single ply with those same material properties. This provided a weak verification of the multi-ply implementation.

4.8.5.2. Example II

This was the pinched cylindrical shell example (section 9.6). Only one eighth of the shell was considered. The computed results were divided by four to account for the fact that the load was applied to a quarter-section. The results are shown in Table 4-49.

Table 4-49. – Normalized Deflections for the Pinched Cylindrical Shell.

mesh	Felippa	**Sierra/SD**
4x4	.0762	.1
8x8	.2809	.45
16x16	.5366	.81
32x32	.8029	.87
128x128		.897

4.8.5.3. Example III. Scordelis-Lo Roof

A quarter of the roof is modeled. The applied load is a gravity load. The boundary conditions at the rigid diaphragms were incorrectly reported in.[23] The correct ones are $u_x = u_z = 0$. With these conditions, the results as shown in Table 4-50 agree well with the expected values.

Table 4-50. – Normalized Deflections for Scordelis-Lo Roof example.

mesh	Carlos	**Sierra/SD**
2x2	1.2928	1.29
4x4	1.0069	1.011
8x8	.9844	.984
16x16	.9772	.979

4.8.5.4. Example IV

This is the twisted beam model. The normalized results, compared with those of Carlos, are given in Table 4-51.

Table 4-51. – Normalized Deflections pretwisted beam example.

mesh	Carlos		**Sierra/SD**	
	in plane	out of plane	in plane	out of plane
1x6	1.0257	.9778	1.014	.929
2x12	1.0041	.9930	.985	.975

4.8.6. *TriaShells for Composite Modeling*

Laminate composites modeling in **Sierra/SD** is implemented by coupling Allman's triangle[2] with the DKT triangle.[5] Combining these elements together does not capture the coupling that can occur between bending degrees of freedom and membrane degrees of freedom. An additional stiffness that couples these degrees of freedom is generated as documented in References[19] and.[1]

In the next sections we list the results of several verification examples for composite TriaShell elements.

4.8.6.1. Example 1

The first verification example is taken from Reference.[19] A rectangular plate with dimensions 6" x 1" x 0.005" is modeled using 2 triangular elements (Figure 4-52). In Figure 4-52, the left side is clamped (nodes 1 and 3) while node 4 has a unit load in the positive z-direction, and node 2 has a unit load in the negative z direction. Each element is composed of 3 layers. Each layer has the following orthotropic material properties: $E_1 = 10e6$, $E_2 = 0.3e6$, $\nu_{12} = 0.25$, and $G_{12} = 4e6$. The fiber orientation for each layer is $45°, 0°$, and $-45°$, respectively.

This mesh is refined 6 times to create 6 other test cases. The convergence of the displacements and rotations at nodes 2 and 4 is compared with the STRI3 element in ABAQUS as shown in figures 4-53 through 4-58 These figures show that the convergence of the Allman/DKT element is good. Both elements have similar convergence rates as the mesh is refined except the drilling degree of freedom. Figures 4-59 and 4-60 compare the x, y, z, θ_x, θ_y, and θ_z displacements at nodes 2 and 4 (see figure 4-52. Again, the Allman/DKT element compares well with the STRI3 element as the mesh is refined. The exception is the drilling degree of freedom.

The 4th mesh refinement model is stored as a test in the "`Salinas/test_tool/fast_regression_tests/triashell`" subdirectory, and is named "`mesh4_test`".

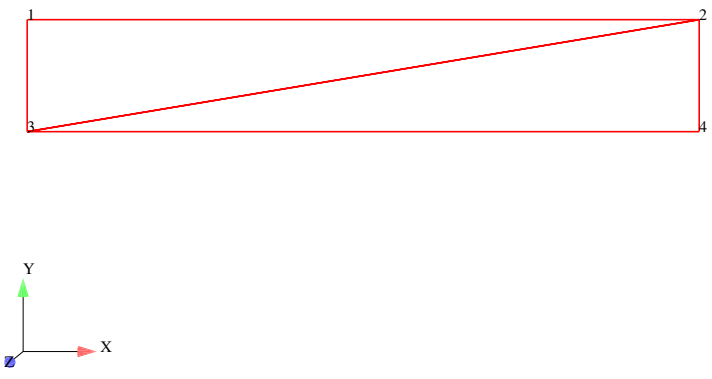


Figure 4-52. – Two Element Test.

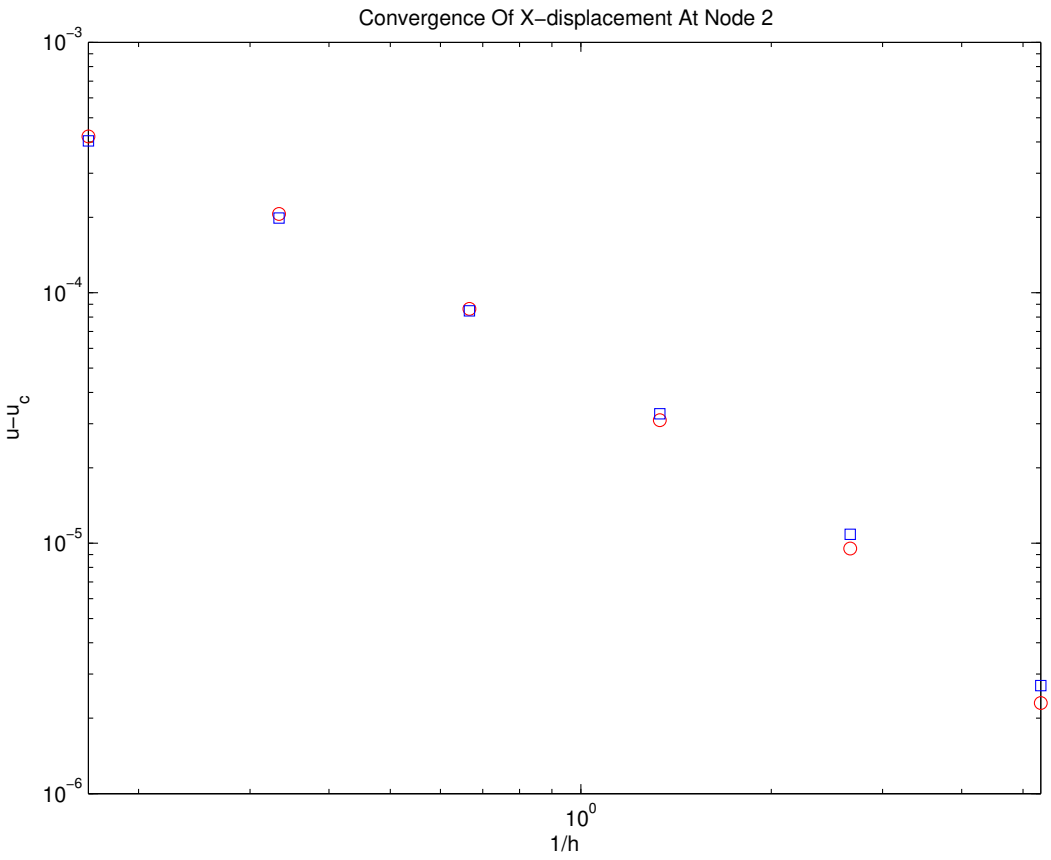


Figure 4-53. – Comparison Of X-displacement Between **Sierra/SD** and ABAQUS.

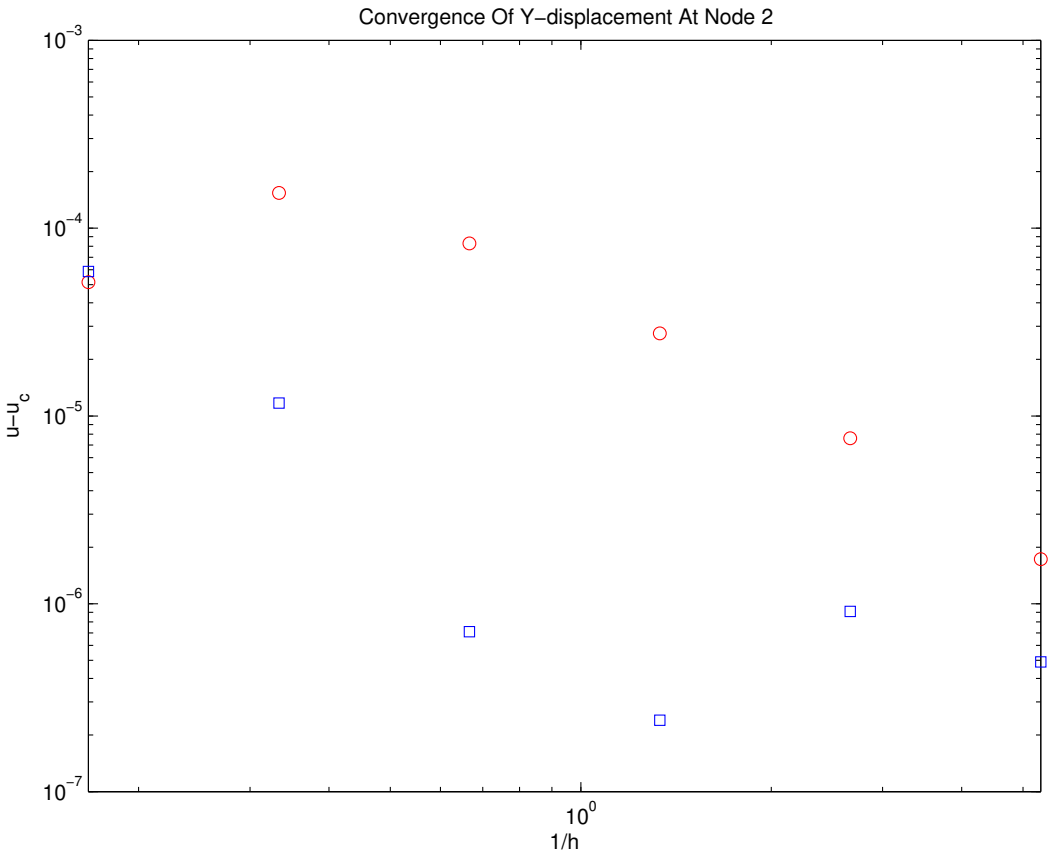


Figure 4-54. – Comparison Of Y-displacement Between **Sierra/SD** And ABAQUS.

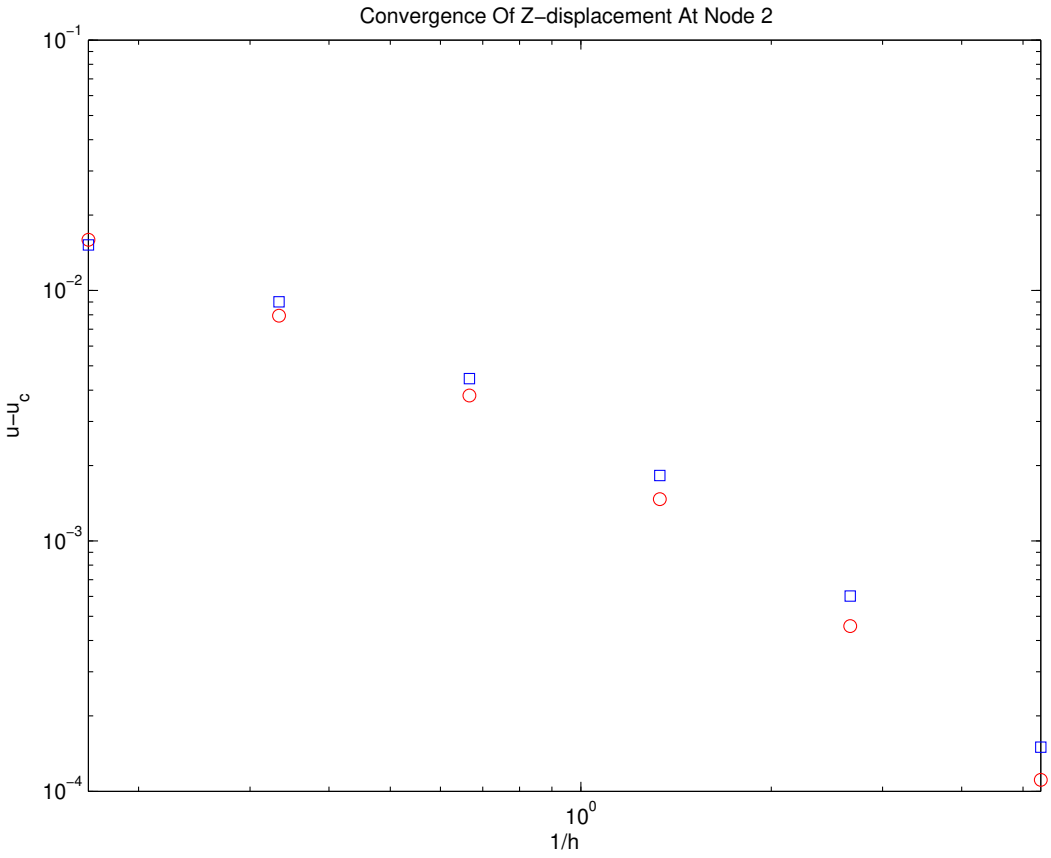


Figure 4-55. – Comparison Of Z-displacement Between **Sierra/SD** And ABAQUS

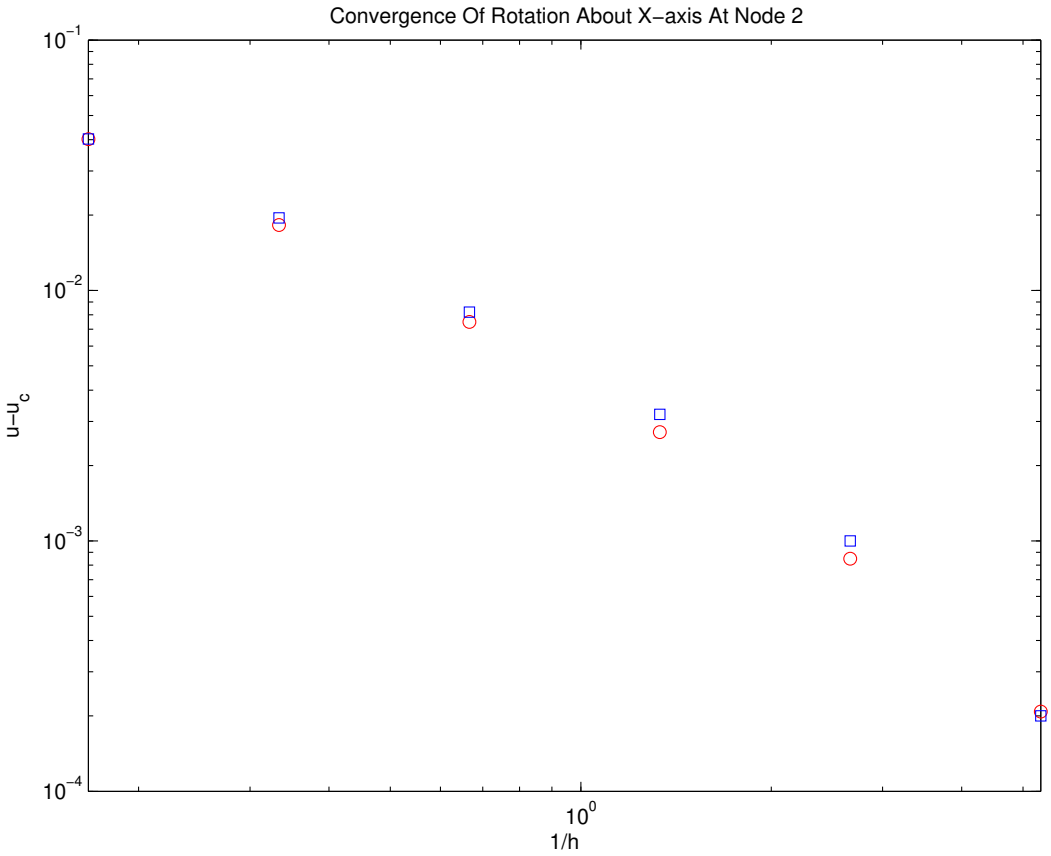


Figure 4-56. – Comparison Of Rotation About X-axis Between **Sierra/SD** And ABAQUS

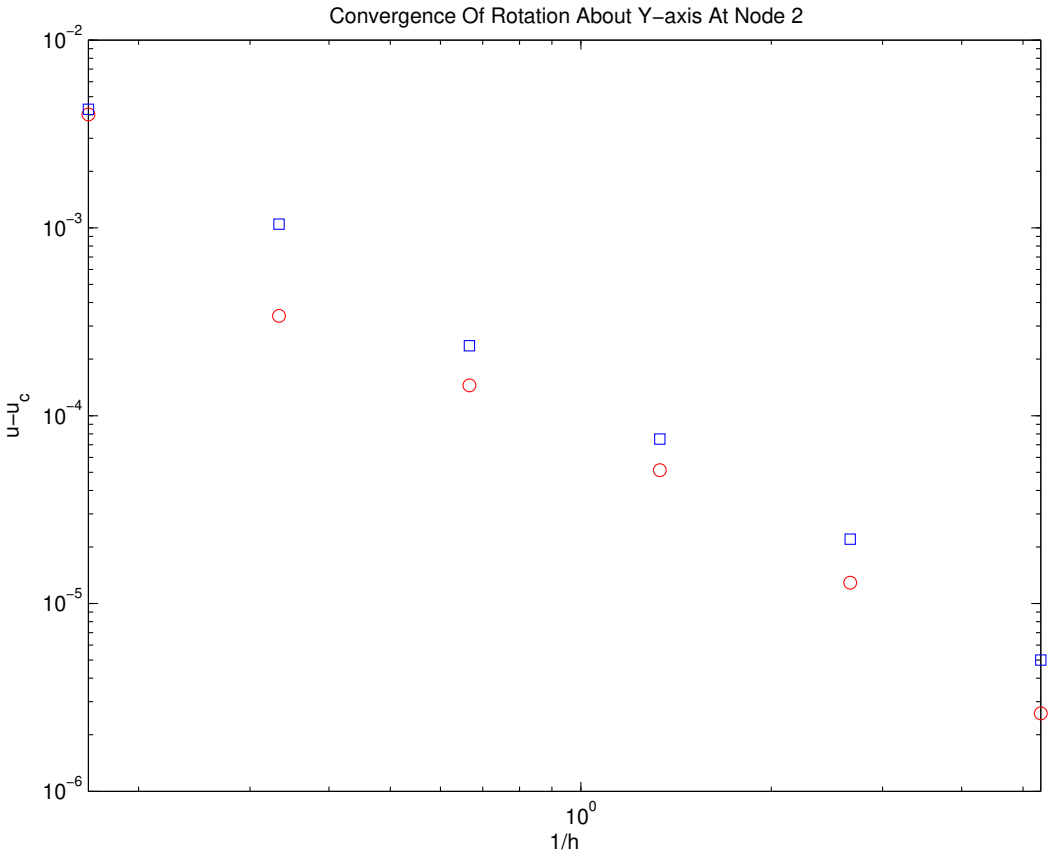


Figure 4-57. – Comparison Of Rotation About Y-axis Between **Sierra/SD** And ABAQUS

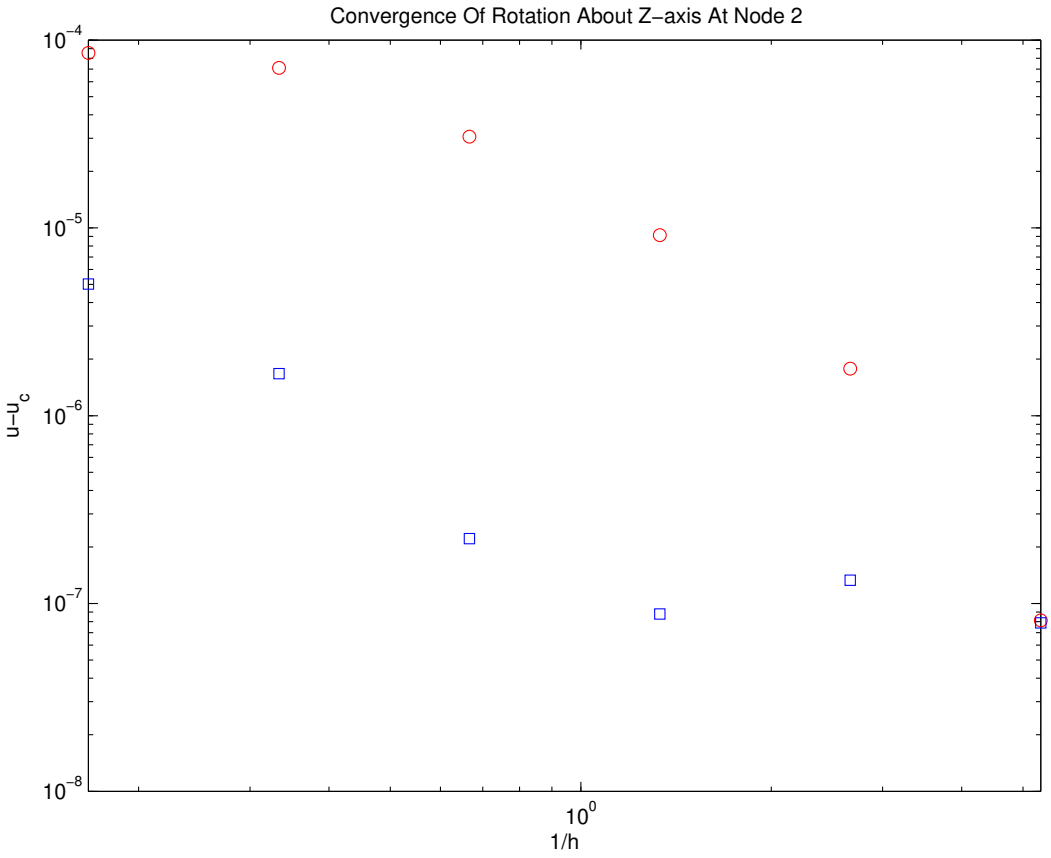


Figure 4-58. – Comparison Of Rotation About Z-axis Between **Sierra/SD** And ABAQUS

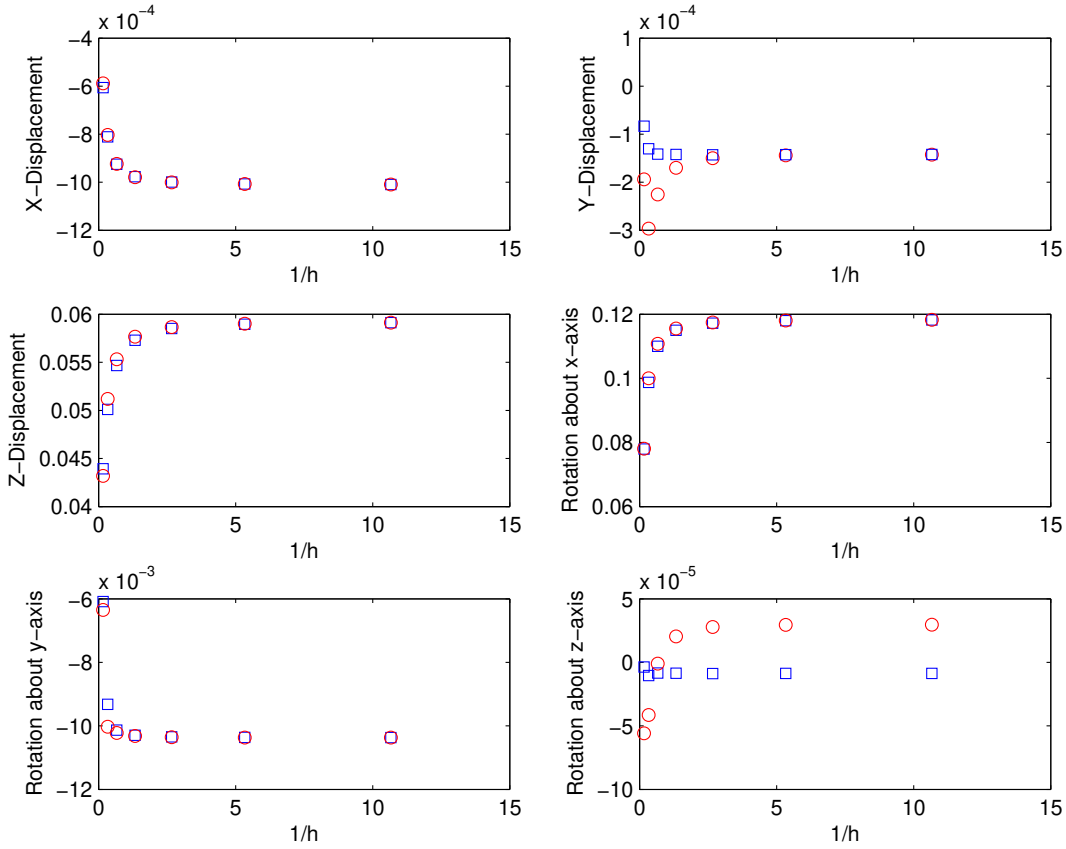


Figure 4-59. – Convergence Of Displacements and Rotations At Node 2.

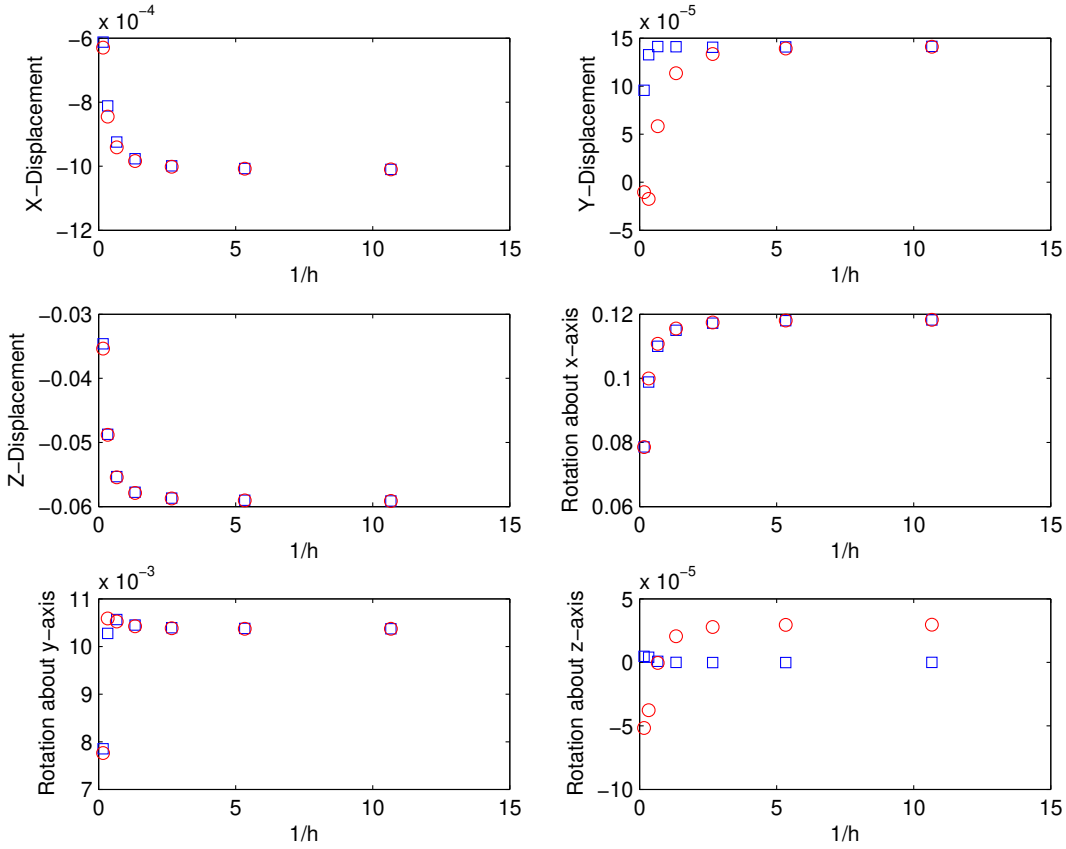


Figure 4-60. – Convergence Of Displacements And Rotations At Node 4.

4.8.6.2. Example II

The second verification example for laminate composite modeling is taken from Reference.[27] A rectangular plate is subjected to a uniform pressure load of q = 0.003 psi. The plate, shown in Figure 4-61 has dimensions 12 in. x 8 in. and is simply supported on each edge. The alternating angle-ply stacking sequence is [-30/30 -30/30 -30/30 -30/30]. Each layer has a thickness of 0.01 in. The orthotropic material properties for each layer are: $E_1 = 26.25e6psi$, $E_2 = 1.49e6psi$, $nu_{12} = 0.28$, and $G_{12} = 1.04e6$ psi.

The transverse displacement at the center of the plate is compared with the analytical solution developed in reference.[27] **Sierra/SD** calculates a value of -2.377e-4, while the analytical solution is -2.38e-4. Again, the DKT/Allman triangle produces a good comparison with the analytical solution.

This test is kept in the Salinas_test repository in the verification/composite subdirectory and is named plate_test.

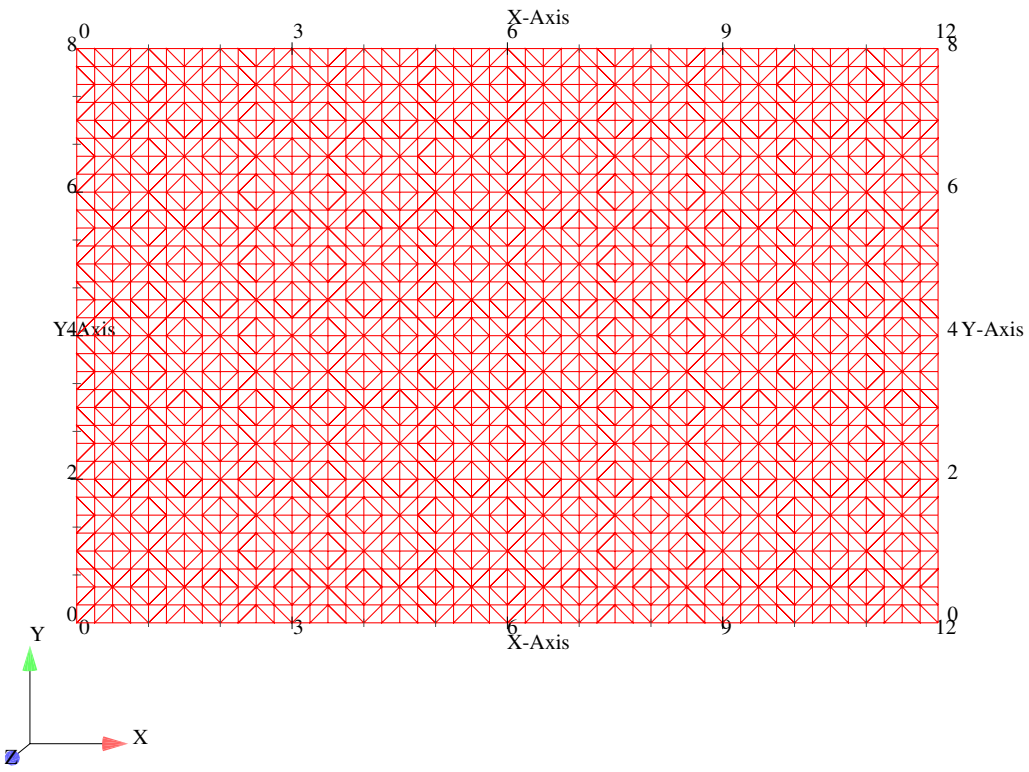


Figure 4-61. – Finite Element Model Of A Flat Plate.

4.8.6.3. Example III

This verification example for laminate composite modeling is also taken from Reference.[27] A cylindrical panel is subjected to a uniform pressure load of q = 0.003 psi. The cylindrical panel ($\frac{1}{4}$ of model is shown in Figure 4-62) has a length of 80 in., while the arc length of the other side is 41.89 in. corresponding to an angle of $\phi = 24°$ and radius of 100 in. The stacking sequence is [0/90/90/0]. Each layer has a thickness of 0.08 in. The orthotropic material properties for each layer are: $E_1 = 18e6psi$, $E_2 = 1.4e6psi$, $nu_{12} = 0.34$, and $G_{12} = 0.9e6$ psi.

The transverse displacement of the free corner is compared with the analytical solution developed in reference.[27] **Sierra/SD** calculates a value of 6.958e-4, while the analytical solution is 6.945e-4. Again, the DKT/Allman triangle produces a good comparison with the analytical solution.

This test is kept in the Salinas_test repository in the verification/composite subdirectory and is named cyl_panel_test.

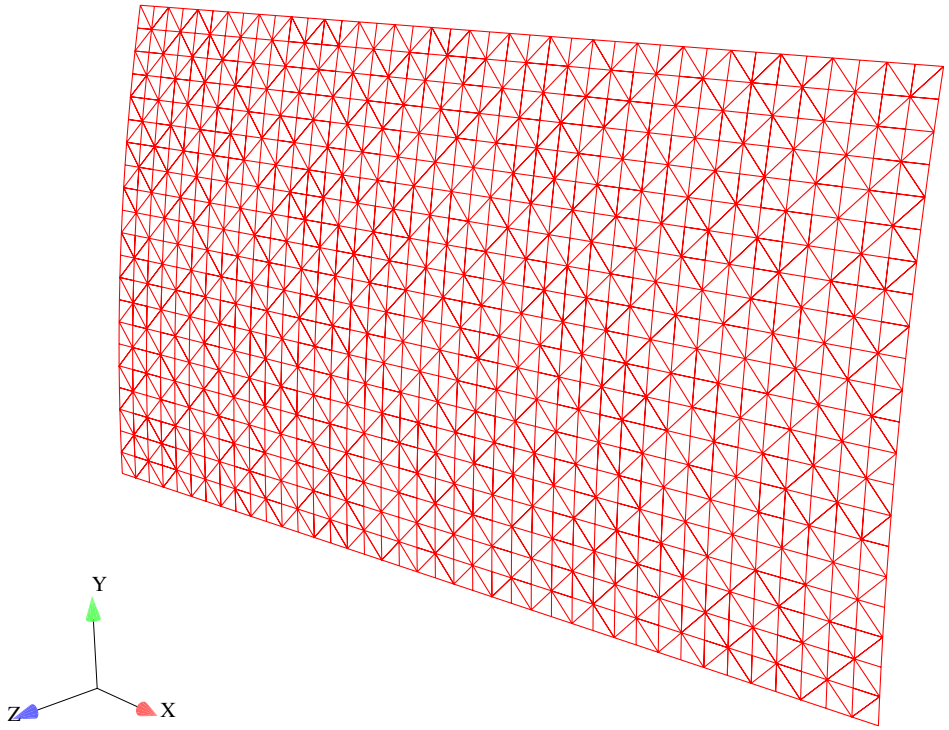


Figure 4-62. – Finite Element Model Of A Cylindrical Panel.

4.8.7. Joint Modeling: Joint2g Element with Iwan Constitutive Model

The Joint2g element permits independent specification of the constitutive relations between each of the relative displacements. Currently, the most prominent of the constitutive equations employed for the "whole joint" modeling approach is the 4 parameter Iwan model. The Joint2g element and the Iwan constitutive model are documented in *User's Manual* and *Sandia National Labs* reports specifically addressing the 4 parameter model.

There exists a closed form expression for the energy dissipation per cycle resulting from harmonic excitation imposed on a joint of this nature. That expression[45] is,

$$D = r^{\chi+3} \frac{4F_s\phi_{max}(\chi+1)}{(\beta + \frac{\chi+1}{\chi+2}(\chi+2)(\chi+3)} \tag{4.8.2}$$

where β, χ, ϕ_{max}, and F_s are model parameters, and r satisfies

$$\frac{F_o}{F_s} = r\frac{(\beta+1) - r^{\chi+1}/(\chi+2)}{\beta + (\chi+1)/(\chi+2)}, \tag{4.8.3}$$

where F_o is the amplitude of the harmonic excitation. Comparison of the exact solution and **Sierra/SD** predictions is presented in Figure 4-63.

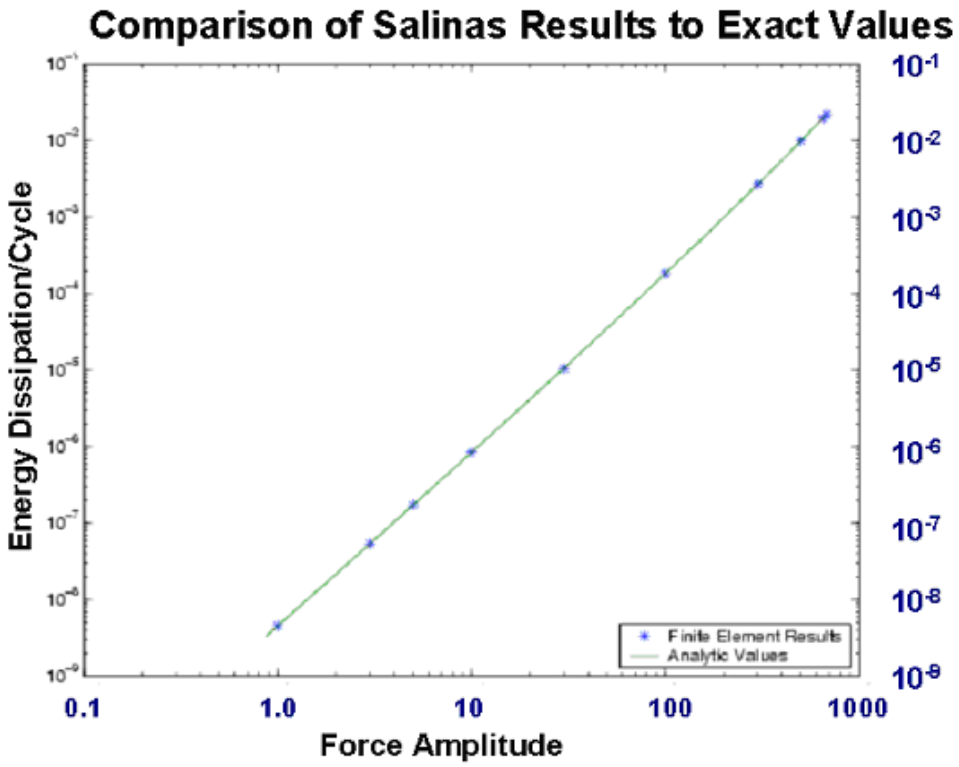


Figure 4-63. – **Sierra/SD** Iwan Element: Comparison to Analytic Solution.
The **Sierra/SD** predictions for unidirectional load on a simple joint agrees with the exact solutions.

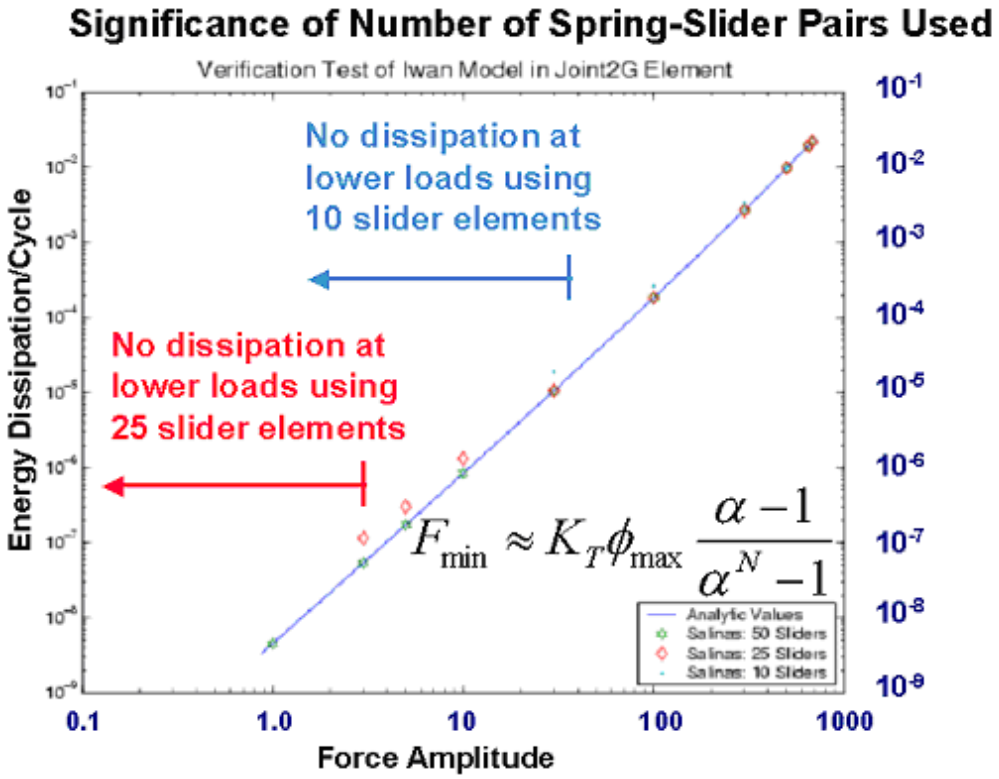


Figure 4-64. – Significance of Number of Spring-Slider Pairs Used.

The number of spring-slider pairs necessary to demonstrate sensitivity to given levels for load in **Sierra/SD** is that predicted by theory.

$$F_{min} \approx K_T \phi_{max} \frac{\alpha - 1}{\alpha^N - 1}$$

There is one integration parameter in **Sierra/SD**, the number of spring-slider pairs used to approximate the continuous distribution of Jenkins elements. The relevant SAND report provides guidance as to the number of elements necessary to manifest proper dissipative response to loads of given size. Figure 4-64 shows that desired accuracy is achieved with the number spring slider pairs predicted by theory.

4.8.7.1. Iwan Macroslip

To evaluate the Iwan model in **Sierra/SD** when it hits macro-slip, a 1D MATLAB test case involving macro-slip and simple dynamics was developed. It was compared with the results of the corresponding 1D Sierra/SD analysis. Here is a sketch of the model.

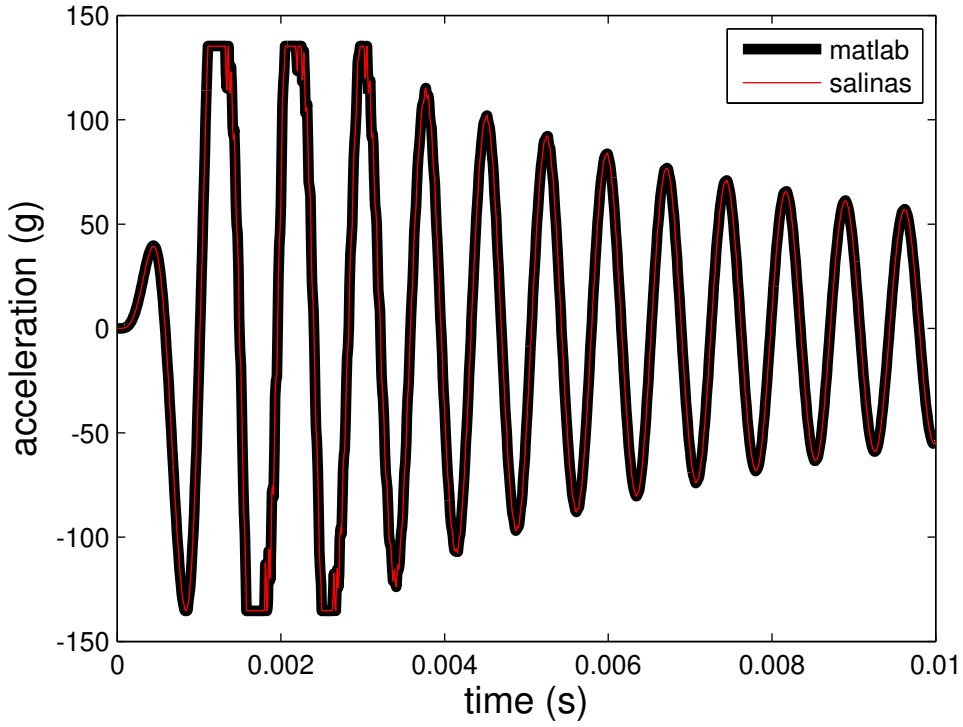
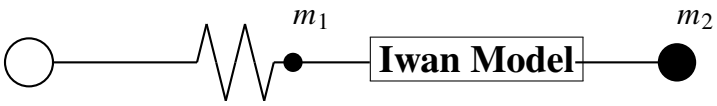


Figure 4-65. – MATLAB and **Sierra/SD** calculation of M_2 acceleration.



source k=9.74e5

Here the source is a 100g wavelet base excitation, $m_1 = 0.05\,lb$, and $m_2 = 4\,lb$.

Analysis is performed both within **Sierra/SD** and MATLAB. The acceleration of the four pound mass for each analysis method is compared in figure 4-65. We see agreement, though the MATLAB result better resolves macro-slip.

The stretch of the Iwan joint is another good indicator of agreement, and is shown in Figure 4-66. The stretch is the relative displacement across the *Iwan* element. Again, the agreement is good, but not perfect.

In both analyses, the acceleration of the spring mass shows significant high frequency response (or hash) as shown in Figure 4-67. The high frequency noise is undesirable, but is a feature of the model constructed of a finite number of slider/spring elements. As the elements begin to slide, high frequency noise is generated.

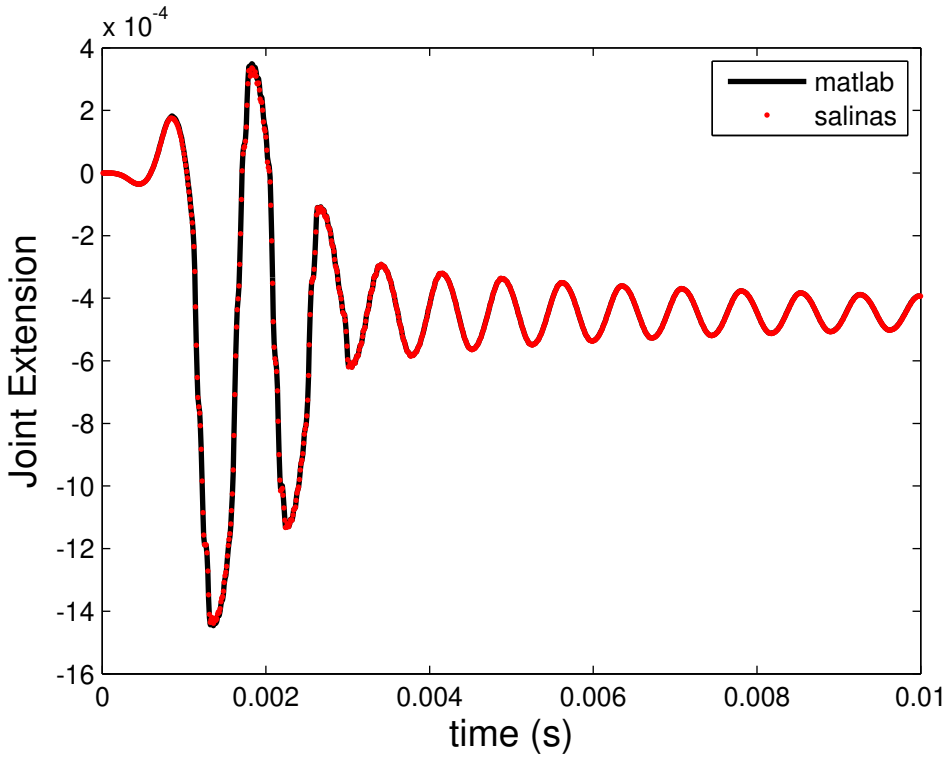


Figure 4-66. – MATLAB and **Sierra/SD** calculation of joint extension.

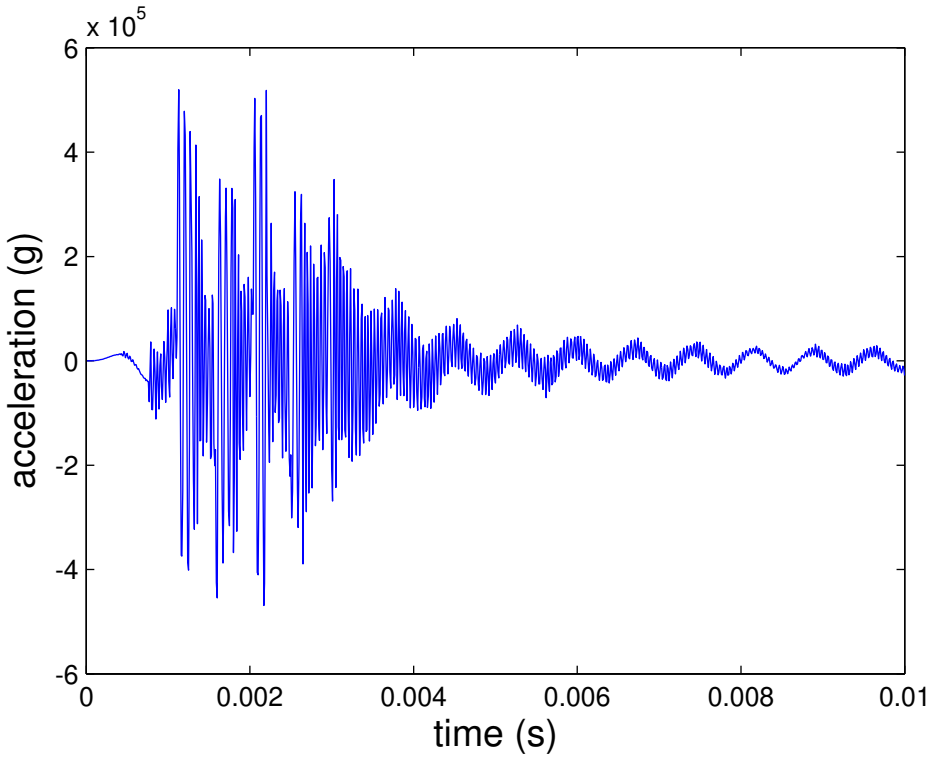


Figure 4-67. – Sierra/SD calculation of M_1 acceleration.

4.8.8. Membranes

Membrane elements are similar to shells, except that they have no rotational degrees of freedom, and have no out-of-plane stiffness in the unstressed state. When they are pulled in tension, an out-of-plane stiffness appears, and takes the form of a geometric stiffening. In the following test cases, we examine the response of the membrane element to both in-plane and out-of-plane deformation. We consider these two loading cases separately.

The first example consists of a square membrane of dimension $1x1$, which is subjected to a uniform tension T in both in-plane directions. After the application of the tension, the membrane boundaries are either fixed, or placed on rollers, and an modal analysis is performed about the stressed state. Since these elements are intended to be used in transfers between Adagio and **Sierra/SD**, we perform the static preload in Adagio, and then transfer the stresses and displacements to **Sierra/SD**. In this way, we also exercise the transfer capabilities for these elements.

The exact eigenvalues for stretched square membranes are given in.[32] In the case of a membrane that is clamped along all boundaries, the frequencies are

$$f_{nm} = \frac{\omega_{nm}}{2\pi} = \frac{c}{2}\sqrt{\left(\frac{n}{L_x}\right)^2 + \left(\frac{m}{L_y}\right)^2} \tag{4.8.4}$$

where $c = \sqrt{\frac{T}{\rho_s}}$ is the speed of sound in the membrane, T is the tension per unit length in the membrane, and ρ_s is the surface density. Note that in the case of a square membrane $L_x = L_y$. Also, the indices $m = 1, 2, 3, \ldots$ and $n = 1, 2, 3, \ldots$. In the case of a free-free membrane, the expression for the frequencies is the same, except that both m and n start at 0. In this way, they allow for a rigid body mode.

Table 4-52 shows a comparison of the first three exact and computed eigenvalues of the square clamped membrane, and Table 4-53 shows the same for the free-free membrane. In both cases, good agreement is seen. For the free-free case, we do not compare rigid body modes in the table, but we verified that they came out to be numerically zero. Note that for both cases, repeated modes are observed.

The Tempo application and its tests have been removed.

For in-plane loading, there are 2 verification tests located at

exact (Hz)	computed
13.178	13.230
20.83	21.126
20.83	21.126

Table 4-52. – Eigenvalue convergence for a fixed-fixed, prestressed membrane. The values given are the natural frequencies, in Hz.

exact (Hz)	computed
9.3169	9.3553
13.178	13.230
18.634	18.941
18.634	18.941

Table 4-53. – Eigenvalue convergence for a free-free, prestressed membrane. The values given are the natural frequencies, in Hz.

```
Salinas_test/patchtests/quadt/quadt-patch8_test
Salinas_test/patchtests/quadt/quadt-patch9_test
```

These tests use in-plane tension, and verify the corresponding deformation of the membrane.

4.8.9. Tied Joint

The tied joint provides a means of connecting two surfaces together while allowing compliance in the shear behavior. The tied joint allows more flexibility in the specification of the normal behavior than previous methods that required a fully rigid surface pair to which a whole joint model (such as a Joint2g) is attached.

A first step in developing the tied joint is replicating the old model behavior. This is done with the two test cases “2x2tied” and “2x2whole”. The first of these couples a block of elements using the new methodology. The “2x2whole” example uses the old approach. The solutions are shown to be identical.

Next, we present transient simulations on a single-leg model. This single leg model was taken from a more complicated three-leg model. The surfaces that join the two pieces are modeled with a tied joint, and then we compare those results with a truth model where the constraints on the interface were implemented manually using the “old" approach of an RBE3 element.

The first example compares the two approaches in the case when the tied joint model is modeled with the following block

```
TIED JOINT
    normal definition = slip
    side = free
         . . .
END
```

Figures 4-68, 4-69, 4-70 shows the comparison of the X, Y, and Z displacements as a function of time, for the tied joint and truth models. Excellent agreement is observed.

The second example compares the tied joint and truth model approaches when the tied joint model is modeled with the following block

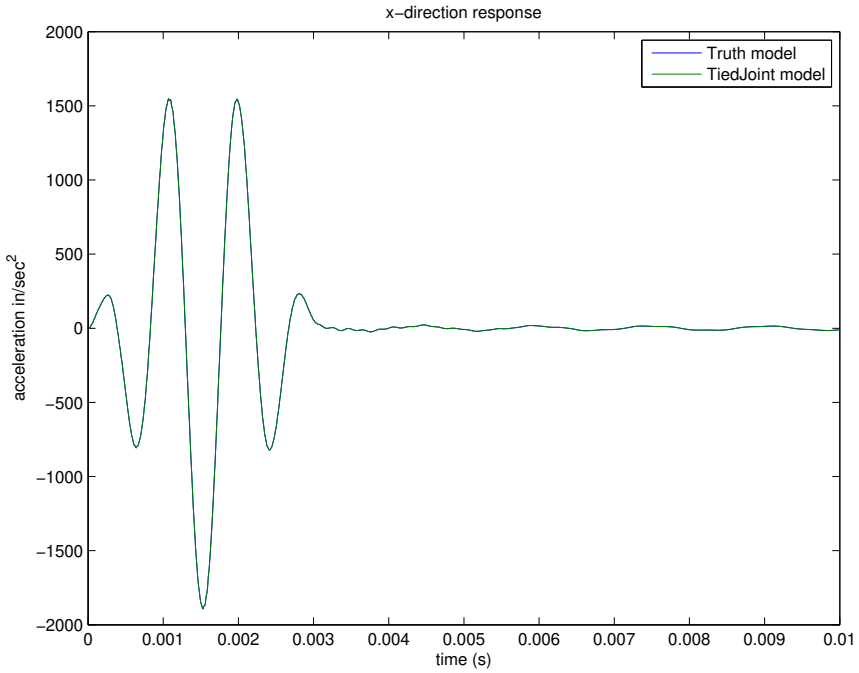


Figure 4-68. – X displacement comparison for tied joint versus truth model, tied=slip, side=free

```
TIED JOINT
     normal definition = none
     side = rigid
        . . .
END
```

Figures 4-71, 4-72, 4-73 shows the comparison of the X, Y, and Z displacements as a function of time for this case, for the tied joint and truth models. Excellent agreement is observed.

These tests are located in the verification test suite in the directory

```
Salinas_rtest/verification/tiedjoint
```

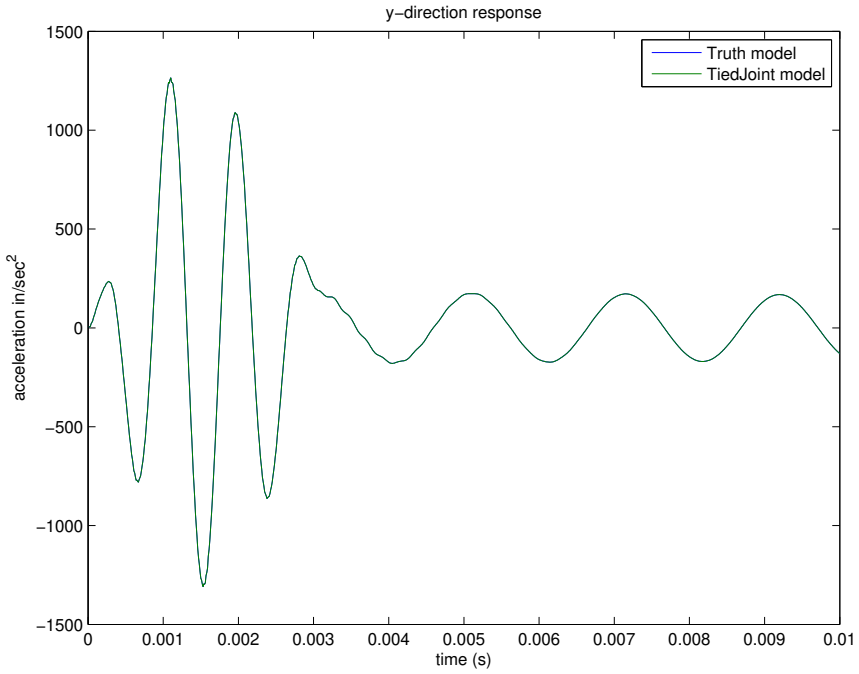


Figure 4-69. – Y displacement comparison for tied joint versus truth model, tied=slip, side=free

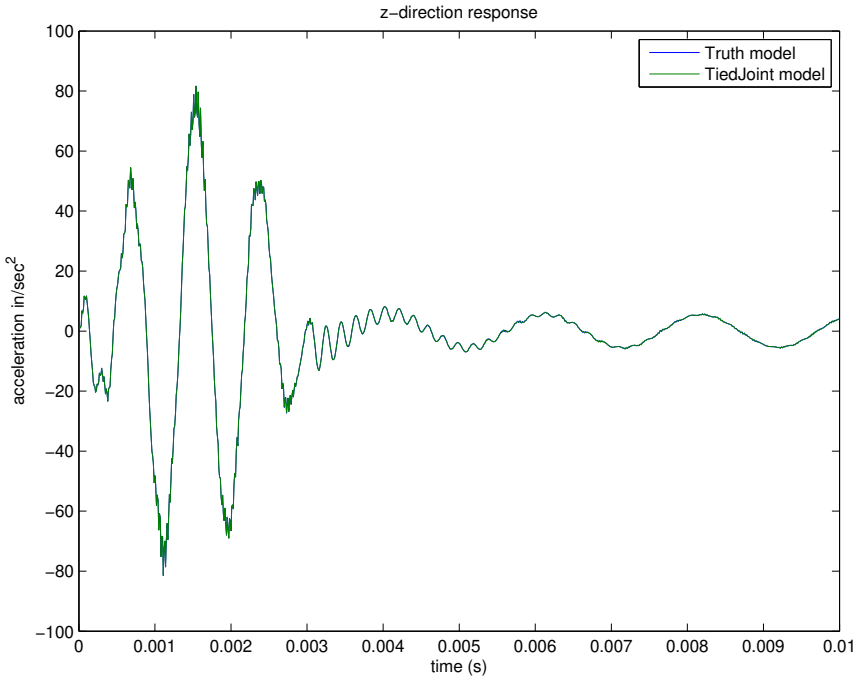


Figure 4-70. – Z displacement comparison for tied joint versus truth model, tied=slip, side=free

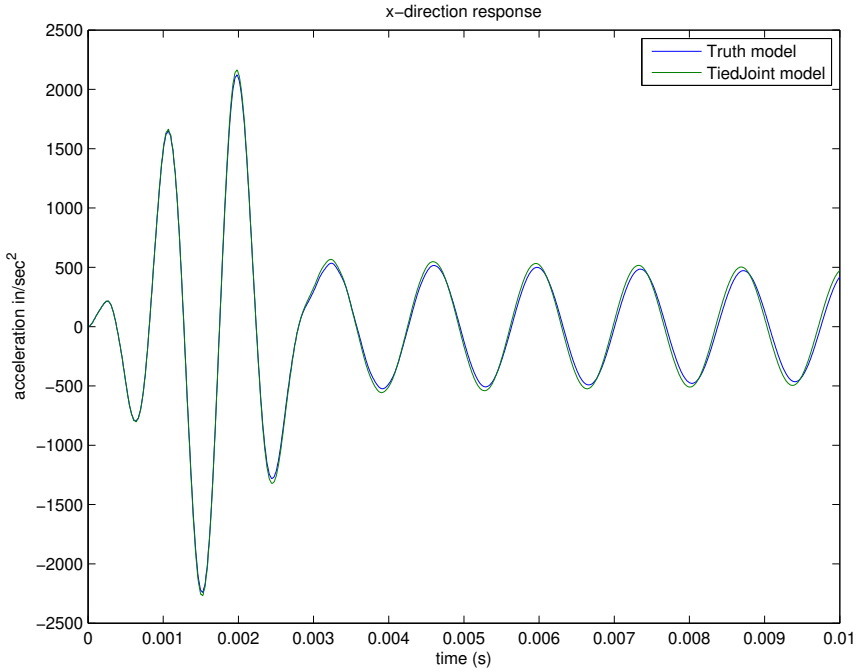


Figure 4-71. – X displacement comparison for tied joint versus truth model, tied=none, side=rigid

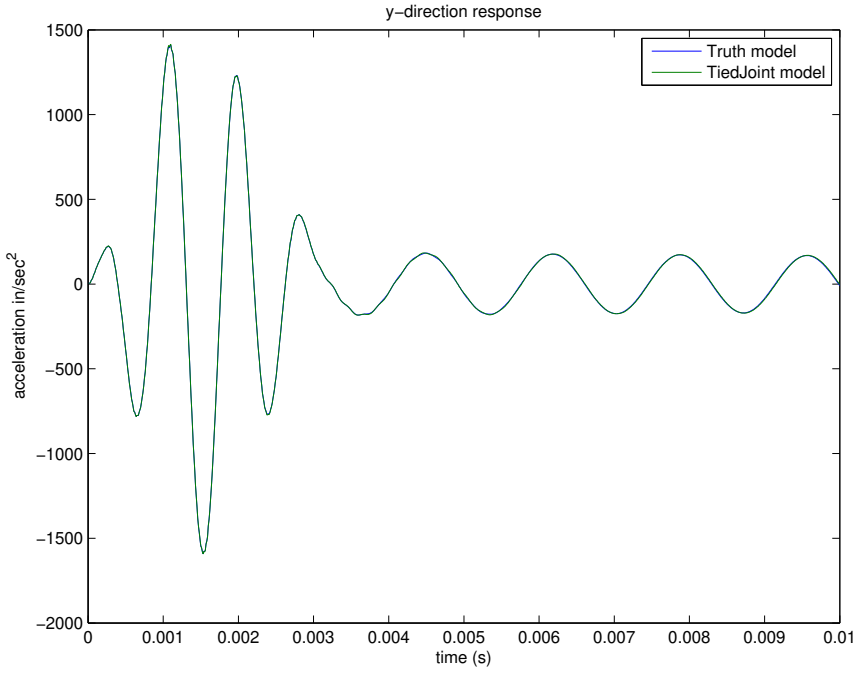


Figure 4-72. – Y displacement comparison for tied joint versus truth model, tied=none, side=rigid

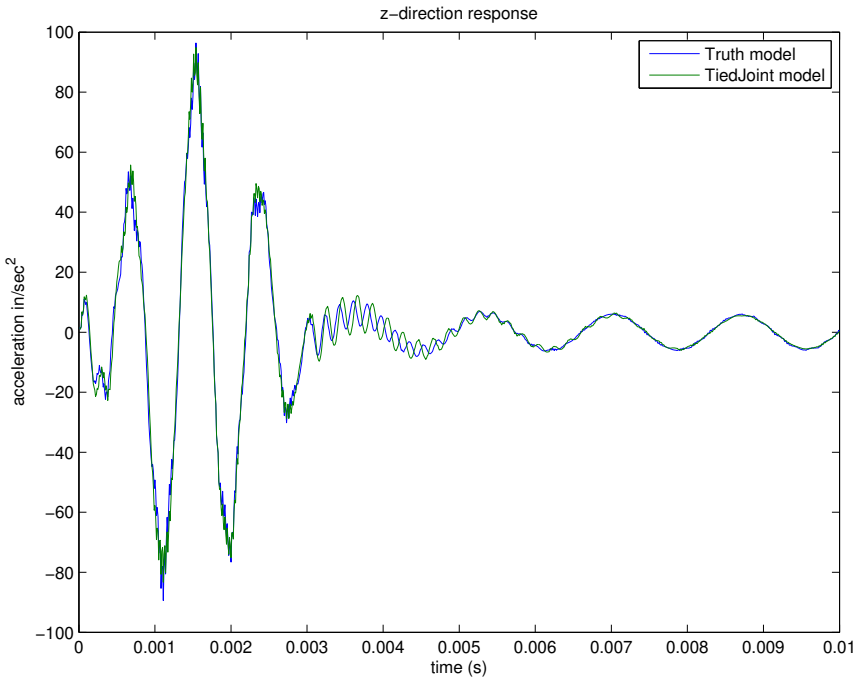


Figure 4-73. – Z displacement comparison for tied joint versus truth model, tied=none, side=rigid

4.8.10. Rrodset

The Rrodset mimics a material like a fiber textile in that it does not oppose bending in any way, but it does oppose tension. Fundamentally, it is identical to placing a collection of Rrods along every edge of a surface. One use is to distribute the shear loading of a tied joint.

To verify that it does not oppose bending, a simple example with 2 connected plates on top of each other was created. The left side was fixed and the right side had loads applied. The top plate was pulled while the bottom plate was pushed with equal force, causing a pivot around the center where the Rrodset can be placed. It was shown that a statics solution produced the same results with or without an Rrodset in the middle. The test is in the fast regression tests suite and is called `Rrodset.` The test is `Salinas_rtest/test_tool/fast_regression_tests/traction/rrodset.test`.

5. BOUNDARY CONDITIONS

5.1. Parallel Distribution of Load through Rbars

The purpose of the verification is to ensure that loads may be properly distributed through a "spider" collection of Rbar elements onto a concentrated mass. The model is shown in Figure 5-1. This is a model of a conmass connected to a hex by spiders using Rbars. Verification that the model works the same running with one processor or six processors.

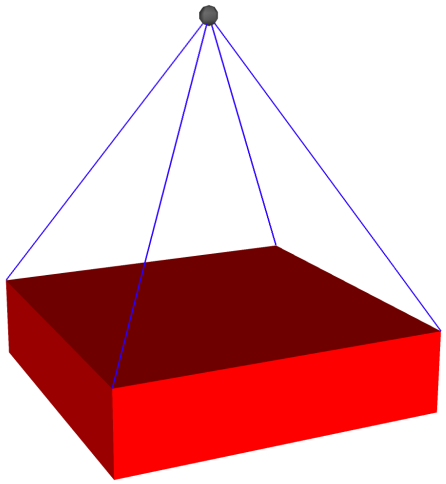


Figure 5-1. – Model for Parallel Distribution of Load through Rbars.

For input deck see Appendix 9.53.

5.2. Perfectly Matched Layers: Offset Sphere

In this section, we describe the verification of the offset sphere problem for the Ellipsoidal PML formulation. Further background, verification, results, and implications are available in [11]. An acoustic source is placed asymmetrically in a spherical domain.

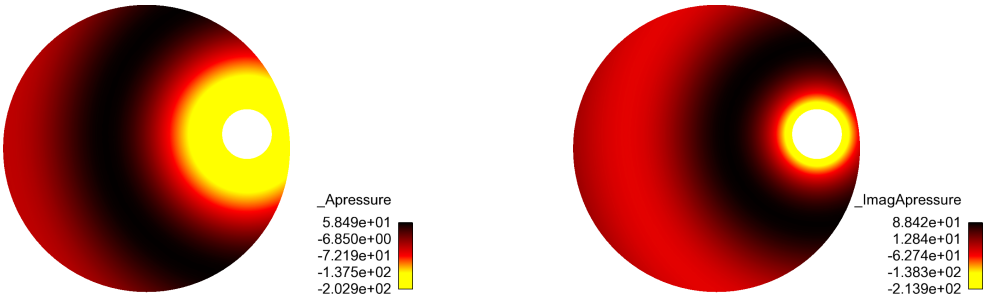


Figure 5-2. – Offset Sphere at 50 Hz.

Figure 5-2 shows the numerical solution for the offset sphere problem. The radius of the outer sphere is 5 meters, and the radius of the inner sphere is 1 meter. An acoustic velocity of $V_0 = 1$ is applied to the normal surface of the inner sphere, to create a monopole excitation. The sphere is composed of 850,000 TET4 elements, and 145,000 nodes. The material modeled is air, where $\rho = 1.293 \frac{\text{kg}}{\text{m}^3}$, $c_0 = 332.0 \frac{m}{s}$.

A 2D representation of the spherical result cut along the plane y=0 is shown. Note that the solution is spherically symmetric about the acoustic source. The exact solution is given as

$$P(r) = \frac{iV_0\Omega\rho a^2}{r(1+ika)} e^{ik(r-a)} \tag{5.2.1}$$

where r is the distance from the center of the inner sphere to a point in the mesh, and a is the radius of the inner sphere.

The relationship between the thickness of the PML boundary, the discretization of the elements within the PML boundary, and the selection of loss parameters is investigated on the Offset Sphere example. The discrete L^2 error norm of the solution at every degree of freedom is compared between the PML formulations, the absorbing boundary conditions, and infinite elements of various orders. We also examine the performance of the iterative solver on these problems, and compare the effects of PML and infinite elements on linear solver performance.

Figure 5-3 shows the results for the offset sphere at a frequency of 50Hz. For this case, the outgoing waves are not perpendicular to the boundary surface, and the spherical wave absorbing boundary condition gives very inaccurate results. The infinite element solution has converged around order 4, and the remaining error compared to the analytic solution corresponds to the discretization error for the mesh. Both the ellipsoidal and spherical PML formulations converge to the discretization error of the mesh. The PML layer converged with 12 layers of elements, a loss parameter of 600, and a thickness of 2 meters. Figure 5-4 shows the magnitude of acoustic pressure in the PML layer of the offset sphere, showing the rapid decay to zero magnitude towards

the outer-most boundary of the PML layer. Note that the ellipsoidal PML formulation is the only supported or accessible formulation in SierraSD.

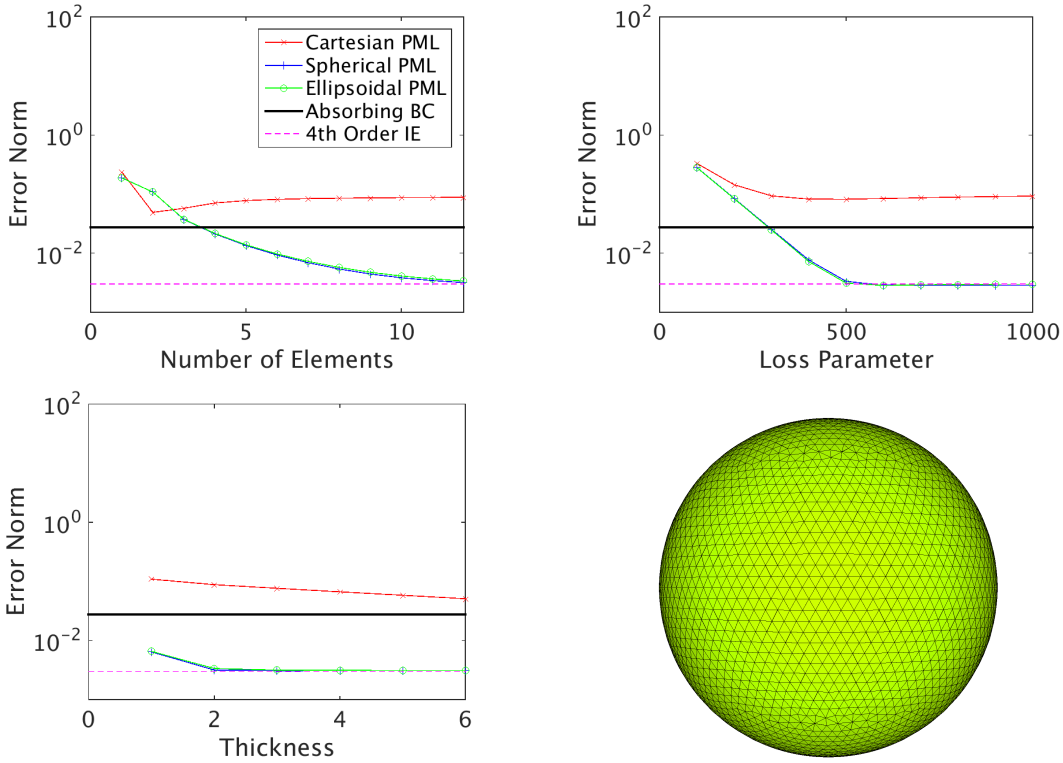


Figure 5-3. – Parameter Studies for OffsetSphere (50 Hz). Note: Ellipsoidal PML is the only supported capability, Cartesian and Spherical have been removed.

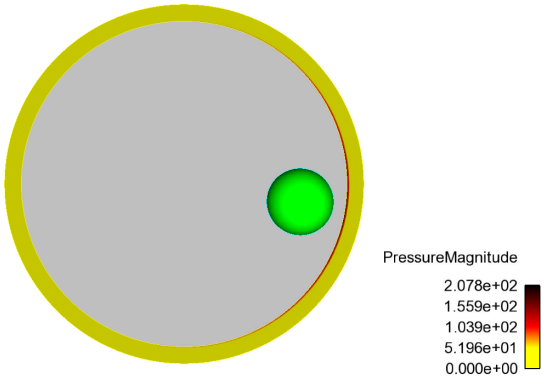


Figure 5-4. – Acoustic Pressure in PML Layer for offset sphere, showing the rapid decay to zero magnitude near the outermost boundary of the PML layer.

For input deck see Appendix 9.54.

5.3. Periodic Boundary Conditions

In material characterization through simulation of representative volume elements, periodic boundary conditions are needed on the opposite faces, with imposed stretch and/or distortion. Similarly, in the context of phononic crystals and acoustic/elastic metamaterials, imposing periodic boundary conditions is a key functionality needed for computation of dispersion curves and band structure. Sierra-SD facilitates the imposition of such periodic boundary conditions.

In this section, we provide the verification of the capability by simulating an infinite bar with linear array of spherical voids, with imposed overall tensile strain. We do not attempt to compare with any analytical/reference solutions, but confirm the consistency of results from applying periodic boundary conditions in two different ways. Specifically, we consider an infinite bar of unit (1x1) square cross-section, with spherical voids of radius 0.4, dispersed uniformly with unit spacing. A global strain of 0.015 is applied along the axis (x direction), and the resulting stresses are to be analyzed (the Young's modulus is 1e4). Such an analysis can be carried out by modeling a periodic cell, which can be any 1x1x1 block along the length of the bar. Correctly implemented periodic boundary conditions must give the same results independent of the choice of the periodic cell. Given this, we compare the results from analyses of two separate periodic cells, one with the spherical void at the center of the periodic cell, and the other with the periodic cell boundaries on both ends cutting through the centers of two adjacent voids. A differential x-displacement of 0.015 units is applied between two edges of the periodic cells to simulate the global strain of 0.015. Rigid body displacements are eliminated through appropriate statically determinate boundary conditions on the center section of the cell. Figure 5-5 contains the meshes for the two periodic cells. Note that each half has identical meshes, indicating that identical discrete systems are being solved, thus eliminating the role of the discretization error and leading to the expectation of almost exact match.

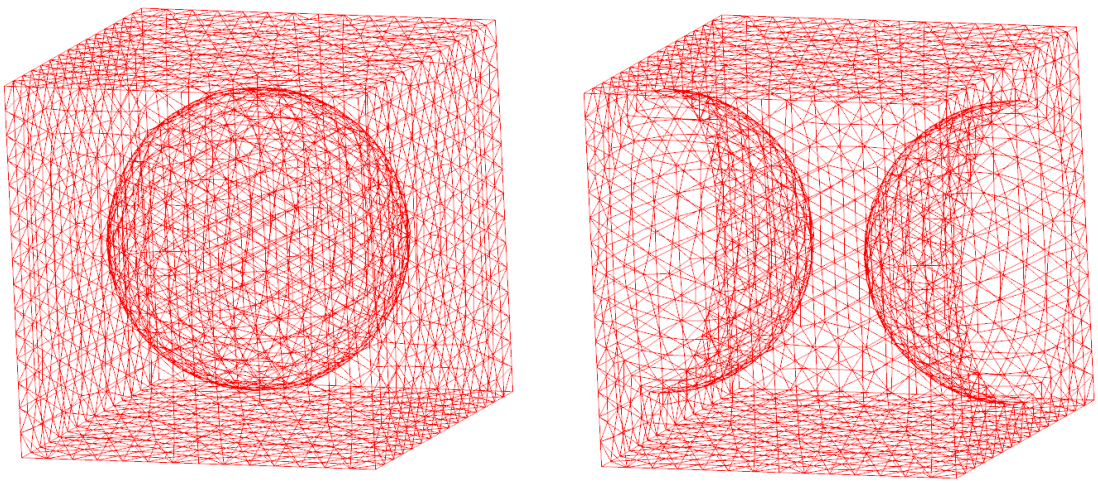


Figure 5-5. – Meshes for two different periodic cells.

The stresses are examined at three different locations on the surface of the spherical void, at the intersection with x,y and z axes respectively. To be precise, the results are obtained at element

centroids closest to the three locations, which are shown in table 5-1. Only one set of results is shown since the computed stresses are identical between the two models, up to 10 significant digits, clearly verifying the implementation of periodic boundary conditions.

Table 5-1. – Stresses near the surface at points cutting various axes.

Stress	y axis	z axis	x axis
σ_{xx}	281.0494288	273.0301545	-1.7271636
σ_{yy}	-8.0119751	0.5224045	-0.2133664
σ_{zz}	-0.2935124	23.5932137	-0.8396841
σ_{xy}	14.8824683	14.2929051	-9.3081820
σ_{yz}	-0.3275614	0.3518104	2.0719204
σ_{xz}	16.8660510	28.2491021	-9.8072644

In addition to the above example, we tested the implementation on homogeneous block with straight and curved surfaces under uniform stretch, resulting in expected uniform stress state with correct values. The details are not presented in this document, but can be found in the test repository. For input see Appendix 9.55

5.4. Multi-directional Periodic BC: Periodic Volume Elements

Representative volume element (RVE) modeling is a standard approach in computing macroscopic materials properties from materials with microstructure. In the context of regular periodic microstructure, RVE reduces to a periodic unit cell of the microstructure, called the periodic volume element (PVE). The boundary conditions for modeling a PVE of a 3D solid would be periodic boundary conditions in all three directions, or in all the directions in which the PVE repeats. This requires multiple begin-periodic blocks, each connecting the faces on the opposite surfaces. The surfaces in one begin-periodic would intersect with surfaces in other periodic blocks, thus testing the associated functionality in Salinas.

We consider the example in Section 5.4, and expand to 3D setting. Specifically, we consider a homogeneous matrix with regularly spaced spherical inclusions in all three directions. Two of the many ways to define a PVE are evaluated, both cubic in shape. In the first PVE, the void is at the center of the cube, while in the second, the void is split into eight quarters, each centered at each of the vertices of the cube. The schematic of the idea, in 2D settings, is illustrated in Figure 5-6. The actual, discretized PVEs are shown in Figure 5-7, where the meshing is done to consistently eliminate the differences due to discretization.

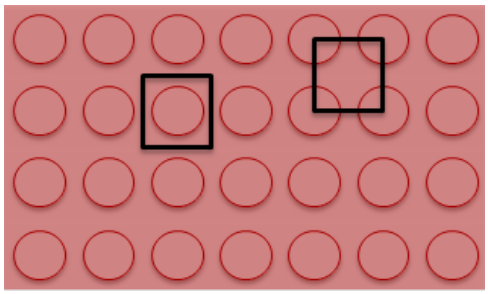


Figure 5-6. – 2D schematic of the two simulated periodic volume elements (PVEs).

Both PVEs are subjected to the same global strain, or equivalently symmetric deformation gradient:

$$\epsilon = \nabla \mathbf{u} = \begin{bmatrix} -1.50 & 1.00 & 0.50 \\ 1.00 & -1.00 & 0.25 \\ 0.50 & 0.25 & -0.50 \end{bmatrix} \tag{5.4.1}$$

Note that since the entire strain is associated with deformation gradient, implicitly, there is no (global) rotation of the PVE. Relative displacement vector for each begin-periodic block is determined by the above tensor applied on the geometric offset vector. Since the geometric offsets are unit vectors in x, y and z directions for each of the three begin-periodic blocks, the relative displacements are essentially the three columns of the deformation gradient (see the input file). Note that the imposition periodic BC in three different directions automatically prevent rigid body rotations, but the translation is not restrained. We eliminate the rigid body translations by fixing the center in the second PVE (and correspondingly vertex in the first PVE), in all three directions.

The deformed shape, along with contours of von Mises stresses are shown in Figure 5-8, which visually confirm that the results are the same between the two PVEs.

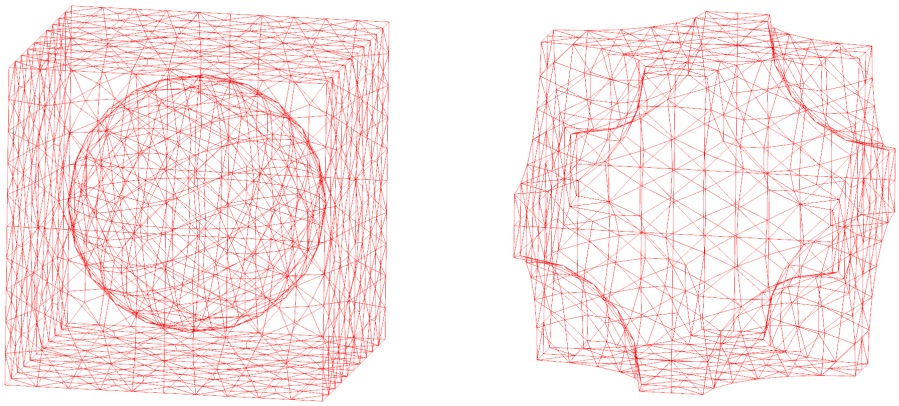


Figure 5-7. – Meshes for two different periodic volume elements.

For quantitative comparison, the stresses are examined at three different locations: at -(0.1,0.5,0.5), -(0.5,0.1,0.5), -(0.5,0.5,0.1), divided by the stress at center of the spherical inclusion. To be precise, the results are obtained at element centroids closest to the three locations, which are shown in Table 5-2. Only one set of results is shown since the computed stresses are identical between the two models, clearly verifying the efficacy of the PVE modeling in Salinas.

Table 5-2. – Stresses computed from PVE model.

Location	-(0.1,0.5,0.5)	-(0.5,0.1,0.5)	-(0.5,0.5,0.1)
σ_{xx}	114.9294	92.5208	112.5307
σ_{yy}	75.9910	77.9703	7.0164
σ_{zz}	37.4338	0.2489	41.6507
σ_{xy}	-48.9834	-79.2754	-101.6064
σ_{yz}	2.5491	20.0710	16.3110
σ_{xz}	-7.1000	-75.8791	-43.3911

In addition to the above example, we tested the implementation on homogeneous block under specified deformation gradient, resulting in expected uniform stress state with correct values (including Poisson's effect). The details are not presented in this document, but can be found in the test repository. For input see Appendix 9.56

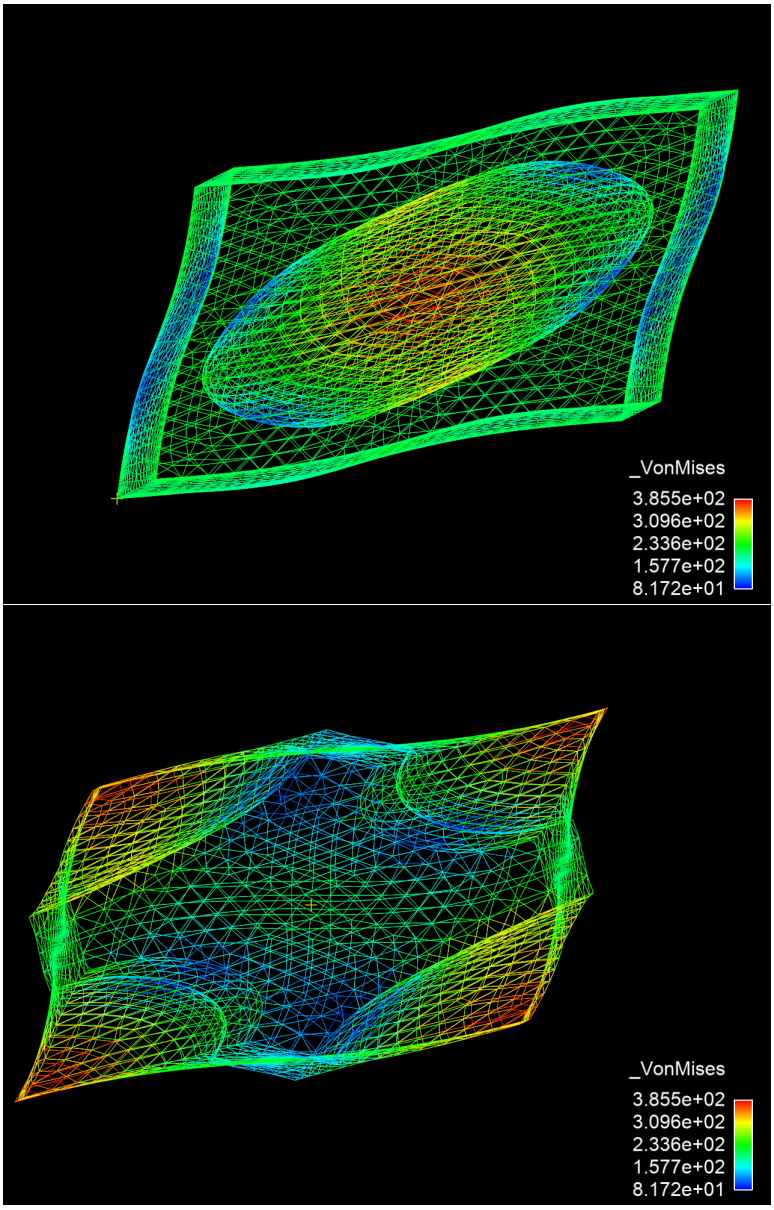


Figure 5-8. – Meshes for two different periodic volume elements.

5.5. Filter Rigid Modes from Loads

5.5.1. Introduction and Purpose

For some analyses, it is advantageous to remove the rigid body component of the deformation. This is the case for a reentry body for example, which may have a static preload followed by a transient response with applied random pressures. Unless the force is exactly self-equilibrated, the static preload leads to a singular system. The transient response is also troublesome. The true physics is complicated and includes a fluid-structure interaction with random pressures as well as flight dynamics which stabilize the structure from rotation. A random pressure load is sometimes a valid physical model. Unfortunately, that load can cause the body to rotate wildly, which is both nonphysical and distracting. As a solution, we filter the input forces to the body so that only self-equilibrated forces are applied. Because of the singularity, and small contributions to various linear solvers, a rigid body displacement may be generated. This component is filtered out after solving, leaving a displacement that has no rigid body component.

5.5.2. Description of the Test

In this test, a small beam of Hex8 elements has a load applied transverse to one end. See Figure 5-9. Because there are no boundary conditions, the resulting system is singular for a statics solution. Figure 5-10 indicates the equilibrated forces applied to the structure, and the resulting deformation.

Verification requires determining the following:

1. The loads are properly equilibrated.
2. The output displacement vector contains no rigid body components.

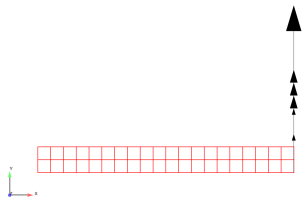


Figure 5-9. – Beam Loading.

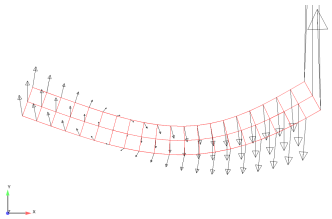


Figure 5-10. – Filtered Beam Forces and Displacements.

5.5.3. Evaluation

The verification is done by MATLAB. Forces and Displacements are loaded into the Matlab engine and simple calculations are performed.

1. The sum of each force component is zero (1.7e-6). This confirms that the translational portion of the force has been equilibrated.
2. The sum of cross terms is zero (1.25e-5).

$$\sum_{i}^{nodes} \vec{F}_i \times \vec{x}_i = 0$$

   This confirms that the net moments are zero. Thus, the loads have been properly equilibrated.

3. We confirm that the output displacement vector contains no rigid body components as follows.

   The net output translational components are summed for each component.

$$\sum_{i}^{nodes} \vec{u}_i = 0$$

   These components are less than 1e-10.

   We also confirm that the net moment is zero.

$$\sum_{i}^{nodes} \vec{u}_i \times \vec{x}_i = 0$$

   The net moment is less than 1.1e-5.

Thus, we have confirmed that the loads are self equilibrated, and that the resulting displacements are orthogonal to rigid body translation and rotation.

For input deck see Appendix 9.57.

5.6. Modal Force Loading

Modal Forces provide an alternative, body-based loading to a structure which can be useful for some solutions. These modal forces are the conjugate of modal force output in the `modaltransient` solution method.

Verification is performed by use of the modal transient method, and is shown in Figure 5-11. The model used is shown in Figure 5-12. The model is first run using physical inputs, and produces two output files: 1) the modal forces, and 2) the output displacements. The second run uses modal force as the input. Finally, the output displacements of the two modal transient runs are compared. Results are identical (except for round-off errors).

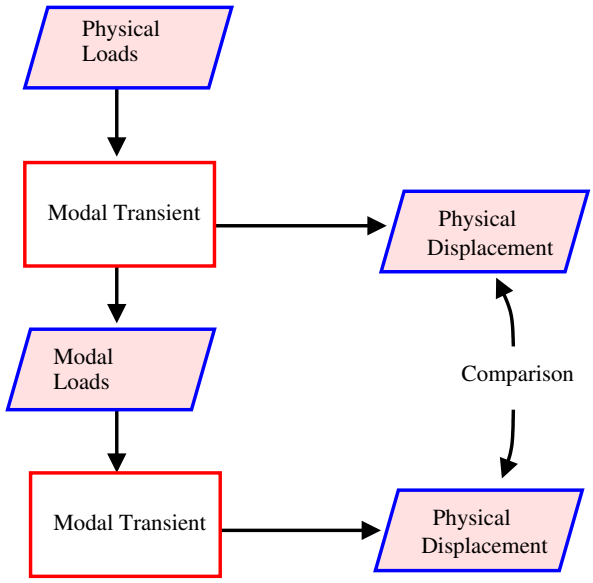


Figure 5-11. – Verification Process for Modal Force.

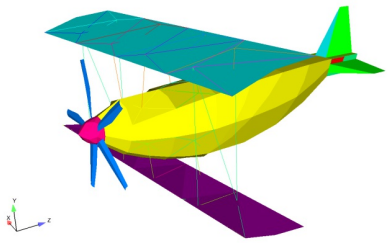


Figure 5-12. – Biplane Model.

For input see Appendix 9.58

5.7. Lighthill Analogy - Helmholtz Resonator

The Lighthill tensor provides a source term for noise generation in aeroacoustic simulations. The Lighthill tensor captures noise generated by unsteady convective fluid flow. Sierra/SD produces a source term from the Lighthill tensor that is applied as a nodeset load in the pressure formulation of acoustics. Sierra/SD produces the Lighthill loading by reading in the time varying divergence of the Lighthill tensor using the `readnodalset` function. The divergence of the Lighthill tensor is used to create an equivalent elemental force vector. It is provided by the Fuego incompressible flow code.

Verification of the Lighthill loading is performed for the Helmholtz resonator shown in Figure 5-13 which has an analytic resonant frequency of 120Hz. The discretized mesh, material properties, initial and boundary conditions used in the Fuego simulation are shown in Figure 5-14. Fuego then calculates the divergence of the Lighthill tensor and writes this out to exodus as nodal data at variable time steps.

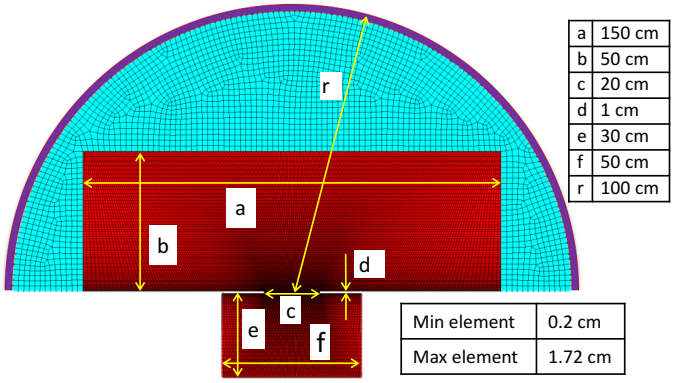


Figure 5-13. – Dimensions of Helmholtz resonator.

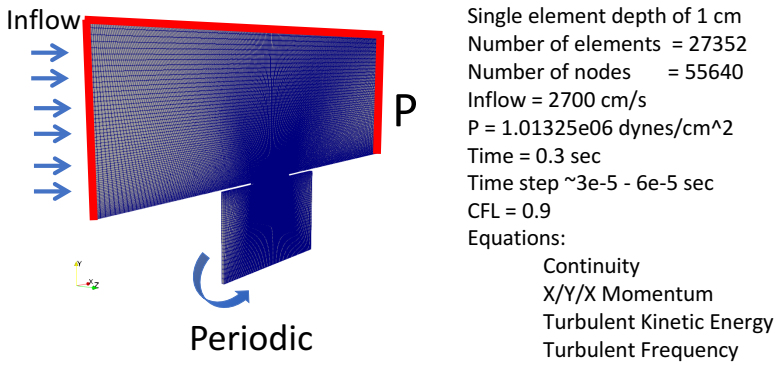


Figure 5-14. – Boundary and initial conditions for Fuego simulation.

The Fuego output is used as input in Sierra/SD with the same discretization of the Helmholtz resonator shown in Figure 5-13 with an additional semi-circular domain in order to apply an acoustic boundary condition. Absorbing boundary conditions are applied to the edge sideset of the semi-circular region, highlighted in red in Figure 5-14. boundary with absorbing boundary

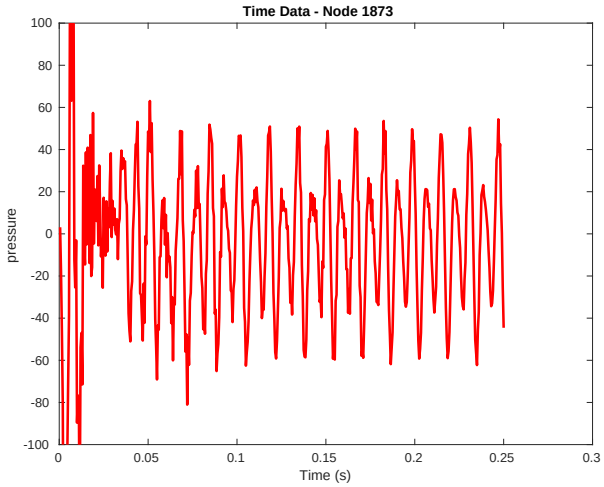


Figure 5-15. – **Sierra/SD** time history of pressure for Lighthill loading.

conditions eliminate the rigid body modes from the solution which can cause a which linear growth in the pressure field. Fuego's nodal DivT field is converted to nodeset data using the ejoin flag -convert_nodal_to_nodesets. Distribution factors for the new nodeset data are increased from 0 to 1. The **Sierra/SD** simulation reads in the time varying nodeset data from Fuego and interpolates it to the nearest time step either linearly or using the closest time step. The double divergence of the Lighthill tensor is then calculated and applied as a source term in the **Sierra/SD** transient acoustic simulation. Results for the **Sierra/SD** acoustic simulation using Lighthill loading are shown in Figure 5-15 for acoustic pressure versus time. An FFT of the pressure data is shown in Figure 5-16 with peaks at 61, 121, and 183. These resonances were also observed in the pressure data sampled in the rigid chamber of the Fuego simulation. The main peak is close to the analytic resonant frequency of 120Hz.

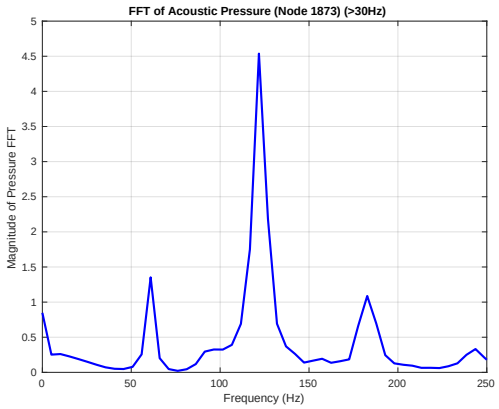


Figure 5-16. – FFT of **Sierra/SD** pressure data shown in Figure 5-15.

For input see Appendix 9.59

5.8. Lighthill Tensor Verification

The Lighthill tensor provides a source term for noise generation in aeroacoustic simulations. The Lighthill tensor, $\mathbf{T}$, captures noise generated by unsteady convection in flow in a fluids simulation. Sierra/SD produces a source term from the Lighthill tensor that is applied as a nodeset load in the pressure formulation of acoustics. Sierra/SD produces the Lighthill loading by reading in the time varying divergence of the Lighthill tensor using the `readnodalset` function. The divergence of the Lighthill tensor, $\nabla \cdot \mathbf{T}$, is used to create an equivalent elemental force vector.

In this verification example we compare Lighthill loading to Point Volume Acceleration loading for a 1-D waveguide shown in Figure 5-17a). The `Lighthill` and `pointVolAcc` load functions are applied to the nodeset indicated by the yellow region. For this example the divergence of the Lighthill Tensor varies only in the x-direction and is given by

$$(\nabla \cdot \mathbf{T})_x = \left(1 + \cos\left(\frac{\pi x}{20}\right)\right) \sin^2\left(\frac{\pi t}{40}\right) \text{ for } t \leq 40\text{s} \tag{5.8.1}$$

where x is the location along the x-axis and t is time. Only a single load pulse is simulated, $t \leq 40$s. The simulation is run for a total time of 550s, giving the pressure pulse time to propagate away from the nodeset. The y and z components of $\nabla \cdot \mathbf{T}$ are zero. This form for Lighthill loading makes $(\nabla \cdot \mathbf{T})_x = 0$ at the end of the nodeset, $x = \pm 20$.

The same pressure response as that given in equation 5.8.1 is produced with a scalar nodal load equal to $\nabla \cdot (\nabla \cdot \mathbf{T})$ properly scaled by the number of nodes and area it is acting over. For the $\nabla \cdot \mathbf{T}$ used in this example,

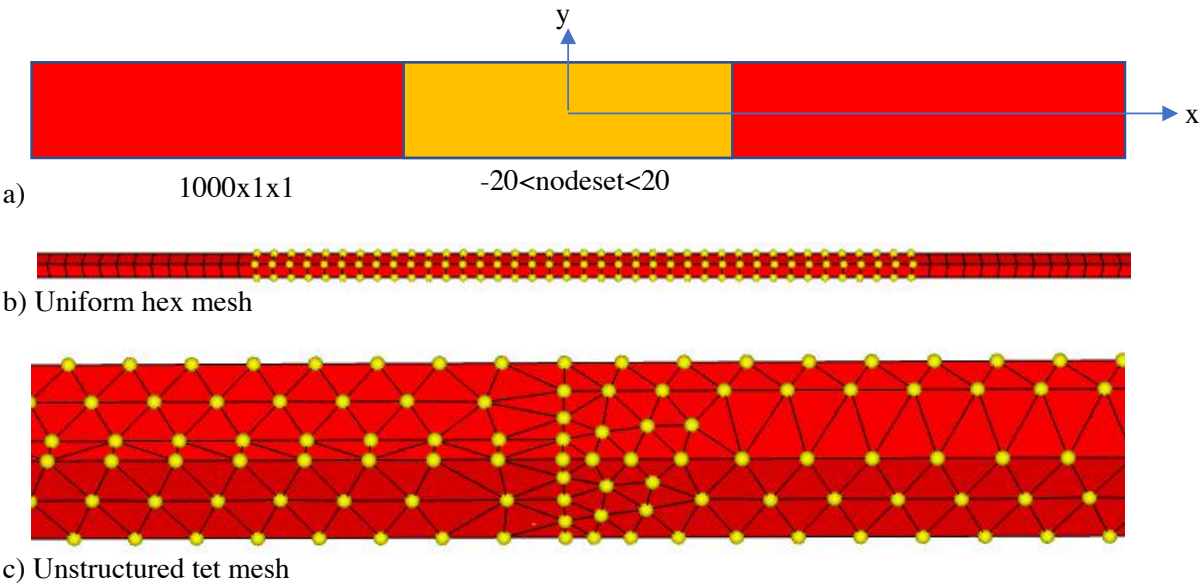
$$\nabla \cdot (\nabla \cdot \mathbf{T}) = -\frac{\pi}{20}\left(\sin\left(\frac{\pi x}{20}\right)\right) \sin^2\left(\frac{\pi t}{40}\right) \text{ for } t \leq 40\text{s} \tag{5.8.2}$$

and the scalar nodal force applied using Point Volume Acceleration is $\frac{1}{4}\nabla \cdot (\nabla \cdot \mathbf{T})$ for the uniform hexahedron mesh shown in Figure 5-17b where each element is 1x1x1.

Figure 5-18 shows the pressure output at t=75s over the length of the waveguide for `Lighthill` and `pointVolAcc` loading given by equations 5.8.1 and 5.8.2, respectively, applied to the uniform mesh shown in Figure 5-17b. These are compared to the analytical result shown by the black line. The results are given after the pressure pulse has been applied, showing the propagation of the pressure wave through the acoustic medium. The percent difference in pressure between the two loading methods and the analytical result is shown in Figure 5-19 at t=75s. The L1 error of the pressure over the domain is shown at each simulation time step in Figure 5-20. This plot shows the L1 error increasing over the duration of the `Lighthill` or `pointVolAcc` load ($t < 40s$) and then remaining steady.

The geometry in Figure 5-17a) was also discretized with an unstructured linear tetrahedron mesh shown in Figure 5-17c) and `Lighthill` loading was applied to the domain. Results for these simulations are also shown in Figures 5-18-5-20 and show the same error as the uniform hexahedron mesh with Lighthill loading.

For input see Appendix 9.60



a)

b) Uniform hex mesh

c) Unstructured tet mesh

Figure 5-17. – a) Schematic of 1000x1x1 waveguide geometry. Geometry extends from x=±500. Yellow region contains the nodeset being loaded. b) Regular hex mesh used to compare Lighthill and Point Volume Acceleration loading. c) Unstructured tet mesh used for Lighthill loading. Yellow nodes in b) and c) indicate nodes in nodeset being loaded.

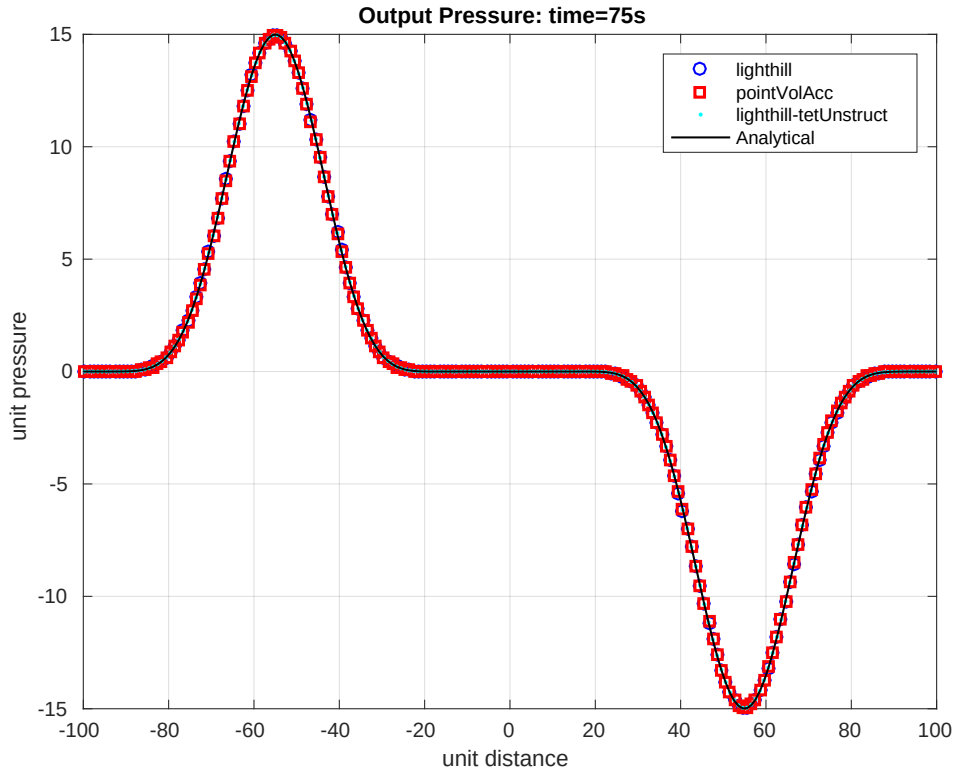


Figure 5-18. – Pressure output for 3 load cases compared to analytical result at t=75s.

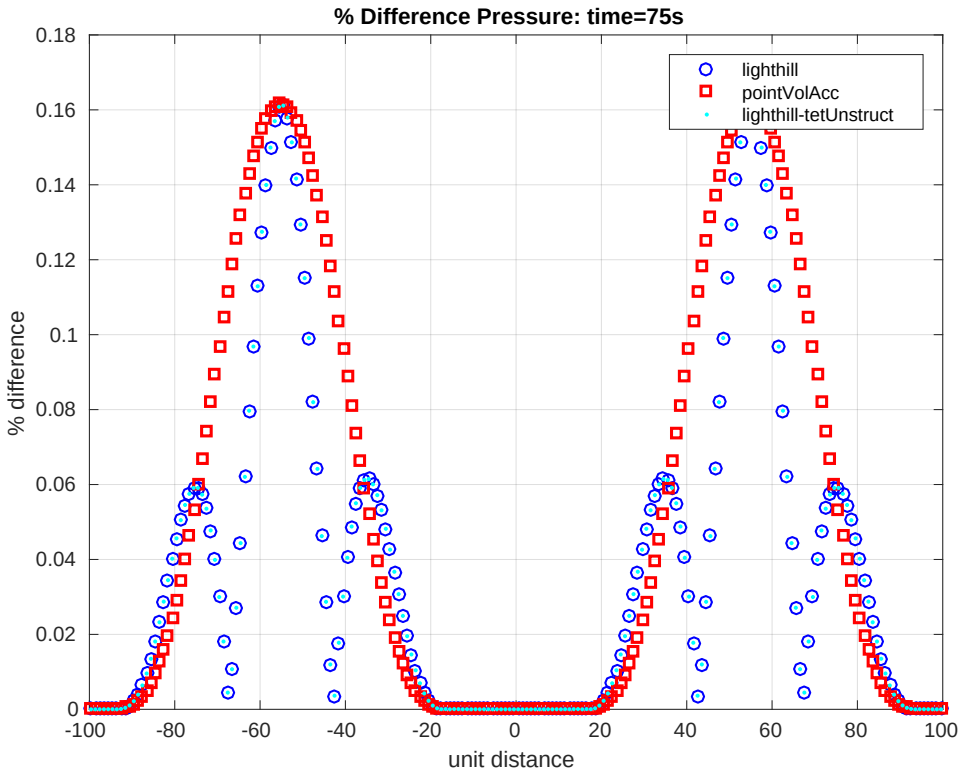


Figure 5-19. – Percent difference in pressure between the three load cases and the analytical pressure for t=75s.

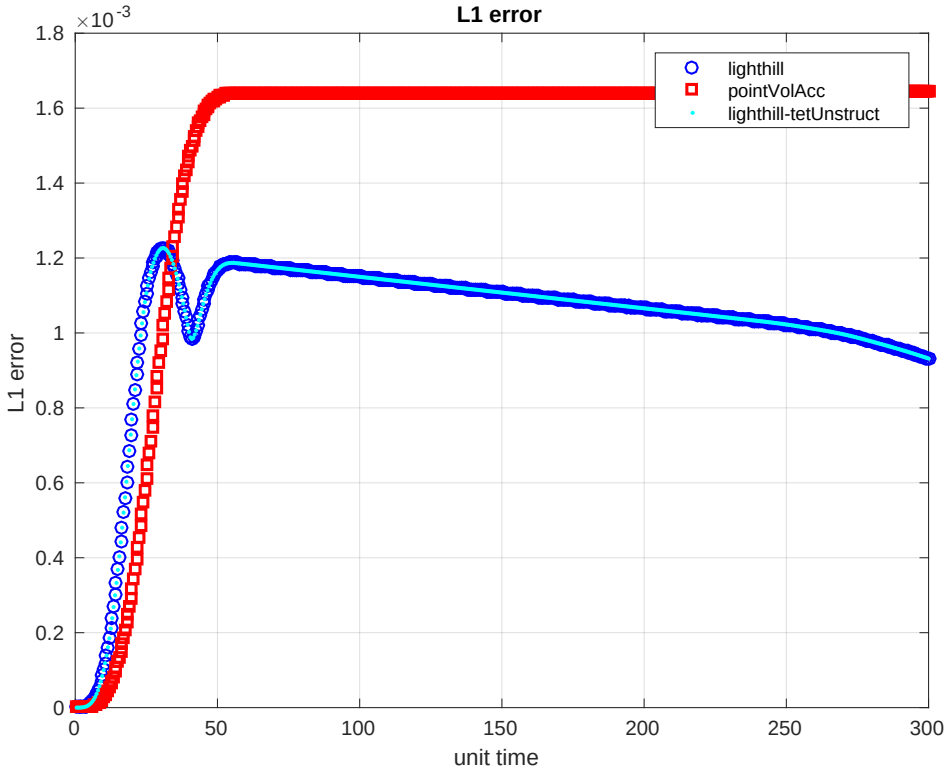


Figure 5-20. – L1 error in pressure for each load type versus time.

5.9. Acoustic Point Source in Frequency Domain

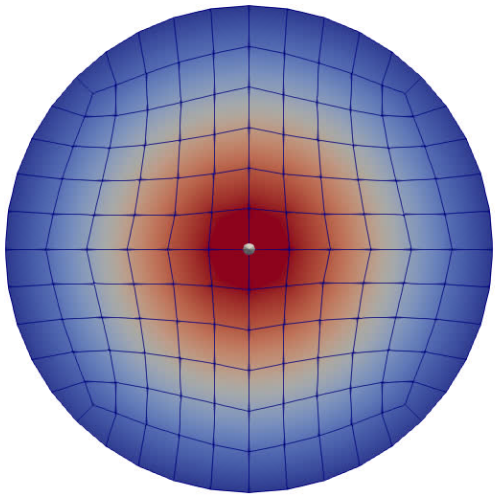


Figure 5-21. – Acoustic Point Source – Coarse Example.

Consider an acoustic point source in a sphere of acoustic elements, shown in Figure 5-21. An absorbing boundary condition is applied at radius 2, representing an unbounded acoustic domain. The source is understood as a pulsating sphere with volume V and time derivative $Q = dV/dt$. The value for Q is specified using the keyword **point_volume_vel**.

In the frequency domain, with ω the circular frequency of the wave and $k = \omega/c$ the wave number, the pressure at a distance $r = |x|$ from the source is given by

$$p = i\omega\rho Q \frac{e^{-ikr}}{4\pi r}; \tag{5.9.1}$$

see the section "Point Acoustic Sources" in the theory manual for a detailed explanation.

Figure 5-22 shows a two-dimensional slice of the result for a frequency of 91 Hz. At a point on the outside of the sphere, with radius $r = 2$ from the point source, the exact and computed solutions are compared. For the SierraSD solutions, a damping term of $\beta = 1.0e - 5$ was added to facilitate solver convergence.

On the boundary of the mesh, with $r = 2$, and a frequency of 91 Hz, the exact answer is Apressure = -5.623 and ImagAPressure = -28.873. For the mesh shown in Figure 5-21, which is relatively coarse, SierraSD calculates Apressure = -4.826 and ImagApressure = -28.600. For the refined mesh shown in Figure 5-22, SierraSD calculates Apressure = -5.513 and ImagAPressure = -28.580.

The verification test suite verifies both nodal point source and element point source options. For input see Appendix 9.61

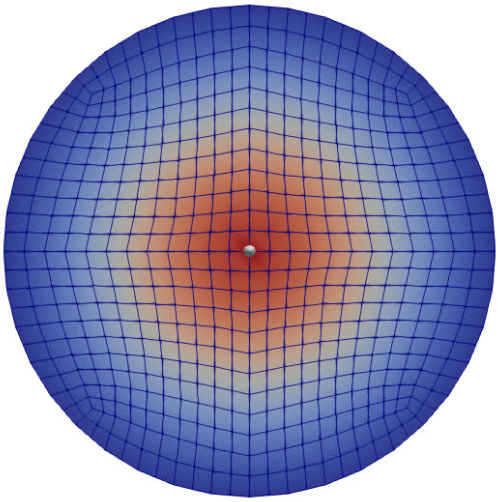


Figure 5-22. – Acoustic Point Source – Refined Example.

5.10. Acoustic Point Source in Time Domain

To verify the transient acoustic point source (Point_Volume_Vel) in Sierra/SD, we consider a spherical domain with a point source at the center. The spherical domain is given absorbing boundary conditions around its boundary, to make the space look infinite in all directions. With this arrangement, we have the problem of a point source in an infinite domain.

The analytical solution to this problem is given by Pierce [39], as follows

$$p(R,t) = \frac{\rho}{4\pi R}\dot{Q}(t - \frac{R}{c})H(t - \frac{R}{c}) \tag{5.10.1}$$

where $p(R,t)$ is the pressure at a distance R from the source and at time t, ρ is the fluid density, c is the speed of sound, $H(t)$ is the Heaviside function, and $Q(t)$ is the time derivative of volume change of the source,

$$Q = \frac{dV}{dt} \tag{5.10.2}$$

In this problem, we chose $Q(t) = \sin(50\pi t)$, and we examined the solution at the exterior boundary of $R = 2$. Inserting this into Equation 5.10.1 gives

$$p(R,t) = \frac{50\rho}{8}\cos(50\pi(t - \frac{2}{343}))H(t - \frac{2}{343}) \tag{5.10.3}$$

Figure 5-23 shows a comparison of the Sierra/SD results for this problem compared against Equation 5.10.3. Excellent agreement is obtained, except for the initial time, where the numerical solution shows some difficulty resolving the abrupt change in the exact solution, which comes from the Heaviside function in Equation 5.10.3. We can also verify the "Point_Volume_Accel" point source with an input of $\dot{Q}(t) = 50\pi\cos(50\pi t)$, and get the same solution.

Two variants of the problem are included in the verification test suite. The first variant uses a node-based point source at a single node at the center of the sphere. The second variant uses an element-based point source at a single element at the center of the sphere. Both variations produce nearly identical results on a relatively coarse mesh and converge to the same analytic solution with refinement. For input see Appendix 9.62

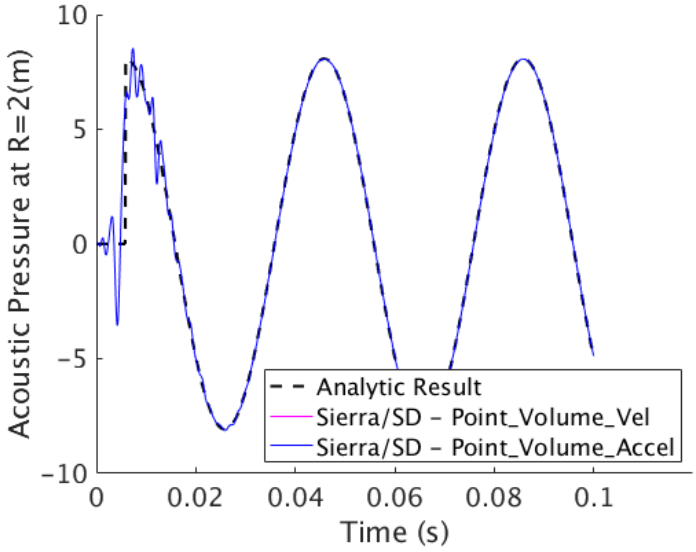


Figure 5-23. – Transient Verification of a PointSource in an Infinite Medium.
Comparison of computed and exact solution for a point source in an infinite medium.

5.11. Acoustic Plane Wave Scattering in Frequency Domain

Consider an acoustic plane wave traveling in the $+x$-direction. In the frequency domain, with $\omega = 2\pi f$ describing the angular frequency and $k = \omega/c_0$ describing the wavenumber, the pressure at every point in space is

$$p = p_0 e^{ikx} \,. \tag{5.11.1}$$

Now, consider scattering of a plane wave incident on a steel cylinder in air. Due to the cylindrical symmetry of the problem, it is useful to expand the spatial dependence of the incident plane wave via an infinite series as

$$e^{ikx} = e^{ikr\cos\theta} = \sum_{n=0}^{\infty} i^n \epsilon_n J_n(kr)\cos(n\theta) \,. \tag{5.11.2}$$

where r is the distance from the origin, θ is the azimuthal angle, ϵ_n is the Neumann factor (equal to 1 for $n = 0$ and 2 otherwise), and $J_n(kr)$ are Bessel functions of order n. The scattered pressure field can then be written as

$$p_{\text{sc}} = p_0 \sum_{n=0}^{\infty} i^n \epsilon_n A_n H_n^{(1)}(kr)\cos(n\theta) \,, \tag{5.11.3}$$

where A_n are scattered field coefficients and $H_n^{(1)}(kr)$ are Hankel functions of order n. Hereafter, the superscript will be dropped for notational convenience. Similar expansions can be written for the displacement fields (both longitudinal and transverse) in the cylinder itself, but those are omitted here.

Continuity of radial displacement, continuity of radial stress, and continuity of tangential stress must be enforced at the surface of the cylinder to find A_n. Since fluids cannot support shear stress, the tangential stress must therefore be zero at the boundary. These boundary conditions are straightforward to enforce, but they result in complicated expressions for the scattered field coefficients. While the general expressions can be viewed in `elastic_cylinder_fluid_medium.m`, the scattered field coefficients for a rigid and immovable cylinder are sufficient in this case because steel is acoustically rigid compared with air. These coefficients are

$$A_n = -\frac{J_n'(kR)}{H_n'(kR)} \,. \tag{5.11.4}$$

For a plane wave of frequency $f = 1$ kHz incident on a cylinder of radius $R = 0.1$, the scattered pressure field is shown in Figure 5-24. A PML boundary condition is applied at radius 0.8, representing an unbounded acoustic domain. The pressure amplitude p_0 is specified to be unity. At the point $(x, y) = (0.2, 0)$, the exact solution answer is Apressure = 0.7072 and ImagAPressure = 0.1875, and For a mesh size of 0.01, SierraSD calculates Apressure = 0.7037 and ImagAPressure = 0.1896. For input see Appendix 9.63

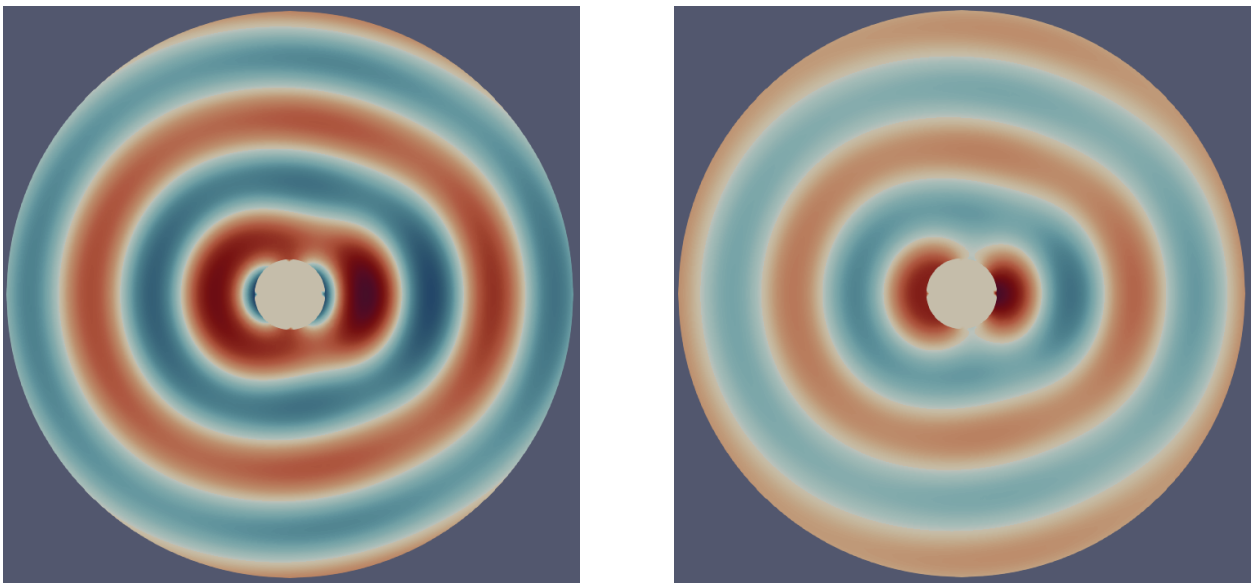


Figure 5-24. – Acoustic Plane Wave Scattering from a Cylinder.

5.12. Transient Reaction Forces

The response of a simple transient system is demonstrated. The transient time integrator, output of kinematic quantities from transient solution, and the output of reaction force quantities from transient solution are verified. The Mathematica notebook contains the derivation of expected quantities.

5.12.1. Finite Element Model

The model consists of four Spring-Dashpot elements connected to a central concentrated mass as shown in Figure 5-25.

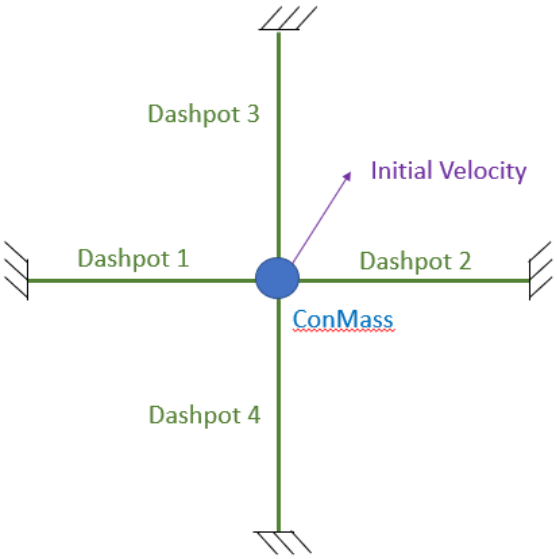


Figure 5-25. – Reaction Force Model.

Dashpots 1 and 2 act only in the X direction. Dashpots 3 and 4 act only in the Y direction. Each Dashpot has a unique stiffness and damping coefficient given by Table 5-3. The central conmass has a mass of 2.5.

Block	Stiffness	Damping
1	1.1	0.7
2	1.2	0.8
3	1.3	0.9
4	1.4	1.0

Table 5-3. – Dashpot Element Properties.

The model can be treated as two independent single degree of freedom systems. One system involving the sum of the stiffness and damping of Dashpots 1 and 2 acting in the X direction and a

second system involving the sum of stiffness and damping of Dashpots 3 and 4 acting in the Y direction. The Z degree of freedom of all nodes is fixed as is the far end of each Dashpots.

5.12.2. Damped Vibration Due to Initial Conditions

The input deck 'initCond.inp' applies an initial velocity (10, 20, 0) to the conmass and then solves for the resultant system response. At standard textbook response is used for vibration of a single degree of freedom damped system given in the equations 5.12.1 to 5.12.4. K is the stiffness given by the sum of the two Dashpot stiffness. M is the mass of the concentrated mass. C is the sum of the two Dashpot damping coefficients. Initial conditions of displacement are given by d_0 and velocity by v_0

$$\omega = \sqrt{\frac{K}{M}} \tag{5.12.1}$$

$$\zeta = \frac{C}{2 * M * \omega} \tag{5.12.2}$$

$$\omega_{damped} = \omega\sqrt{1 - \zeta^2} \tag{5.12.3}$$

$$d(t) = e^{-\zeta\omega * t}(d_0 cos(\omega_{damped} t) + sin(\omega_{damped} t)\frac{v_0 + \zeta\omega d_0}{\omega_{damped}}) \tag{5.12.4}$$

In Sierra/SD the dynamics are integrated through time using the Newmark-Beta time integrator. A small time step is used so that the results have a high degree of time accuracy. The tests checks equivalence between the analytic and Sierra/SD results kinematic quantities at specific time steps in the solution.

The expected reaction forces can be found by considering the fundamental system equation given in 5.12.5.

$$Ku + C\dot{u} + M\ddot{u} = f \tag{5.12.5}$$

5.12.3. Prescribed Acceleration

A second tested case involves constant prescribed acceleration on the central node. The velocity and displacement of the central node can be found via integration of the acceleration. Based on the kinematic motion the forces are given 5.12.5. For the prescribed acceleration case the total damping matrix is formed from the C of the Dashpot and mass proportional (0.1) and stiffness proportional (0.2) damping coefficients.

For input see Appendix 9.64.

6. OUTPUT

6.1. Relative Displacement PSD

A typical requirement often met using random vibration analysis is to determine the probability of interference of two nodes. This use case is discussed in subsubsection RMS Output, subsection Random Vibration, Section Solution Procedures of the Theory Manual, and details about usage can be found in the *User's Manual*, in Outputs under Relative_Disp.

In the following examples, we consider a 1D problem. Specifically, we investigate the relative displacement output of a joint2G element in response to two conmass nodes.

6.1.1. In Phase Response: Nodal closest distance user output

Figure 6-1. – Diagram of nodes moving in phase.

In this example the motion of both, the face and the node, is precisely in phase (see figure 6-1). In that case, the difference in motion between the face and the node should report no response, as seen in figure 6-2. This test verifies that the gap differencing element does not report the rigid motion of the element.

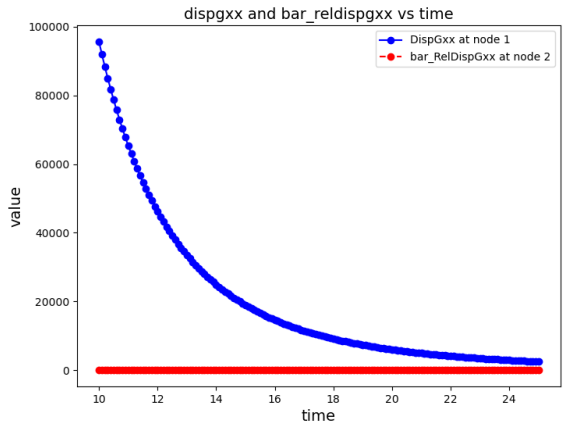


Figure 6-2. – Closest Distance: Nodes moving in phase.

6.1.2. *Element relative_disp output*

In this example, the motion of both nodes is precisely in phase (see figure 6-3). In that case, the difference of the two nodes should report no response, as seen in figure 6-4. This test verifies that the gap differencing element does not report the rigid motion of the element.

Figure 6-3. – Diagram of nodes moving in phase.

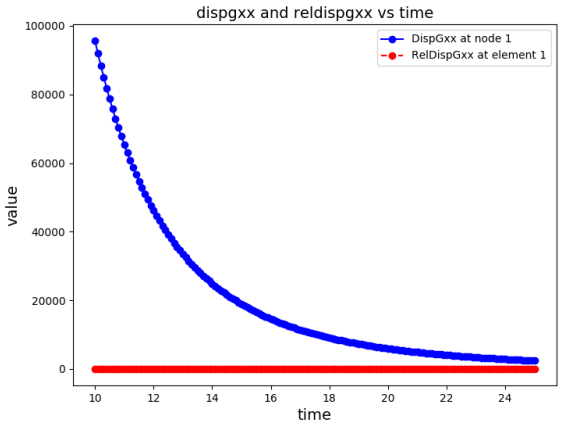


Figure 6-4. – Relative Distance: Nodes moving in phase.

6.1.3. *Opposite Phase Response: Nodal closest distance user output*

In this example, the motions of the face and that of the node are precisely out of phase with each other (see figure 6-5). This is a direct test of the gap differencing element's measurement of the difference in motion between two points. In the time domain, this condition would result in the gap element reporting twice the response at each node. Likewise, in the frequency domain the gap element should report 4 times the response. This relation is seen in figure 6-6.

Figure 6-5. – Nodes moving exactly out of phase: diagram.

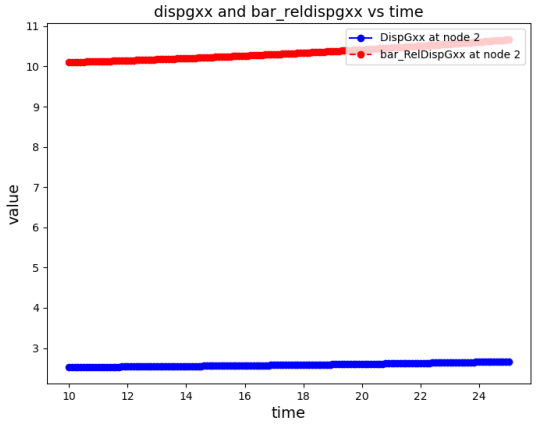


Figure 6-6. – Nodes moving exactly out of phase: results.

6.1.4. Element relative_disp output

Figure 6-7. – Nodes moving exactly out of phase: diagram.

In this example, the motion of the two nodes are precisely out of phase with each other (see figure 6-7). This is a direct test of the gap differencing element's measurement of the difference in motion between two points. In the time domain, this condition would result in the gap element reporting twice the response at each node. Likewise, in the frequency domain the gap element should report 4 times the response. This relation is seen in figure 6-8.

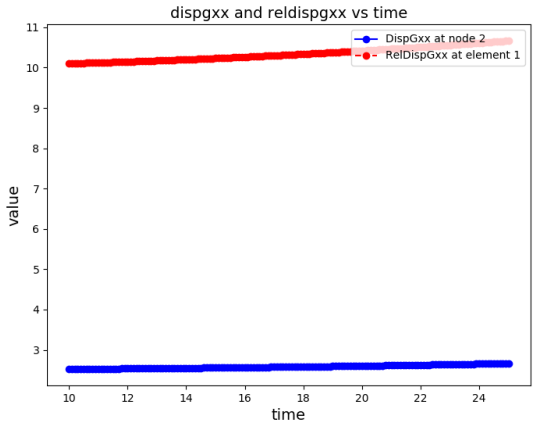


Figure 6-8. – Nodes moving exactly out of phase: results.

6.1.5. One Node Fixed Response: Nodal closest distance user output

In this example, the face is fixed, and the node is free as shown in figure 6-9. The results figure 6-10 have the expected behavior: the difference in motion between the node and the face coincides with the motion of the free node.

Figure 6-9. – Face fixed and node free diagram.

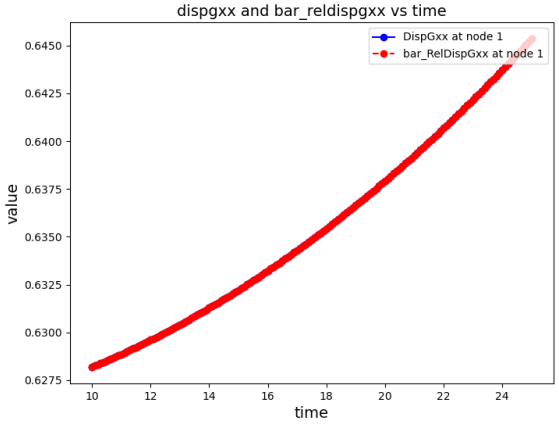


Figure 6-10. – Face fixed and node free results.

6.1.6. Element relative_disp output

Figure 6-11. – Left node fixed and right free diagram.

In this example, one node is fixed, and the other is free as shown in figure 6-11. The results figure 6-12 have the expected behavior: the difference in motion between two nodes is equal to the motion of the free node,

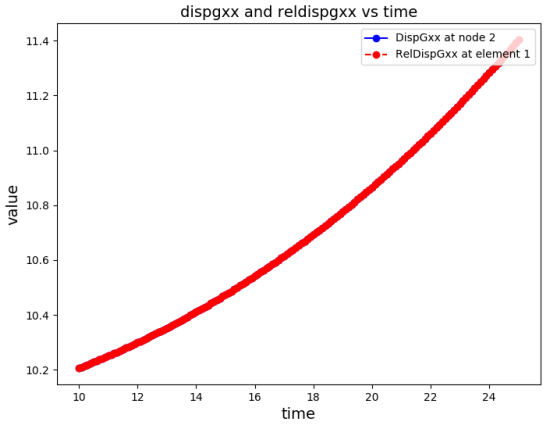


Figure 6-12. – Left node fixed and right free results.

6.1.7. *Tuning fork response*

In this test, we verify the gap calculation from **Sierra/SD** by directly computing the expected gap PSD from modal displacements. This test also involves several overlapping tied joints, as shown in figure 6-13.

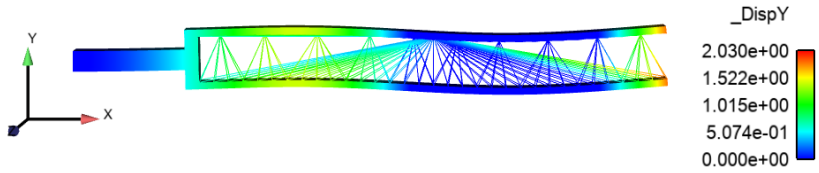


Figure 6-13. – Tuning fork with multiple overlapping tied joints.

The PSD, $G(\omega) \in \mathbb{R}^{34\times34}$, is given by

$$G = H\, S_{ff}\, H^{*} \tag{6.1.1}$$

where $S_{ff}(\omega) \in \mathbb{R}^{34\times34}$ is the forcing PSD, and the transfer function $H(\omega) \in \mathbb{R}^{34\times34}$ is given by

$$H = \Phi\, \widetilde{H}\, \Phi^{T} \tag{6.1.2}$$

where $\Phi \in \mathbb{R}^{34\times12}$ is the matrix of mode shapes (dofs x mode shapes), and the modal transfer function $\widetilde{H}(\omega) \in \mathbb{R}^{12\times12}$ is a diagonal matrix given at each mode n as

$$\widetilde{H}_{nn} = \frac{1}{\omega_n^2 + 2i * \omega\gamma\omega_n - \omega^2} \tag{6.1.3}$$

For input see Appendix 9.65.

6.2. Projection of Gauss Point Stresses to Nodes

Sierra/SD has the capability of projecting Gauss point stresses to nodes using a least squares minimization approach. Details of this projection approach can be found in the L_2 Projection of Gauss Point Stresses chapter of the Theory Manual. Currently, this capability is restricted to HEX20, TET10, WEDGE15 and HEX8 elements.

The first set of verification tests deals with meshes of HEX20, TET10 and WEDGE15 finite elements (see Figure 6-14) subjected to a uniform gravitational field (g_x, g_y, g_z) in the three global axes. The domain for the rectangular mesh is $x \in [-L_x/2, L_x/2]$, $y \in [-L_y/2, L_y/2]$ and $z \in [-L_z/2, L_z/2]$. Normal displacements are constrained to zero along three faces of the mesh associated with minimum coordinate values. That is, $u(-L_x/2, y, z) = 0$, $v(0, -L_y/2, 0) = 0$ and $w(0, 0, L_z/2) = 0$. In order to have a simple closed-form solution, Poisson's ratio of the isotropic material is set to zero. Further, the elastic modulus and mass density of the material are denoted by E and ρ.

The displacements vary quadratically with the spatial coordinates according to

$$u(x, y, z) = \frac{\rho g_x L_x^2}{2E}[-(x/L_x)^2 + (x/L_x) + 3/4],$$
$$v(x, y, z) = \frac{\rho g_y L_y^2}{2E}[-(y/L_y)^2 + (y/L_y) + 3/4],$$
$$w(x, y, z) = \frac{\rho g_z L_z^2}{2E}[-(z/L_z)^2 + (z/L_z) + 3/4],$$

while the normal stresses varying linearly and are given by

$$\sigma_{xx}(x, y, z) = \frac{\rho g_x L_x}{2}(1 - 2x/L_x),$$
$$\sigma_{yy}(x, y, z) = \frac{\rho g_y L_y}{2}(1 - 2y/L_y),$$
$$\sigma_{zz}(x, y, z) = \frac{\rho g_z L_z}{2}(1 - 2z/L_z).$$

The shear stresses σ_{xy}, σ_{xz} and σ_{yz} are all zero.

Because quadratic displacements can be approximated exactly by the shape functions of HEX20, TET10 and WEDGE15 elements, the finite element solution should match the closed-form solution for displacements at the nodes and for stresses at the Gauss points. In addition, since the projection approach uses linear shape functions, the projected stresses should also match at the nodes, which they indeed do.

The next test is identical to the first, but here only HEX8 elements are considered. The shape functions for HEX8 elements cannot approximate quadratic displacements exactly, so we don't expect Gauss point stresses projected to the nodes to match linearly varying stress fields. Nevertheless, we do expect nodal stresses to converge to the known stresses with mesh refinement. Figure 6-15 shows how the maximum absolute value of stress errors is reduced with mesh refinement. Here, h is the mesh interval length used by the Cubit mesh generation code. We note

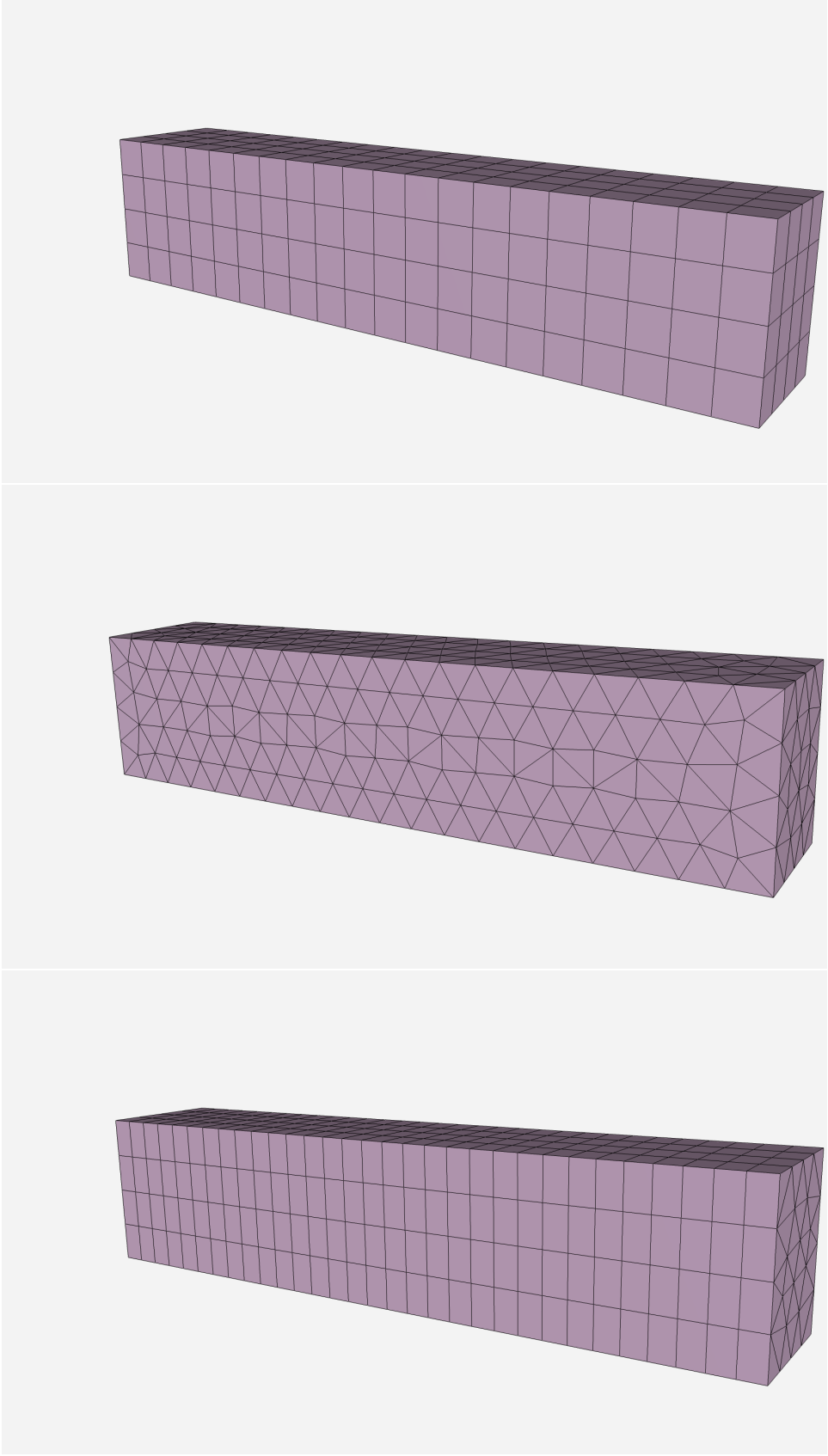


Figure 6-14. – Meshes of HEX20, TET10 and WEDGE15 elements use in first verification test.

that $h = 0.25$ in the top mesh of Figure 6-14 and results in 20 elements along its length. Also, the results in Figure 6-15 suggest a convergence rate proportional to the first power of h.

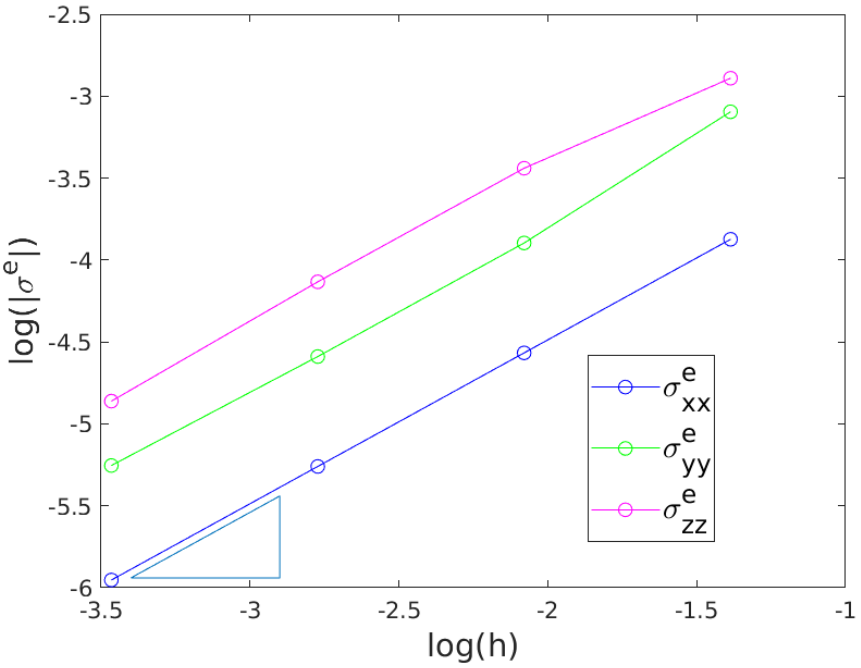


Figure 6-15. – Convergence of nodal stresses for HEX8 elements. The slope of the triangle near the bottom of the figure is 1.

The final test involves a unit cube geometry rotated by 30 degrees about an axis in the direction $(1, 2, 3)$. We again consider meshes of HEX20, TET10 and WEDGE15 elements (see Figure 6-16 for the TET10 mesh). As before, Gauss point stresses are projected to the nodes. These nodal stresses are then used to calculate principal and Von Mises stresses at the nodes.

Uniform gravity loads are again applied, but they are now aligned with the three axes of the cube. To help explain, let θ denote the angle of rotation about a unit vector axis (u_x, u_y, u_z). Also, let $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$ denote unit vectors fixed in the global coordinate system while $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$ are unit vectors fixed in the cube mesh. Entries of the direction cosine matrix are defined as

$$c_{ij} = \boldsymbol{n}_i \cdot \boldsymbol{b}_j$$

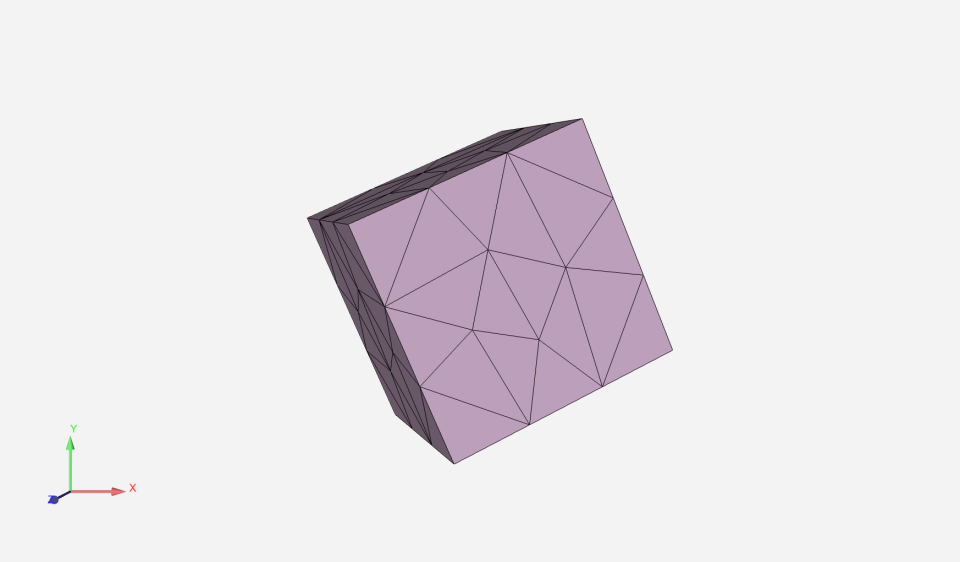


Figure 6-16. – Rotated mesh of TET10 elements.

and are given by

$$
\begin{aligned}
c_{11} &= \cos(\theta) + u_x^2(1-\cos(\theta)),\\
c_{22} &= \cos(\theta) + u_y^2(1-\cos(\theta)),\\
c_{33} &= \cos(\theta) + u_z^2(1-\cos(\theta)),\\
c_{12} &= u_x u_y(1-\cos(\theta)) - u_z\sin(\theta),\\
c_{23} &= u_y u_z(1-\cos(\theta)) - u_x\sin(\theta),\\
c_{31} &= u_z u_x(1-\cos(\theta)) - u_y\sin(\theta),\\
c_{13} &= u_x u_z(1-\cos(\theta)) + u_y\sin(\theta),\\
c_{21} &= u_y u_x(1-\cos(\theta)) + u_z\sin(\theta),\\
c_{32} &= u_z u_y(1-\cos(\theta)) + u_x\sin(\theta).
\end{aligned}
$$

Given a position vector $\boldsymbol{r} = x\boldsymbol{n}_1 + y\boldsymbol{n}_2 + z\boldsymbol{n}_3$, we have

$$\tilde{x}\boldsymbol{b}_1 + \tilde{y}\boldsymbol{b}_2 + \tilde{z}\boldsymbol{b}_3 = x\boldsymbol{n}_1 + y\boldsymbol{n}_2 + z\boldsymbol{n}_3.$$

Taking dot products of each side of this expression with $\boldsymbol{b}_j$ for $j = 1, 2, 3$, we obtain

$$
\begin{aligned}
\tilde{x} &= c_{11}x + c_{21}y + c_{31}z,\\
\tilde{y} &= c_{12}x + c_{22}y + c_{32}z,\\
\tilde{z} &= c_{13}x + c_{23}y + c_{33}z,
\end{aligned}
$$

which relate global coordinates to body-fixed coordinates.

The gravity load is given by $\boldsymbol{g} = \boldsymbol{b}_1 + 10\boldsymbol{b}_2 + 100\boldsymbol{b}_3$ and a unit density is used. This gives the exact

stresses (in body-fixed coordinates)

$$\sigma_{11} = 1/2 - \tilde{x},$$
$$\sigma_{22} = 10(1/2 - \tilde{y}),$$
$$\sigma_{33} = 100(1/2 - \tilde{z}).$$

All shear stresses, again in body-fixed coordinates, are zero. With these exact expressions in hand, the minimum, intermediate and maximum principal stresses are compared with those calculated using Sierra/SD. Similarly, the exact Von Mises stresses are also compared. As expected, these comparisons are found to be in agreement for cube meshes of HEX20, TET10 and WEDGE15 elements.

For input see Appendix 9.66.

This page intentionally left blank.

7. CONTACT, CONSTRAINTS AND MPCS

MultiPoint Constraints (MPCs) are applied in structural dynamics for some reasons. Typical uses include spreading a load over many input nodes, attaching dissimilar meshes, connecting lumped structures, applying boundary conditions and approximating rigid structures. The variety of uses for MPCs makes verification of their application difficult. Only small problems may typically be solved analytically.

Analytic problems for which some degrees of freedom may be eliminated using constraints will be compared with solutions from **Sierra/SD**. The problems for which these comparisons may be made are still to be determined at this time.

In addition to analytic problems, code comparisons for practical problems will be made. While code comparisons suffer from a number of problems, they have the advantages of comparing solutions to the type of problems expected in practice, and they provide some level of verification for components of the software which could otherwise not be tested.

7.1. Tied-Joint with Joint2G and Spring. Slip and Rigid

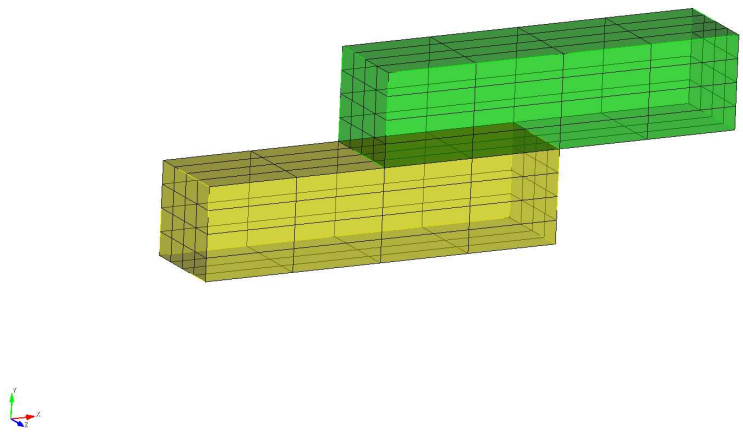


Figure 7-1. – Tied-Joint Model Geometry.

7.1.1. Purpose

The “Tied Joint” structure is a meta structure that provides an efficient and robust means of modeling a joint structure. The purpose of this document is to verify that both the tied-joint and conventional methods produce the same solution. Showing the results are the same encourages the use of tied-joints rather than the more tedious conventional method which involves replicating nodes and the use of multi-point constraints (MPCs). Generally, the input file for the tied-joint method is much simpler since all the constraints are accounted for, rather than having to list them by hand. Also, for the tied-joint input files the necessary constraints become included in the method itself, resulting in a simpler model for the input geometry file.

7.1.2. Lap Joint Comparison

7.1.2.1. Model Geometry

The lap joint model used for both the conventional and tied-joint tests consists of two partially overlapping rectangular blocks, as seen in Figure 7-1. The end of one of the blocks is fixed, while the opposite end of the other block is loaded with a constant applied force. The particular model seen here and used in the following results was created using Cubit and exported as an exodus file.

7.1.2.2. Building the Tied-Joint model

7.1.2.3. Non-slip

```
Tied Joint
        Normal Definition = none
                surface 1,2
        Shear Definition
                side = rigid
                connect to Block 33
end

Block 33
        Spring
                Kz = Elastic 1e9
                Kx = Elastic 1e9
                Ky = Elastic 1e9
end
```

Figure 7-2. – Tied-Joint Non-Slip Input.

The exodus file of the original model as described in 7.1.2.1 is the geometry file used for the tied-joint input. The non-slip tied joint model requires the use of a new block. The relevant portions of the input file for the Tied-joint model are seen in Figure 7-2. Using the tied-joint model results in two virtual nodes being created. The exodus output file obtained from using the tied-joint approach is then used as the input geometry file for the conventional non-slip method, and the extra nodes are included using MPCs as explained later.

7.1.2.4. Slip

```
Tied Joint
        Normal Definition = slip
                surface 1,2
        Shear Definition
                side = rrod
                connect to Block 3
end

Block 3
        Spring
                Kx = Elastic 1e9
                Ky = Elastic 1e9
                Kz = Elastic 1e9
end
```

Figure 7-3. – Tied-Joint Slip Input.

The geometry file used for the tied-joint slip input is also the original exodus file created from Cubit. However, some changes to the Sierra/SD input file are made to incorporate slipping. In the Tied-Joint block the normal definition is set to slip and the side is set equal to “rrod” under the shear definition. Everything else in the file is kept the same, as seen in Figure 7-3. The output of the tied-joint slip file creates two extra blocks that constrain the overlapping surfaces from stretching, allowing the surfaces to move together as one. This output is in turn used in the input file of the conventional slip model, as described later.

7.1.3. Building the Conventional Model

7.1.3.1. Non-slip

The input model used for the conventional approach is the output of the tied-joint model. The tied-joint model produces an additional block to connect the virtual nodes that are created internally, and thus an additional block with spring or joint2g properties is explicitly added to the input file of the conventional method. The difference between the joint2g and the spring properties, is that the joint2g includes rotational degrees of freedom, everything else within the

input file remain the same. The input file requires rigidsets and MPCs linking the duplicate nodes that the tied-joint model creates to the “original” nodes on the corresponding faces. The rigidset input section with the spring connection is seen in Figure 7-4.

```
Rigidset
        sideset 1
end
Rigidset
        sideset 2
end

Block 33
        Spring
                Kz = Elastic 1e9
                Kx = Elastic 1e9
                Ky = Elastic 1e9
end
```

Figure 7-4. – Conventional Non-Slip Input.

7.1.3.2. Slip

The geometry file used for the conventional slip input is the output from the tied-joint slip input. The extra blocks created from the tied-joint slip output are defined "dead" for this input file when using a spring. In their place, a new section called Tied Data is added to incorporate slipping. When a joint2g is used, these extra blocks are defined as "rbe3", replacing the use of MPCs. This can be seen in Figure 7-5. The Tied Data is specified to be a transverse slip that applies to the overlapping surfaces. Rrodsets are also added instead of the rigidsets that are seen in the conventional non-slip input file. Figure 7-6 shows a section of the input file when using a spring connection for conventional slip.

```
Rrodset
        sideset 1
end
Rrodset
        sideset 2
end

Block 3
        Joint2G
                Kx = Elastic 1e9
                Ky = Elastic 1e9
                Kz = Elastic 1e9
                Krx = Elastic 1e9
                Kry = Elastic 1e9
                Krz = Elastic 1e9
end
Block 4
                rbe3
                method=new
end
Block 5
                rbe3
                method=new
end

Tied Data
        surface 1,2
        transverse slip
end
```

Figure 7-5. – Conventional Slip Input with Joint2G.

```
Rrodset
        sideset 1
end
Rrodset
        sideset 2
end

Block 3
        Spring
                Kx = Elastic 1e9
                Ky = Elastic 1e9
                Kz = Elastic 1e9
end

Tied Data
        surface 1,2
        transverse slip
end
```

Figure 7-6. – Conventional Slip Input with Spring.

7.1.4. Comparison of Results

Exodiff was used to compare the tied-joint and the conventional model for both the slip and non-slip models. While the results from using the tied-joint method and the conventional method were not exactly the same, they were extremely close. These results show that the tied-joint method is just as accurate as the conventional approach. This, in addition to the previously mentioned advantages of offering the user a simpler input and model definition, make the case for the continued use of Tied-Joints in Sierra/SD.

For input deck see Appendix 9.69.1 and Appendix 9.69.2.

7.2. Slide RBE2. Selected DOFs

This test exercises the RBE2 element as a slider. The model and results are shown in Figure 7-7. The base plate is clamped. The perpendicular plate is clamped on the left, and pulled from the right. We are interested in the behavior of the RBE2 links that connect the two plates.

In this example, the RBE2 (which are translated as RBARS in Nasgen) provide a connection in only selected dofs. In particular, the 13456 dofs are constrained, while the 2 is left free. This leaves translation in the Y axis unconstrained.

Figure 7-7 indicates a uniform displacement in the Y direction on the loaded side of the perpendicular plate. This is in agreement with the NASTRAN results. NASTRAN results indicate a maximum displacement of 0.00213, while the QuadT displacement is 0.0023220022994. The discrepancy is expected based on the difference in element formulations. The results indicate that the plate is free to translate, but constrained in the other directions.

Figure 7-8 uses an identical geometry but the load is augmented with a Z component of load. As can be seen in the example, addition of an orthogonal loading does not restrict the sliding behavior.

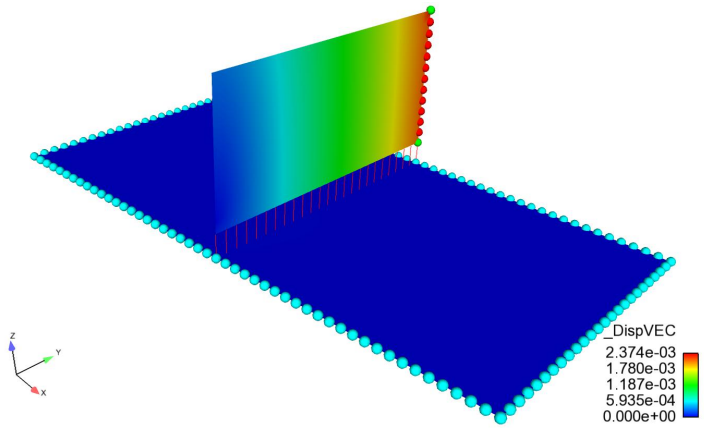


Figure 7-7. – Model and Results of Selective DOF RBE2 Test.

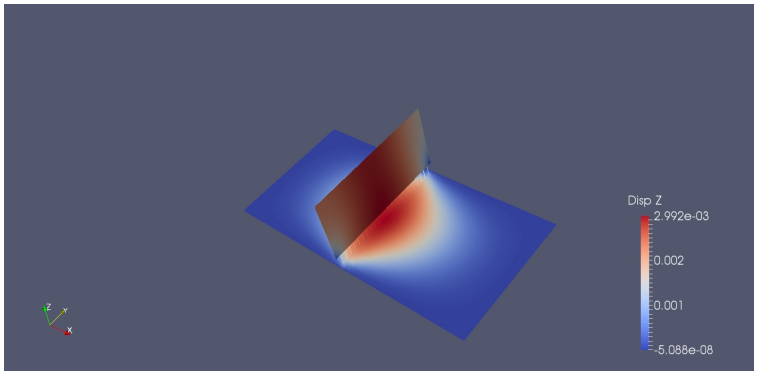


Figure 7-8. – Model and Results of Orthogonally loaded Test.

For input see Appendix 9.70

7.3. Rigidset Compared to Rbar

The purpose of this test is to verify Rigidsets. Verification means that the Rigidsets do the same thing as an equivalent block of Rbars.

A Rigidset is a tool to define a set of nodes as completely Rigid. It is done by creating a sideset (or a nodeset, but sidesets are preferred) and defining that sideset as a Rigidset in the input deck. While Rbars can be used to produce the same rigidity, the process with Rigidsets is much easier. Setting up an equivalent block of Rbars involves creating a block of beams that are not redundant, which gets trickier with more nodes. This step can take more time than desired. Then the block is defined with Rbars in the input deck. Rigidsets are much easier to use and produce similar results.

While the results are the same, the means of obtaining them are different. This can be seen through the MPCs (Multi-Point Constraint equations). Consider the single hex model in Figure 7-9. Since this meshed model contains only a single hex, it only has eight nodes. A sideset has been assigned to one of the hex surfaces, shown in green in Figure 7-9. This sideset is used to define the Rigidset. Rbars are defined by three of the edges on this surface, constrained as a block of BEAM elements.

As previously mentioned, the Rigidset is defined by a sideset. A wireframe of the single hex's Rigidset can be seen in Figure 7-10. There are 18 MPCs and three node connections that are used in the constraint equations. The node connections here are between nodes 3 and 4, 2 and 1, 3 and 1, as represented by the dashed red lines in Figure 7-10. There are 6 constraint equations for each connection. Together, these constraint equations make a perfectly rigid surface.

The MPCs for the block of Rbars also create a perfectly rigid surface, but the equations and node connections differ from those used in the Rigidset. Figure 7-11 shows the block of Rbars created from three edges of the surface. Notice that there cannot be a connection between nodes 3 and 4. A connection between nodes 3 and 4 would require an Rbar there, which would cause redundancy in the constraint equations. One of the difficulties in creating a block of Rbars is making sure there are no redundancies. As shown by the dashed red lines, the connected nodes here are 4 and 1, 1 and 2, 2 and 3. Each connection still has 6 constraint equations, making 18 MPCs in all. The result is the same as Rigidsets, but the means of getting there is different.

Rigidsets and Rbars use different constraint equations, but both can create a rigid set of nodes with the same eigenvalues. This means that Rigidsets can be verified by comparing the results to Rbars. For input deck see Appendix 9.67.

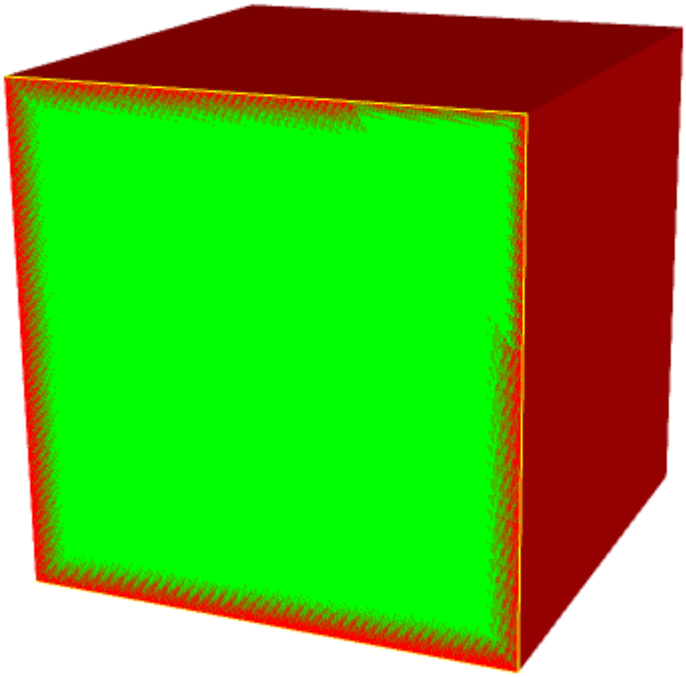


Figure 7-9. – A model of a single hex.

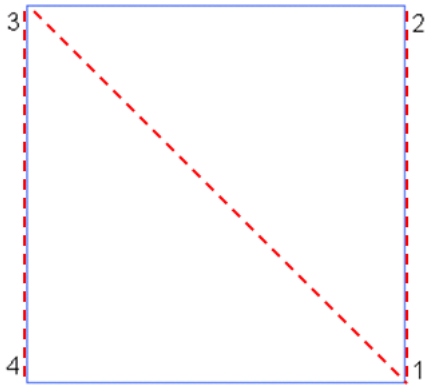


Figure 7-10. – A wireframe view of the sideset used for the Rigidset in Figure 7-9.

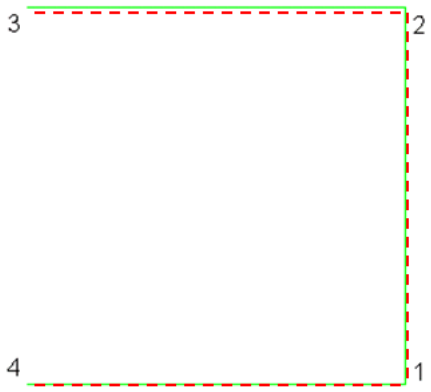


Figure 7-11. – A wireframe view of the block of beams used for the Rbar collection in Figure 7-9.

7.4. Multiple Tied-Surfaces and Curved Surfaces

The purpose of this test is to verify the behavior of multiple tied surfaces. The model is shown in Figures 7-12 through 7-19. Included are several figures that show the model broken down into blocks and the relationships between the surfaces and blocks. Note that Block 3 is actually Block 10 in the input files.

We verify that the eigen analysis retains 6 rigid body modes, and that the structure is appropriately tied on the planar and curved surfaces. Note that 6 rigid body modes are not calculated due to poor conditioning of the constraint matrix if con_tolerance 1e-3 is commented out in the GDSW solver block. Figure 7-20 shows mode 15 of the solution, with a large degree of deformation.

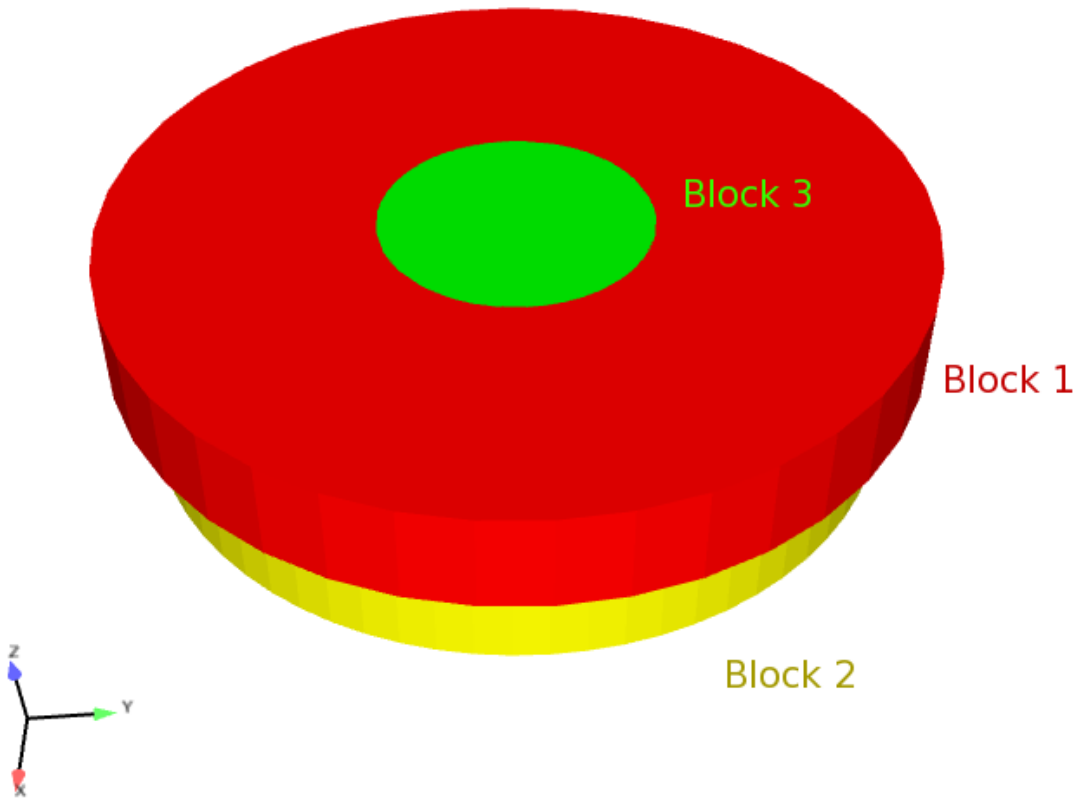


Figure 7-12. – All three blocks from an above angle.

For input deck see Appendix 9.68.

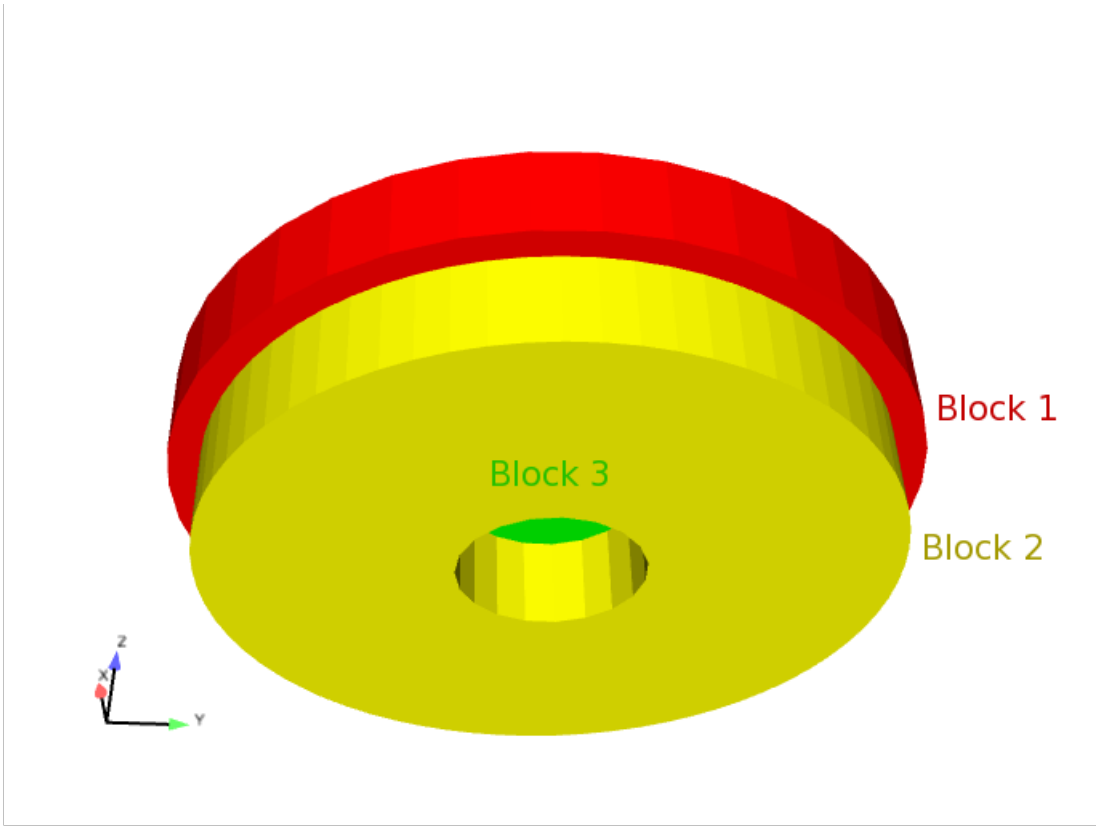


Figure 7-13. – All three blocks from a below angle.

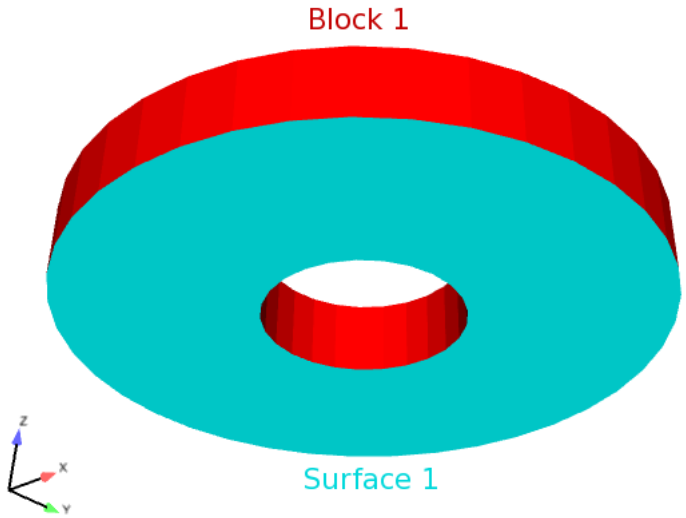


Figure 7-14. – Block 1 and Surface 1.

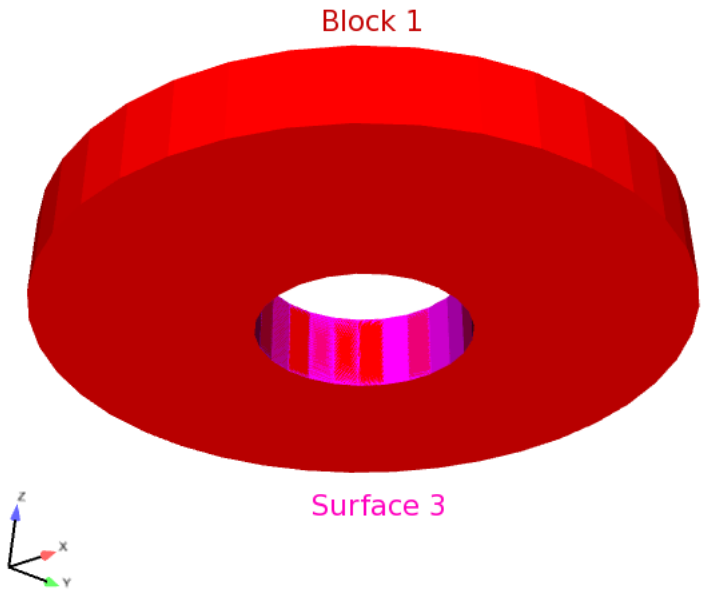


Figure 7-15. – Block 1 and Surface 3.

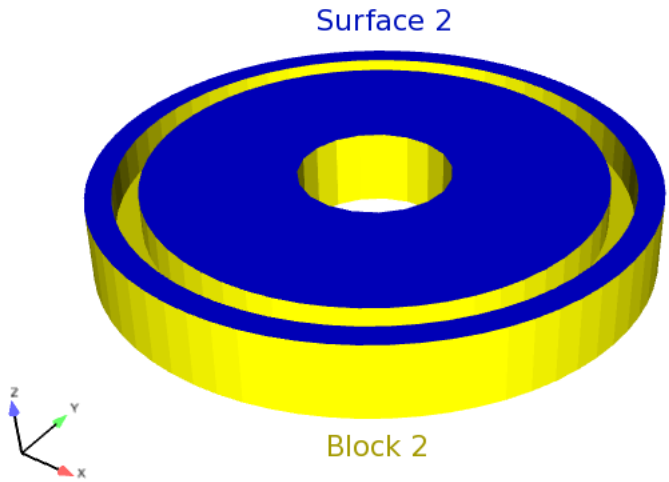


Figure 7-16. – Block 2 and Surface 2.

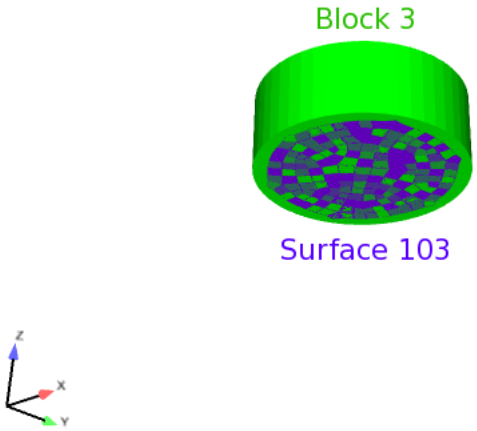


Figure 7-17. – Block 3 and Surface 103.

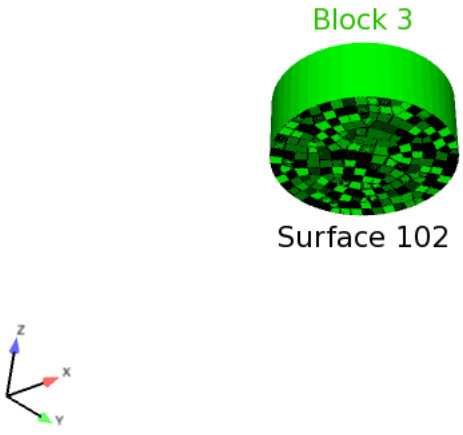


Figure 7-18. – Block 3 and Surface 102.

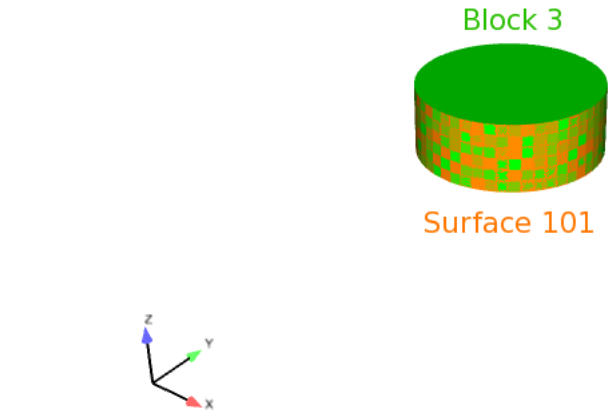


Figure 7-19. – Block 3 and Surface 101.

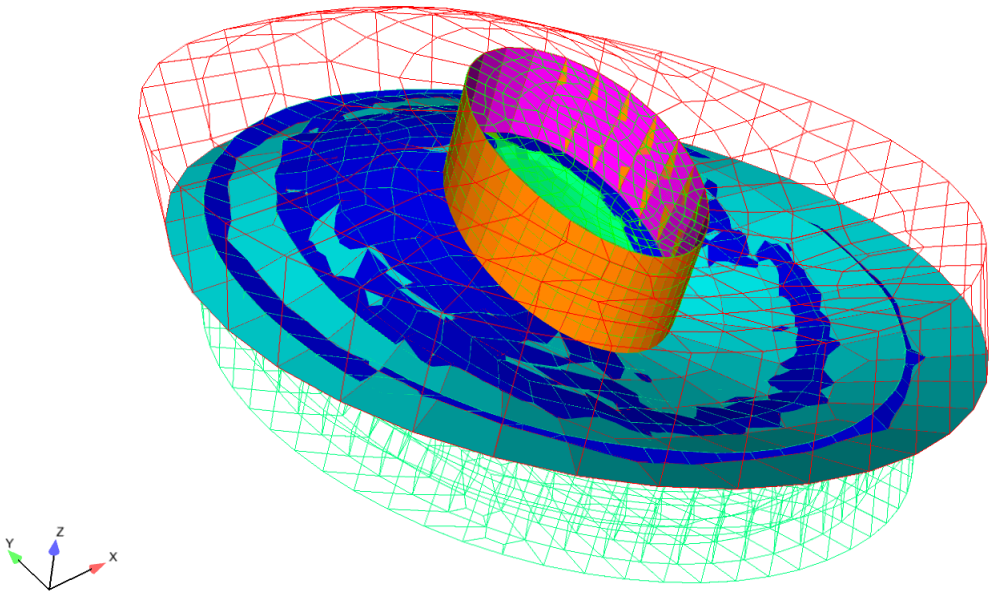


Figure 7-20. – Mode 15 with sideset Tying.

7.5. Contact Verification

In this section we provide a series of verification tests for a conceptually monolithic bar created by tying together separate element blocks. This verification test documents the solution convergence rate for a contiguous mesh versus a discontinuous mesh tied along planar or curved boundaries. Additionally, the test investigates the effect of tied data gap removal and face/node pairings. Evidence based usage guidelines for tied data are provided based on the results.

7.5.1. Description of the Test

The first of the three load cases, shown in Figure 7-21 (a), is a gravity load on a cantilever beam. Next a bar fixed at one and with a traction load on the other is shown in the (b). The last is a free-free eigenvalue problem. To ensure planar notionally 2D results, the Poisson's ratio of the material is set to zero and boundary conditions constrain motion to the xy plane. In the Hex8 element mesh of Figure 7-22, the top mesh is a conforming mesh that is used as a comparison baseline, and refined significantly for a "truth" solution. Straight interfaces between the block partitions are shown in the middle mesh. The bottom mesh uses curved interfaces between the block partitions.

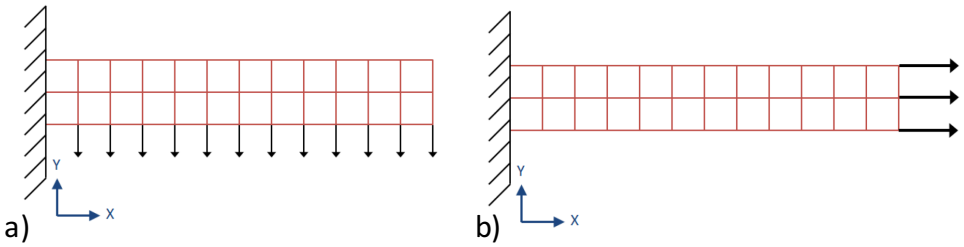


Figure 7-21. – Beam under (a) gravity loading and (b) traction loading.

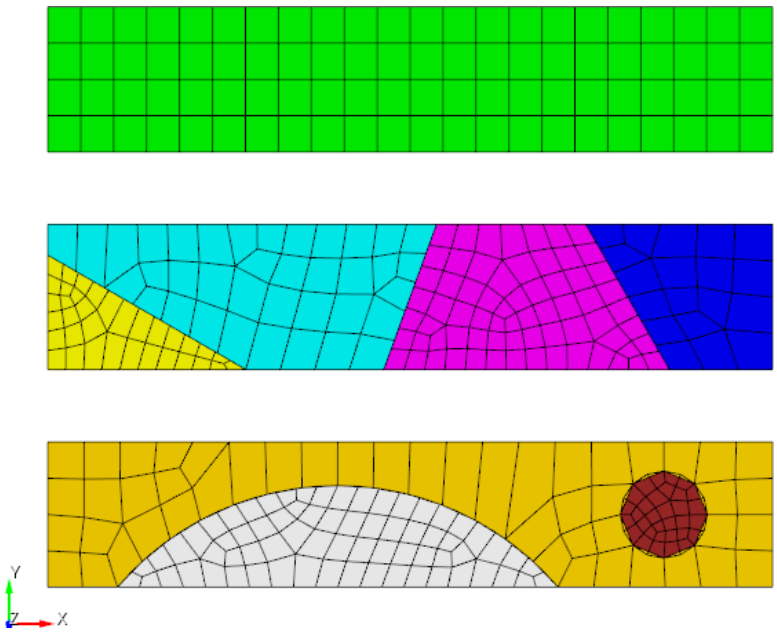


Figure 7-22. – Mesh Geometry.

7.5.2. Expected Results

The eigen modes, cantilever beam displacement, and axial pull solution all have approximate solution based on beam theory. However, as the meshed beam has finite thickness, ultimate verification is done against a "truth" solution generated by a highly refined contiguous mesh.

For the free-free eigen case, the first three modes should be rigid body modes. These tests investigate the preservation of rigid body modes with tied data and the convergence of the first three flexible modes. For the cantilever beam problem, the quantity of interest is tip displacement and total strain energy, again compared versus a highly refined contiguous truth solution. For the axial bar pull analysis the quantity of interest is maximum stress, which is expected to be artificially high when tied interfaces are used. The axial bar pull analysis is effectively a patch test that should produce an exactly known uniform stress state. Any deviation from this expected stress state is considered error.

7.5.3. Evaluation of Free-Free Eigen Load Case

The bar is constrained to deform in plane only. Thus, the bar should have three rigid body modes: two translational, and one rotational. The expected mode shapes for the first three flexible modes are shown in Figures 7-23(a) 535.5 Hz, (b) 1272.6 Hz, and (c) 1453.9 Hz.

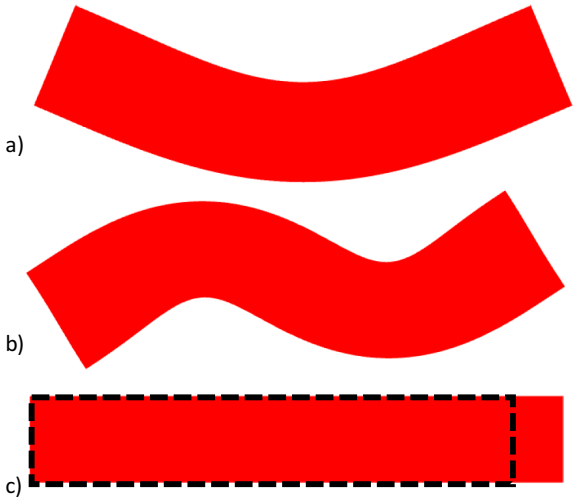


Figure 7-23. – Flexible mode shapes (a) mode 1 (b) mode 2 and (c) mode 3 (non-uniform axial elongation).

7.5.3.1. Convergence Rate for Eigenvalues

The mesh convergence for the first three flexible modes are shown in Figures 7-24(a)-(c). Note the third flexible mode is the axial bar extension mode. This mode approaches the correct solution with very few elements due to the complete lack of any bending in the mode shape. As a result, the convergence plot is not particularly informative, but is shown here for completeness. Generally second order convergence rates are achieved with or without contact. The contiguous mesh tends to have moderately less absolute error at any given refinement.

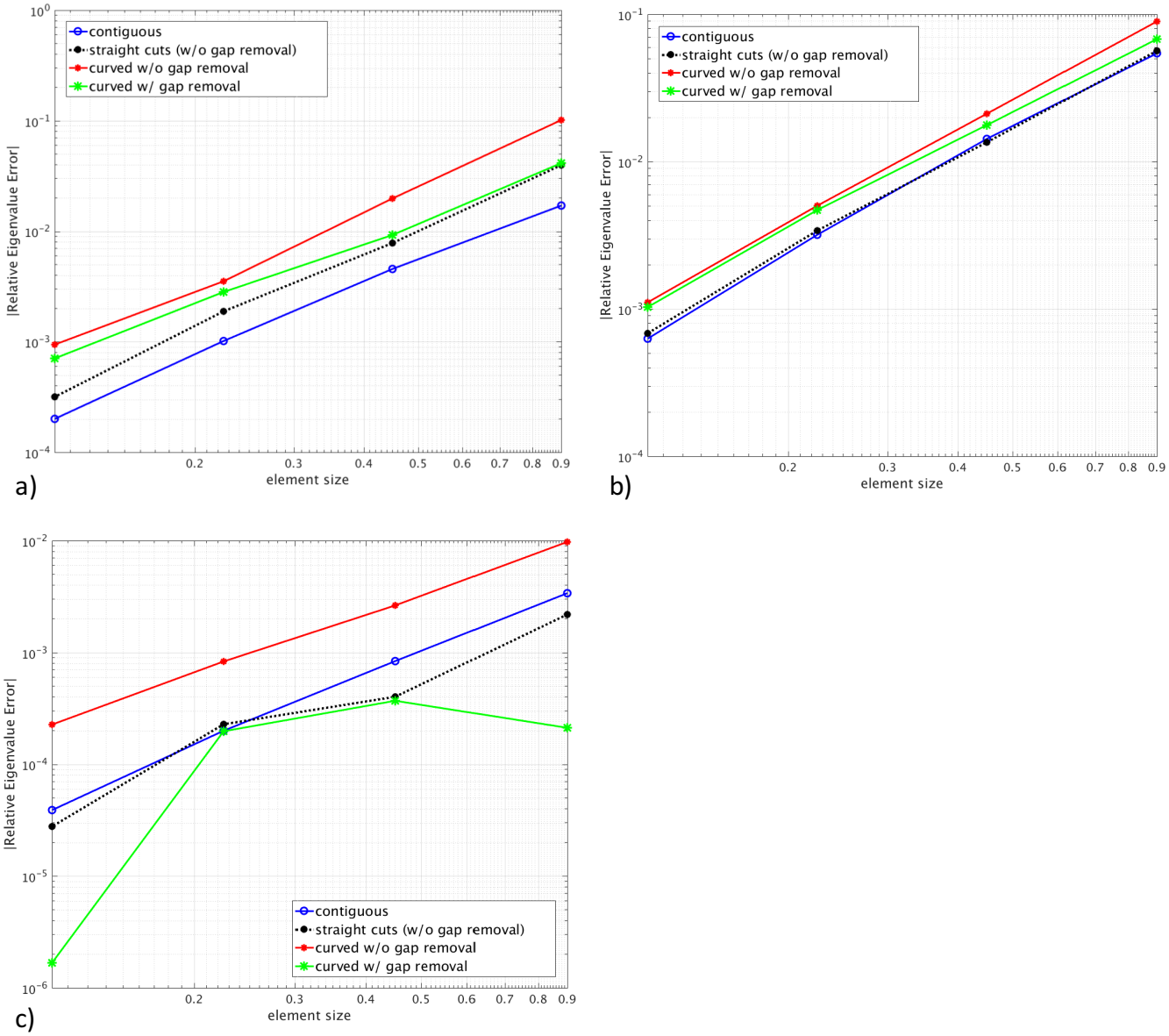


Figure 7-24. – Convergence rates for flexible modes. (a) First elastic mode converges to 534.5 Hz (b) Second elastic mode converges to 1272.6 Hz (c) Third elastic mode converges to 1453.9 Hz.

7.5.3.2. Invariance to Rigid Body Rotation

Figure 7-25 shows how accurately the rigid body rotation mode is preserved. Ideally, this rigid body rotation mode will have zero stiffness. In practice there is a very small stiffness due to round-off errors and finite solver convergence tolerance. However, for the curved contact case with gap removal off there is a very significant error in the rigid body rotation mode. Using the faceted curved cuts, there are finite gaps between the nodes and faces on the two sides of the contact interface. When tied contact constraints are defined across finite gaps, the constraints artificially constrain rotations. The smaller the gap, the less artificial constraint is produced. As the mesh is refined the node to face gap shrinks, and the solution converges toward the exact solution. However, as seen in both the rigid body rotation mode, and the results for the flexible modes, the error from these constraints with gaps is large.

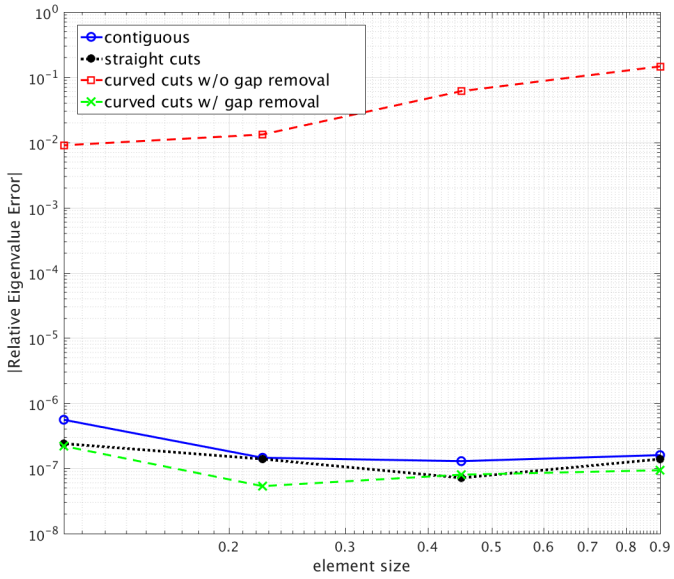


Figure 7-25. – Error in rigid body rotation divided by first flexible mode.

7.5.3.3. Effect of Node Face Interaction Pairing

Simulation fidelity may be enhanced by carefully ordering the pair of surfaces in a tied data interaction. The previous results were made with the recommended setting of using the finer meshed surface as the node surface, and the coarser surface as the face (faces). The face and node surfaces are selected by the order of surfaces in the tied data section of the input deck. As an example, the below syntax selects the nodes of surface 101 as the nodes and the faces of surface 100 as the face.

```
TIED DATA
  SURFACE 100, 101
END
```

The opposite face node pairing is given by:

```
TIED DATA
  SURFACE 101, 100
END
```

If face and node surfaces are selected properly, MPC_Status, which is specified by the constraint_info output option, will appear as shown in Figure 7-26(a). If the wrong surface is chosen as face, then the results appear as shown in Figure 7-26(b). Notice that many nodes on the tied surfaces have begun to separate from, or penetrate into, the opposing surface. This is a result of the relative refinements between the two surfaces. In the incorrect example, the more refined surfaces were chosen as the face surface, and many interactions were missed. The reason for this lies in the way that tied data functions; specifically, tied data requires that all nodes on the node surface lie on the faces of the face surface, but does not impose the same requirement on the nodes of the face surface. If both surfaces are at approximately the same refinement, it does not matter which side is the face surface, but when the face surface is at a significantly higher refinement than the node surface, there will be some faces of the face surface which are not constrained to any nodes, and are allowed to move without any stiffness contribution from the node surface. Note that the MPC_Status variable is not a foolproof check of correct interactions. It clearly shows the issues on the small circular region, but is not a sufficient check on the larger arc.

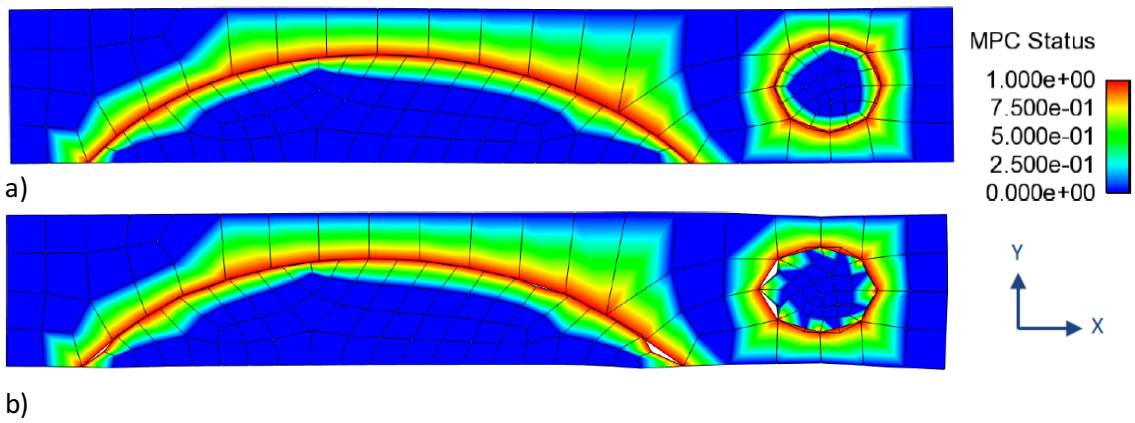


Figure 7-26. – MPC Status (a) correct and (b) incorrect.

The eigen mode convergence with reversed face/node interactions is shown in Figure 7-27(a)-(c). With the non-recommended face/node pairing the convergence rate becomes sporadic. The eigen shape solution will contain obvious errors local to the contact interface. A decent eigen value solution can sometimes be obtained when these errors cancel. On the whole though, the eigen value solutions are much worse with the non-recommended face/node pairings.

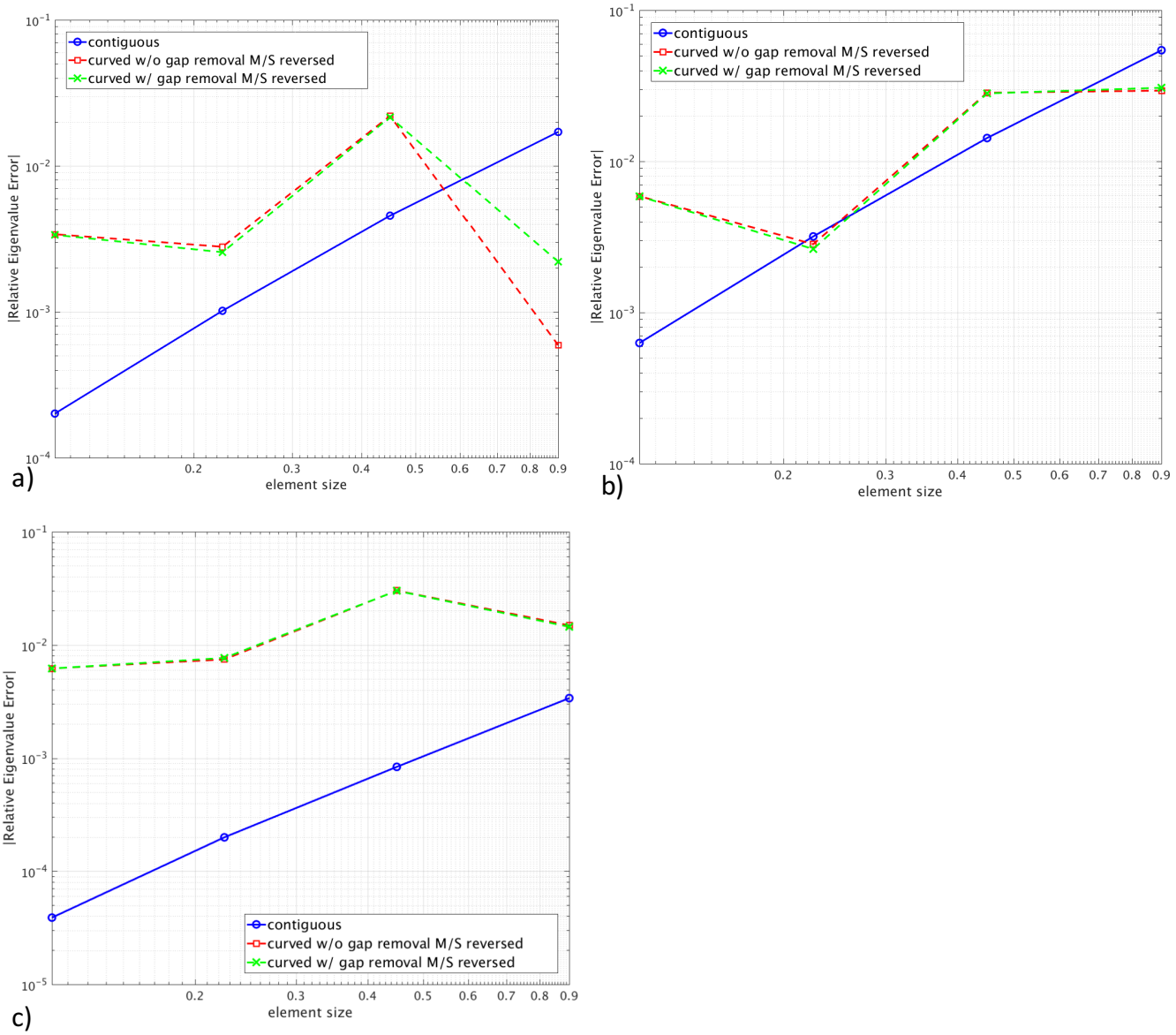


Figure 7-27. – Flexible mode convergence rates with reversed face/node. (a) First flexible mode converged to 534.5 Hz. (b) Second flexible mode converged to 1272.6 Hz. (c) Third flexible mode converged to 1453.9 Hz.

7.5.4. Evaluation of Cantilever Beam Static Results

The result for contiguous cantilever beam is shown in Figure 7-28.

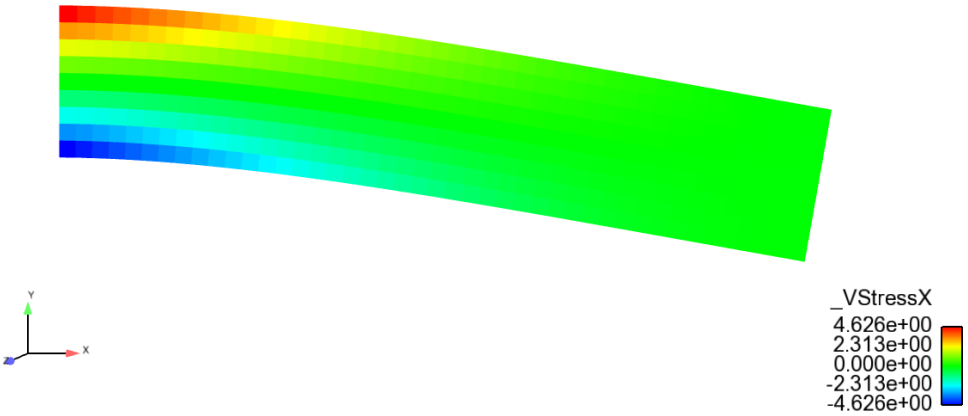


Figure 7-28. – Cantilever Beam Deformed result (greatly magnified).

Convergence Rate The mesh convergence of tip displacement for the cantilever beam is shown in Figure 7-29. Convergence is quadratic with or without contact. As in the eigen mode solution, addition of contact does add some error for a given mesh density. Likewise, the presence of finite gap constraints introduces additional error into the solution.

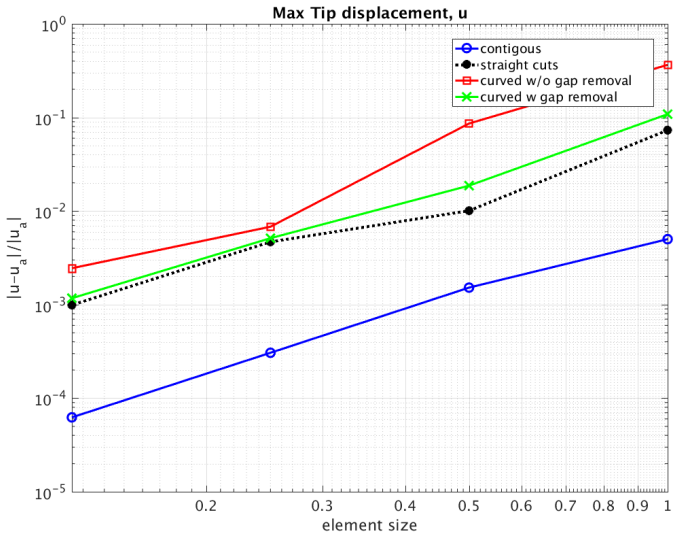


Figure 7-29. – Cantilever Beam Convergence For Tip Displacement.

7.5.4.1. Symmetric Contact

It is possible to inadvertently add symmetric contact to a model. In symmetric contact the nodes of surface one are constrained to the faces of surface two while simultaneously the nodes of

surface two are constrained to the faces of surface one. For example, including both the following tied data sections in an input deck would add symmetric contact to a model:

```
TIED DATA
  SURFACE 101, 100
END
TIED DATA
  SURFACE 100, 101
END
```

Symmetric contact is not expected to work correctly. Symmetrically constrained interfaces are over constrained. Such interfaces can rotate, stretch, and shear, but they cannot bend. The convergence of the cantilever bar with symmetric constraints is shown in Figure 7-30. With symmetric constraints there is no convergence to the correct solution. As seen in Figure 7-31 the symmetric contact interfaces cannot bend, leading to a completely spurious displacement and stress result.

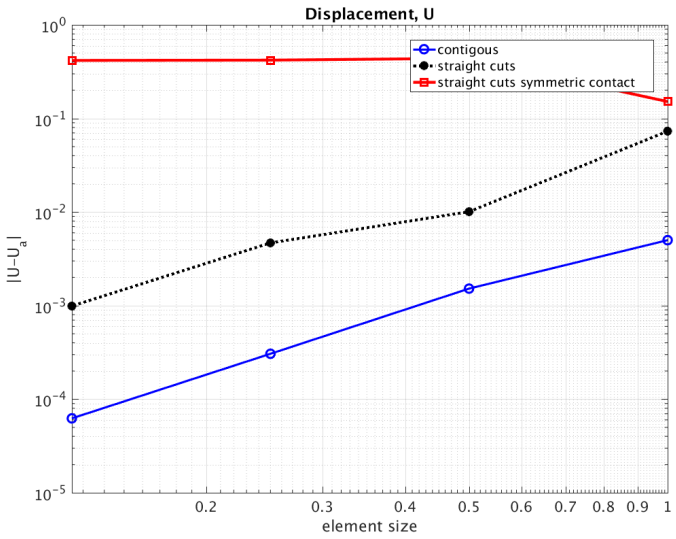


Figure 7-30. – Cantilever Beam Convergence with Symmetric Constraints.

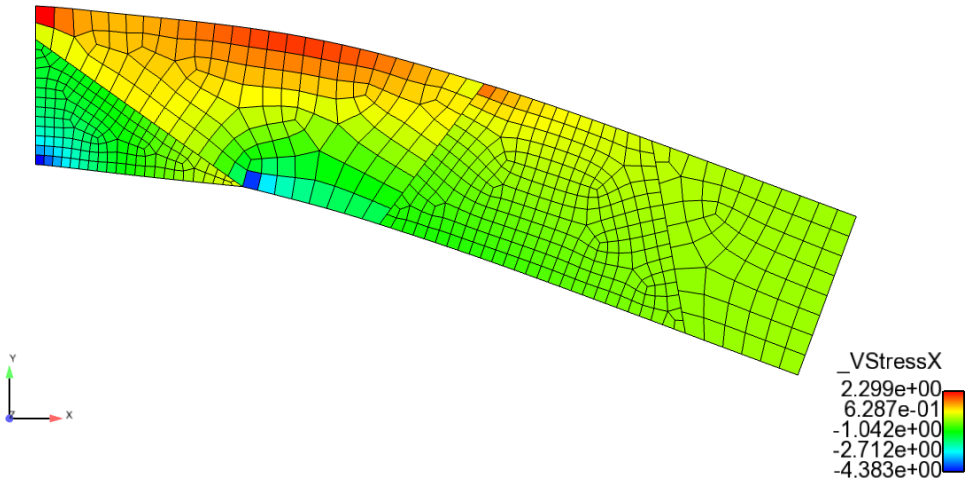


Figure 7-31. – Incorrect Cantilever Beam Result with Symmetric Contact.

7.5.5. Evaluation of Axial Pull Results

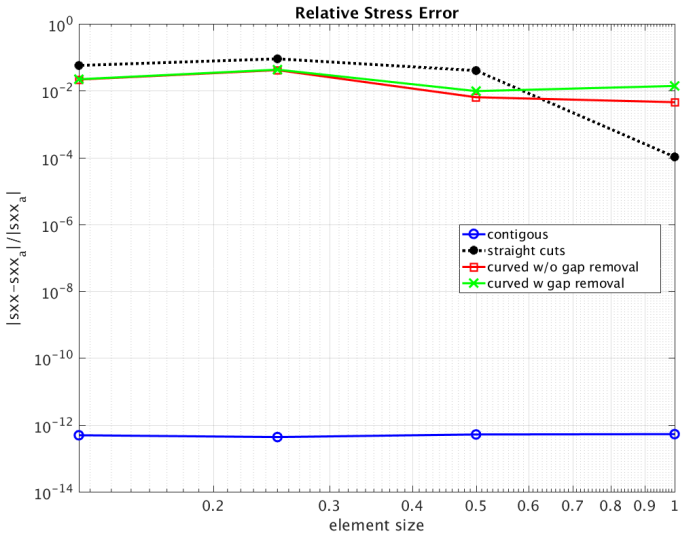


Figure 7-32. – Axial Pull Convergence for Maximum Stress.

An axial pull produces an exact uniform XX direction stress of 1000. However, the nature of tied contact constraints produces artificial stress concentrations at the contact interface. Figure 7-32 shows the convergence of stress. Figure 7-33 shows the distribution of stress on two mesh resolutions. The magnitude of tied data stress concentrations are not remedied by mesh refinement. The stress concentrations do become somewhat more localized with mesh refinement.

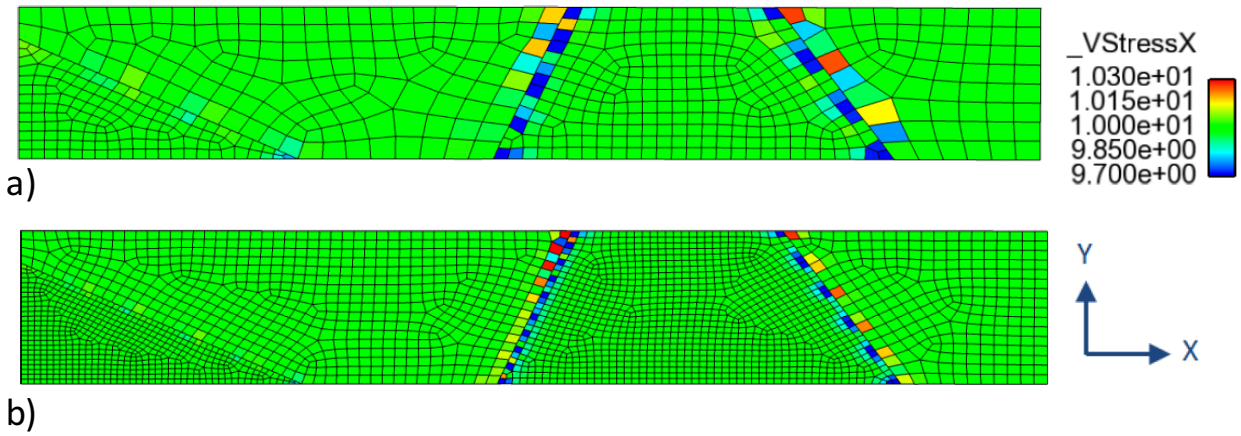


Figure 7-33. – Spurious Local Stress Concentrations with (a) coarse and (b) fine meshes.

7.5.6. Usage Guidelines

Used carefully, tied data can greatly simplify the model creation process by eliminating the need for contiguous meshes. However, there are some significant areas for concern when setting up tied data.

- Models using tied data can achieve quadratic convergence for both eigen modes and static displacement. However, results will generally by at least mildly inferior to a contiguous mesh at the interface.
- Using gap removal will significantly improve the accuracy of contact at curved interfaces.
- For optimal accuracy, the finer meshed surface should be used as the 'nodes' of tied data interactions and the coarser surface the 'faces'.
- Symmetric contact constraints should always be avoided as they lead to major errors and a non-convergent solution.
- Tied contact introduces irresolvable local stress concentrations at the tied interface. If an accurate stress is needed near the tied interface, a contiguous mesh should be used.

For input see Appendix 9.71

7.6. Moving Mesh MPCs: 1D Balloon Pop waveguide

This verification test demonstrates the use of acoustic multipoint constraints (MPC) to tie together two waveguides at different pressures. The waveguide configuration and boundary and initial conditions used in this test mimic a balloon popping in 1D. The purpose of this test is to verify the correction used to equilibrate a multipoint constraint as described in the theory document. This test will verify that the pressure jump between the two structures is in equilibrium after three steps and equilibrates to the average pressure of the two domains. This is verified in the test file by using compare values to ensure both nodes across the interface are equal after three time steps.

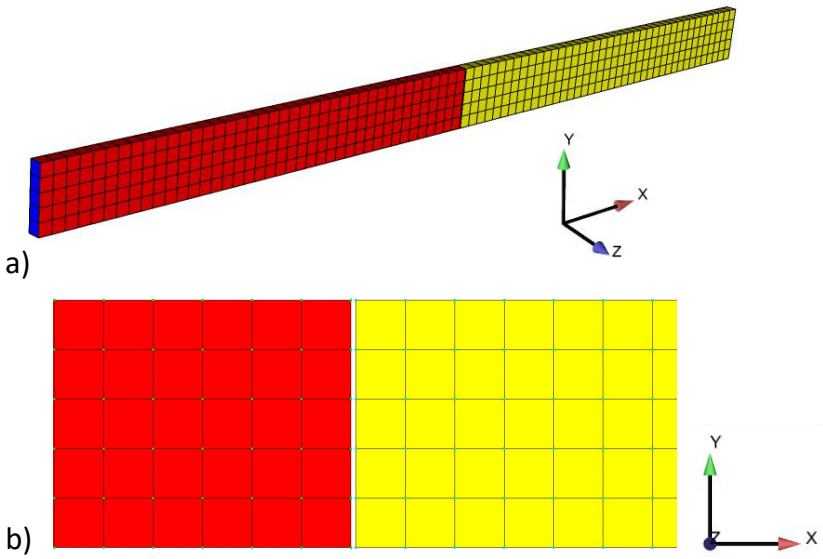


Figure 7-34. – a) Schematic of two 4.0 x 0.5 x 0.1m wave guides with block 1 in red at 4Pa and block 2 in yellow at 2Pa. (b) Close-up of the gap where constraints will tie together the pressure across the interface.

The waveguide configuration shown in Figure 7-34 represents a 1-D equivalent of a balloon popping with mirror symmetry at the centerline of the balloon. The over pressured block in red represents the balloon containing air at a high pressure and the surrounding lower pressure atmosphere is shown in yellow. A free surface boundary condition is used on the end of the red domain and absorbing conditions are placed on the end of the yellow domain. The initial conditions for the red block is 4Pa and the yellow block is at 2Pa. At time t=0, the balloon is popped and the pressure waves will propagate away from the red-yellow interface. When the pressure waves reach the end of the red domain they will reflect with opposite phase.

Each block of the wave guide in Figure 7-34 is 4.0 x 0.5 x 0.1 meters meshed with 0.1m Hex8 elements. The blocks are separated by a 1cm gap. The purpose of the gap is for visualization only, it has no effect on how the inhomogeneous MPCs tie together the two domains. By including the gap, clearly no nodes are being shared between the blocks. The initial pressure of block 1 is set to 4 Pa and block 2 is set to 2 Pa using an `initial-conditions` block with `acoustics = by_block` in the input file. The `acoustics` initial condition refers to applying an initial condition on the primary variable. For the pressure form of the acoustic equation used in this test, the

primary variable is pressure. A block by block application of the initial conditions is set using `by_block`, and each block in the input file that an initial condition is applied to must have the keyword `acoustics` followed by its value. The `LOAD` block is used to apply an `acoustic_accel` that is zero for all time. This switches the acoustics formulation to use pressure as the primary variable instead of the velocity potential. A similar two block conformally meshed waveguide using the same input file is used to verify the results.

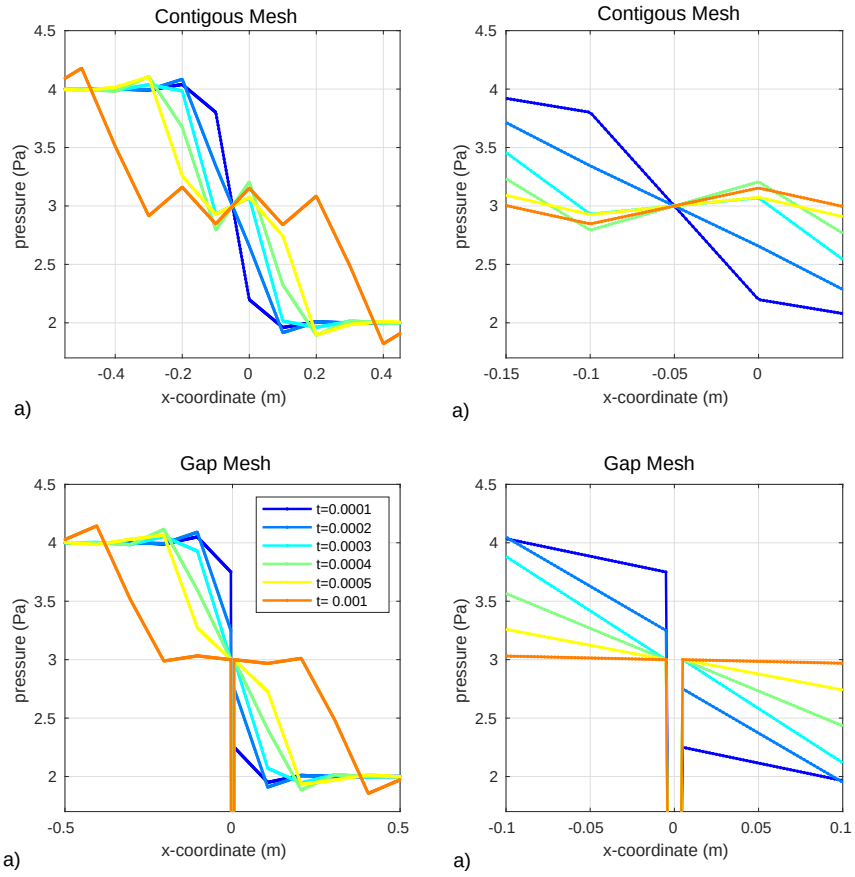


Figure 7-35. – Pressure profiles measured across the length of the waveguide are shown at the times given in the legend shown in (c). (a) and (b) show far and near field pressure profiles for the contiguously/conformally meshed waveguide. (c) and (d) show pressure profiles for the mesh containing a gap and constraints.

Figure 7-35 shows the first time steps after the balloon is popped for the conformal mesh and mesh containing a gap with constraints. Data for these plots is obtained using `linesample` with 1000 sample points taken between $-1 \leq x \leq 1$. The location of the interface is shifted by half an element (0.05m) for the conformal mesh because of where the nodal pressure initial conditions are applied. The pressure profiles for the conformal mesh are shown in Figure 7-35 (a) and (b). For the conformal mesh, the nodal pressure at x=0 is initially 2Pa and increases to 3Pa over 3 time steps while the nodal pressure at x=0.1m is initially at 4Pa and drops to 2Pa. The interpolated pressure in the element at x=-0.05 is held constant at 3Pa for the time steps shown. Nodal

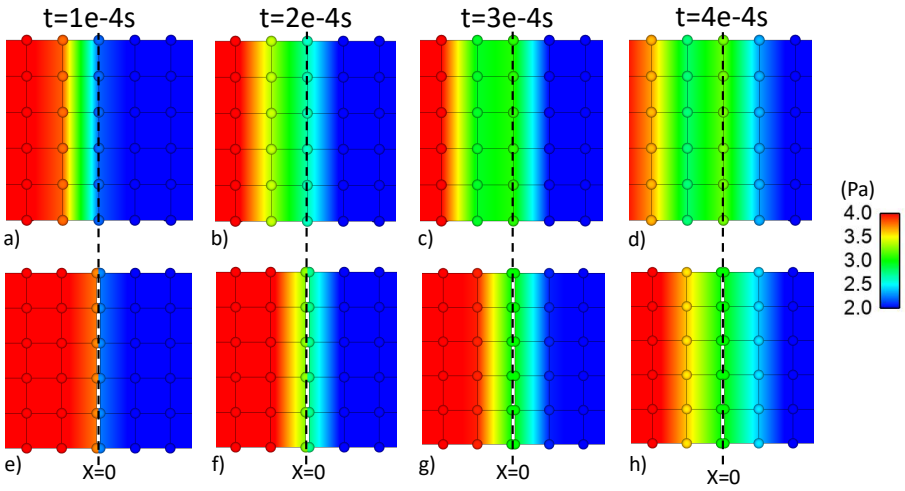


Figure 7-36. – Nodal pressure output shown on the meshed geometry for the time steps plotted in Figure 7-35. (a)-(d) are for the conformal mesh and (e)-(f) are for the constrained mesh containing a gap. The dashed line indicates x=0 for both meshes.

pressures for the conformally meshed geometry are shown visually in Figure 7-36 (a)-(d).

The nodal pressures for the gap mesh with MPCs is shown visually in Figure 7-36 (e)-(h). The different nodal pressure across the gap can be seen for (e)t=1e-4s and (f)2e-4s. The blue pressure profiles for t=1e-4s and 2e-4s in Figure 7-35 (d) show the size of the pressure jump across the interface. The MPC correction brings the nodal pressures into equilibrium across the interface in three steps as shown by the nodes being the same color in Figure 7-36 (g) for t=3e-4s. Pressure equilibrium is shown by the continuity of the cyan line in Figure 7-35 (c) and (d).

The pressure profiles between the two meshes in Figure 7-35 are nearly identical once equilibrium is enforced. The delayed enforcement of equilibrium caused by the MPC correction leads to smaller pressure oscillations at later times. The delay in pressure enforcement also causes a small delay in the pressure pulse. The delay can be reduced by reducing the time step as shown in Figure 7-37. The peak of the pressure profiles for the conformally meshed waveguide at t=0.015s is approximately at x=-2.6m. This is nearly the same value for dt=5e-5 and is 0.1m ahead of dt=1e-4 and 0.3m ahead of dt=2e-4.

Increasing the time step is also shown to smooth out the profile of the wave. For the largest time step of dt=1e-3 shown in Figure 7-37, the wave would travel nearly 1m or 10 elements over the three steps required by the MPC's to reach equilibrium.

For input see Appendix 9.72

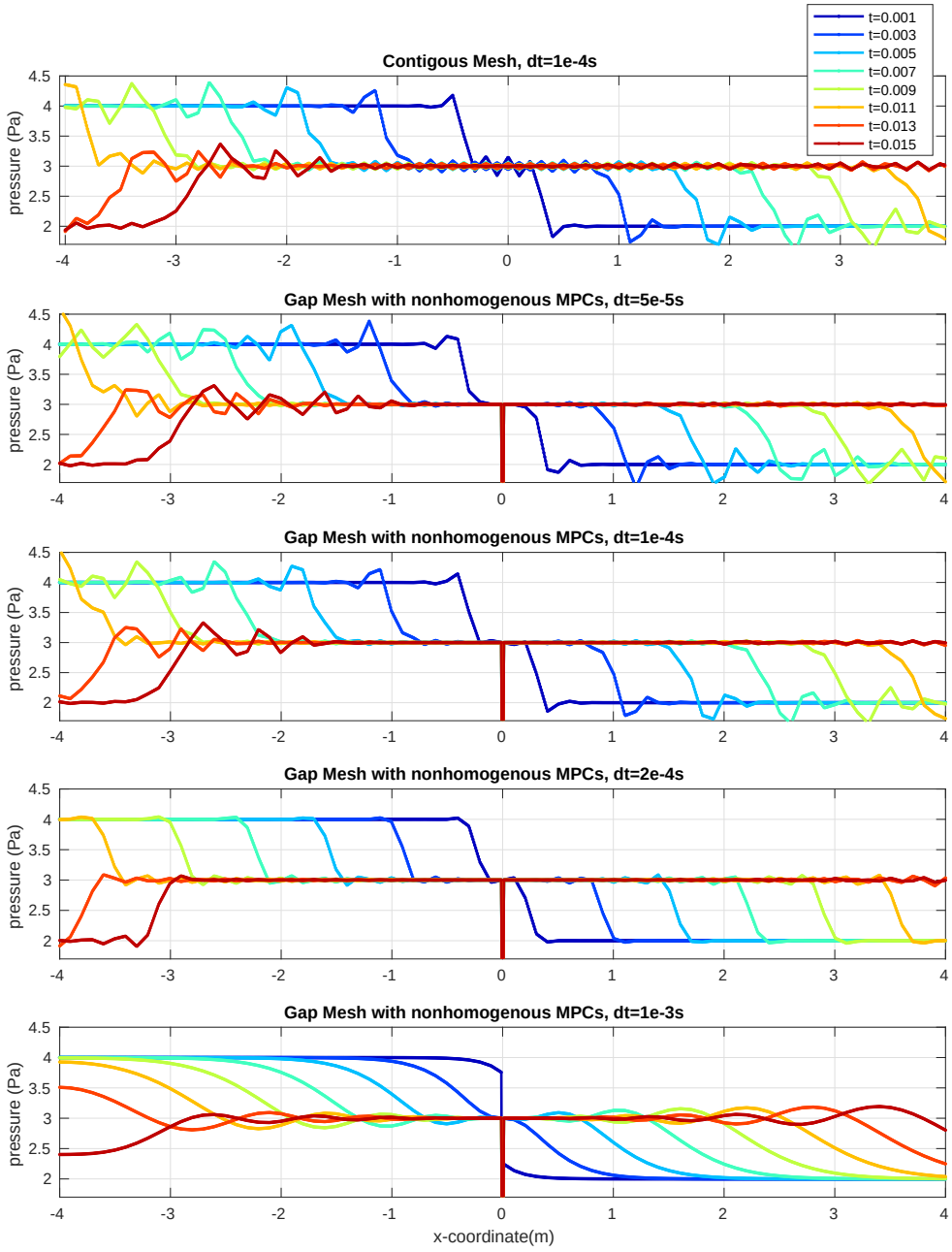


Figure 7-37. – Effect of time step on the pressure profile for the MPC mesh containing a gap compared to the contiguous/conformal mesh at the top.

This page intentionally left blank.

8. LEGACY TESTS

Table 8-1. – Nightly tests corresponding to this chapter.

Dir/Name of Test	Doc. Table	Row	Ref Table	Element Type
beam_analytic/cantilever_free_beam2_test	8-32	2	*3 1a*	Beam2
./cantilever_free_tria3_test	8-32	3	*3 1a*	Tria3
./cantilever_free_tria3r_test	8-32	4	*3 1a*	Tria3⊥
./simply_simply_beam2_test	8-32	7	*3 1e*	Beam2
./simply_simply_tria3_test	8-32	8	*3 1a*	Tria3
./simply_simply_tria3r_test	8-32	9	*3 1a*	Tria3⊥
./cantilever_guided_beam2_test	8-32	11	*3.3b*	Beam2
beam-curved/roark_table17_1_test	8-33	2	*17.1*	Beam2
./roark_table17_2_test	8-33	4	*17.1*	Beam2
./roark_table17_3_test	8-33	6	*17.1*	Beam2
beam_eigen/free_free_test	8-17	2	*8-1.1*	Beam2
beam_eigen/free_sliding_test	8-17	4	*8-1.2*	Beam2
beam_eigen/clamped_free_test	8-17	6	*8-1.3*	Beam2
beam_eigen/pinned_pinned_test	8-17	8	*8-1.5*	Beam2
beam-mass/blevins_table6-2_19_test	8-16	2	*6-2.19*	Beam2
beam-mass/blevins_table6-2_20_test	8-16	4	*6-2.20*	Beam2
beam-mass/blevins_table6-2_22_test	8-16	6	*6-2.22*	Beam2
plate_annular/roark_table24_1a_test	8-34	2	*24.1a*	Tria3
plate_annular/roark_table24_1b_test	8-34	4	*24.1b*	Tria3
plate_annular/roark_table24_1e_test	8-34	6	*24.1e*	Tria3
plate_rectangular/roark_table26_1a_test	8-35	3	*26.1a*	QuadT
plate_rectangular/roark_table26_1a_t_test	8-35	2	*26.1a*	Tria3
plate_rectangular/roark_table26_8a_test	8-35	6	*26.8a*	QuadT
plate_rectangular/roark_table26_8a_t_test	8-35	5	*26.8a*	Tria3
spring-mass/blevins_table6-2_2_test	8-15	2	*6-2.2*	spring
spring-mass/blevins_table6-2_18_test	8-15	4	*6-2.18*	spring

Tables 8-1

identifies the nightly tests corresponding to the verification tests described here. Tests are available on Sandia's Restricted Network in `/projects/sierra/tests/Salinas_rtest/` in the subdirectories `test_tool` or `verification`.

Table 8-2. – Test Matrix (cont).

Dir/Name of Test	Doc. Table	Row	Ref Table	Element Type
thinShellsOfRevolution/.				
./roark_table28_1a_hex8_test	8-36	3	*28.1a*	Hex8
./roark_table28_1a_tria3_test	8-36	2	*28.1a*	Tria3
./roark_table28_1b_hex8_test	8-36	6	*28.1b*	Hex8
./roark_table28_1b_tria3_test	8-36	5	*28.1b*	Tria3
shaft/fixed_free_beam2_test	8-18	2	*8-19.2*	Beam2
shaft/fixed_free_hex8_test	8-18	3	*8-19.2*	Hex8
plate_eigen_circ/free_test	8-20	2	*11-1.1*	QuadT
plate_eigen_circ/simple_test	8-20	4	*11-1.2*	QuadT
plate_eigen_circ/clamped_test	8-20	6	*11-1.3*	QuadT
plate_eigen_circ/clamped_mass_test	8-20	8	*11-1.12*	QuadT
plate_eigen_rect/all_edges_free_test	8-21	2	*11-4.1*	Tria3
plate_eigen_rect/all_edges_fixed_test	8-21	4	*11.4.21*	Tria3
plate_eigen_rect/all_edges_simple_test	8-21	6	*11-4.16*	Tria3
plate_eigen_rect/sFixed_lFree_sSS_lFree_test	8-21	8	*11-4.6*	Tria3

8.1. Linear Acoustics

In the following examples computational results are compared to analytic solutions.

8.1.1 Eigen Analysis of Wave Tube

8.1.2 Eigen Analysis with Multiple Fluids

8.1.3 Eigen Analysis of Elliptic Tank

8.1.5 Direct Frequency Response

8.1.5 Transient Acoustics with Pressure Release

8.1.6 Nonconforming Acoustic-Acoustic Discretizations

8.1.7 Direct FRF of Tied Structural/Acoustics

8.1.8 Radiation from a uniformly-driven spherical shell

8.1.9 Radiation from a spherical acoustic surface

8.1.10 Scattering from a Flat Plate

8.1.11 Transient Scattering from a Flat Plate

8.1.12 Scattering a Plane Step Wave by a Spherical Shell

8.1.13 Infinite Elements on Ellipsoidal Surfaces

8.1.14 Comparison of spherical and ellipsoidal infinite elements

8.1.15 Absorbing Boundary Conditions for Infinite Elastic Spaces.

8.1.16 Impedance Boundary Conditions

5.10 Point Acoustic Source

8.1.17 Moving Point Source

8.1.18 Infinite Elements for Transients

8.1.19 Comparison with Absorbing Boundary Conditions

8.1.20 Acoustic-Structure Directfrf with Viscoelastic Material

8.1.1. Eigen Analysis of Wave Tube

The first example consists of a convergence study for the natural frequencies of an acoustical tube that is driven at the left end and has a rigid cap the right end, as shown in Figure 8-1. The eigenvalue problem for this configuration was solved by uniformly refining a linear hexahedron mesh.

Table 8-3 shows the numerical results, and demonstrates that the first three natural frequencies approach the exact values. Table 8-4 demonstrates quadratic convergence for the natural frequencies, as expected for linear elements.

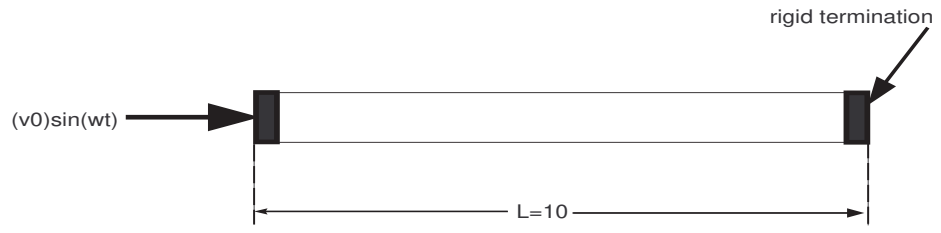


Figure 8-1. – Acoustical waveguide with rigid end cap.

exact (Hz)	80 elements	640 elements	5120 elements	40960 elements
16.6	16.61707	16.60426	16.601065	16.600265
33.2	33.33669	33.23414	33.20853	33.20213
49.8	50.26197	49.9153	49.828799	49.8072

Table 8-3. – Eigenvalue convergence for a piston-driven tube with rigid cap at end. The values given are the natural frequencies, in Hz.

80 elements	640 elements	5120 elements	40960 elements
.0103	.0257	6.415e-3	1.596e-3
.4117	.10283	.0257	6.416e-3
.9277	.2315	.05783	.01446

Table 8-4. – Relative error in computation of natural frequencies for a piston-driven tube with rigid cap at end. The reduction by a factor of 4 each time the element size is halved demonstrates quadratic convergence in natural frequencies.

8.1.2. Eigen Analysis with Multiple Fluids

A subtlety when working with fluids of spatially varying properties is that the linear wave equation, which is typically written in the form

$$\frac{1}{c^2}\ddot{p} - \Delta p = 0 \tag{8.1.1}$$

is no longer valid. Assumptions were made in the derivation of this equation that restricted its applicability to a homogeneous fluid. When density and speed of sound change with position in the fluid, the linear wave equation takes the form[39]

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) - \frac{\ddot{p}}{B} = 0 \tag{8.1.2}$$

where ρ is the fluid density, B is the fluid bulk modulus, and p is the acoustic pressure. If we assume that the speed of sound is $c = \sqrt{\frac{B}{\rho}}$, then this equation can also be written as

$$\rho\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) - \frac{\ddot{p}}{c^2} = 0 \tag{8.1.3}$$

Next, we consider how the heterogeneous wave equation is implemented in **Sierra/SD**. We note that **Sierra/SD** uses the form in equation 8.1.3. Since we want to allow the density to vary with position, we have to first divide by density before multiplying by a test function and integrating by parts. This is because the factor of ρ in front of the first term in equation 8.1.3 varies with position, and thus we will not be able to move the ∇ symbol over to the test function. Thus, we have

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) - \frac{\ddot{p}}{\rho c^2} = 0$$

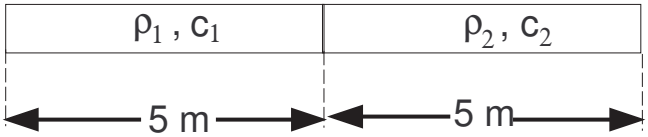


Figure 8-2. – Two fluid acoustic waveguide.

We solve for the time derivative of pressure in **Sierra/SD**. Thus, we substitute $p = \dot{\phi}$ into equation 8.1.2, and then integrate in time to obtain

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) - \frac{\ddot{\phi}}{\rho c^2} = 0$$

The gradient ∇ can be moved to the test function in equation 8.1.2. Thus, this is the formulation that is used in **Sierra/SD** to construct the finite element implementation.

In deriving the analytic solution, we note that the analytical solutions to equations 8.1.2, 8.1.3, 8.1.2, and 8.1.2 will all be the same (assuming we converted the final analytic solution from equation 8.1.2 into pressure), since these equations differ by a scale factor. Thus, we use equation 8.1.2 to derive the analytical solution. If we consider the eigenvalue problem, equation 8.1.2 becomes

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) + \lambda \frac{p}{B} = 0$$

This equation will serve as the basis for deriving the analytical solution.

We consider three cases. All three cases involve the geometry shown in Figure 8-2.

An exact solution for the eigenvalues of the geometry in Figure 8-2 can be derived by considering each fluid separately and applying appropriate compatibility conditions on the fluid-fluid interface. The equations are as follows

$$\frac{d^2 p_1}{dx^2} + \lambda \frac{\rho_1}{B_1} p_1 = 0; \quad x < L/2$$
$$\frac{d^2 p_2}{dx^2} + \lambda \frac{\rho_2}{B_2} p_2 = 0; \quad x > L/2$$

Here B_1 and B_2 are the bulk moduli of the two fluids. The compatibility conditions at the interface $L/2$ are

$$p_1 = p_2, \quad \frac{1}{\rho_1} \frac{dp_1}{dx} = \frac{1}{\rho_2} \frac{dp_2}{dx}. \tag{8.1.4}$$

At the endpoints, there are two options. Either we could have rigid caps ($\frac{dp}{dx} = 0$), or we could

exact (Hz)	computed, h=1	computed, h=0.5
17.7322	17.7505	17.7333
34.1990	34.3411	34.2079
53.1689	53.6642	53.1998

Table 8-5. – Eigenvalue convergence for a two-fluid system with rigid cap at end. The values given are the natural frequencies, in Hz.

have pressure release boundary conditions ($p = 0$). The solution will have the form

$$p_1(x) = C_1 \cos\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) + C_2 \sin\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \quad 0 \le x \le \frac{L}{2} \tag{8.1.5}$$

$$p_2(x) = C_3 \cos\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) + C_4 \sin\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \quad \frac{L}{2} \le x \le L \tag{8.1.6}$$

$$\tag{8.1.7}$$

Inserting these into equations 8.1.2, applying the compatibility conditions 8.1.4, and using the appropriate boundary conditions at the endpoints, we get two transcendental equations that give the exact eigenvalues. For the pressure release (Dirichlet) end cap case, we obtain

$$\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right)\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right) = -\sqrt{\frac{\rho_1 B_1}{\rho_2 B_2}}\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right)\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \tag{8.1.8}$$

$$\tag{8.1.9}$$

For the rigid (Neumann) case, we obtain

$$\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right)\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right) = -\sqrt{\frac{\rho_1 B_1}{\rho_2 B_2}}\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right)\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \tag{8.1.10}$$

$$\tag{8.1.11}$$

Equations 8.1.8 and 8.1.10 can be solved to obtain the exact eigenvalues of the system shown in Figure 8-2.

First, we consider the case $\rho_1 = 1.293$, $\rho_2 = 2.5860$, $c_1 = 332.0$, $c_2 = 366.0$. Table 8-5 shows the comparison when rigid walls are placed at either end of the tube, and Table 8-6 shows the comparison with pressure release conditions at both ends. Convergence is seen in all cases.

The next case is an impedance matching condition, in which $\rho_1 c_1 = \rho_2 c2$. In this case, we take $\rho_1 = 2\rho_2$, and $c_1 = 0.5c_2$. Thus, the parameters are different but the impedances are the same. The computed and theoretical results are shown in Table 8-7. Again, good convergence behavior is observed.

Finally, we consider a case with air and water. The same two-fluid case from the previous example was used, with rigid boundary conditions. The comparison between theoretical and computed eigenvalues is shown in Table 8-8.

exact (Hz)	computed, h=1	computed, h=0.5
17.0965	17.1143	17.0976
35.4575	35.6039	35.4666
51.3135	51.7932	51.3435

Table 8-6. – Two-fluid eigenvalue convergence with pressure release BC.

exact (Hz)	computed, h=1	computed, h=0.5
11.0667	11.0797	11.0675
22.1333	22.2632	22.1414
33.2000	33.6067	33.2256

Table 8-7. – Eigenvalue convergence for a two-fluid system with rigid cap at end. The values given are the natural frequencies, in Hz.

exact (Hz)	computed, h=1	computed, h=0.5
33.1974	33.3341	33.206
66.3825	67.4755	66.4506

Table 8-8. – Eigenvalue convergence for an air/water system with rigid cap at ends. The values given are the natural frequencies, in Hz.

8.1.3. Eigen Analysis of Elliptic Tank

Contributed by Jerry Rouse.
The acoustic modal analysis capability of **Sierra/SD** was further verified using a three-dimensional elliptic cylindrical tank. The dimensions of the tank are shown in Figure 8-3. The verification involved two boundary condition configurations. For the first configuration all boundaries of the enclosure were rigid, which requires the normal component of acoustic velocity be zero at all points along the boundary. For the second configuration, the end caps of the tank were rigid, and the sidewall of the tank was a pressure release surface. A pressure release boundary requires that the acoustic pressure be zero at the boundary.

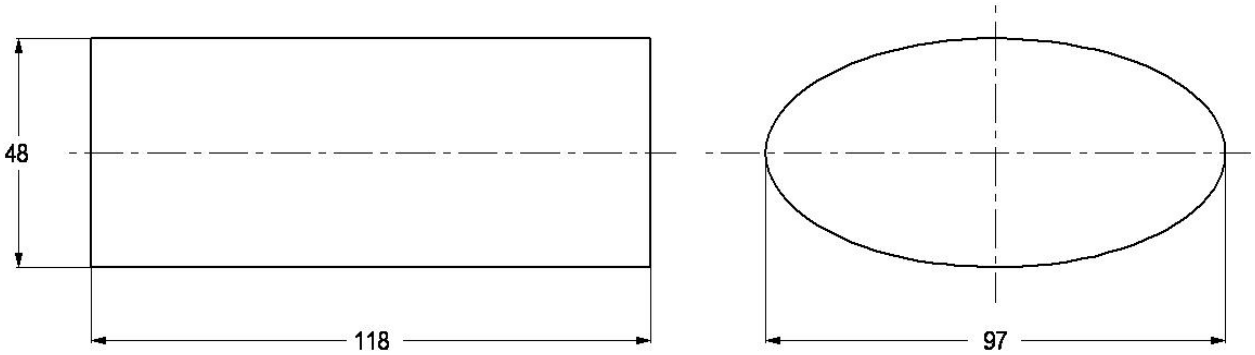


Figure 8-3. – Dimensions of the elliptic cylindrical tank model. All dimensions in inches.

To determine theoretically the resonance frequencies for the elliptic cylindrical tank, the linear wave equation was solved in elliptic cylindrical coordinates. The coordinate system is illustrated in Figure 8-4. This coordinate system is not commonly encountered, and therefore the solution of the wave equation is described. The linear wave equation in terms of acoustic pressure is given by

$$\nabla^2 p + \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0. \tag{8.1.12}$$

In elliptic cylindrical coordinates the Laplacian has the form

$$\nabla^2 = \frac{1}{h^2(\sinh^2(u) + \sin^2(v))}\left(\frac{\partial^2 p}{\partial u^2} + \frac{\partial^2 p}{\partial v^2}\right) + \frac{\partial^2 p}{\partial z^2} \tag{8.1.13}$$

where $x = h\cosh(u)\cos(v)$, $y = h\sinh(u)\sin(v)$, and $h = \sqrt{a^2 - b^2}$ with a equal to half the major axis, and b equal to half the minor axis. For the tank dimensions shown in Figure 8-3 $a = \frac{97}{2}$, $b = 24$, $h = \frac{7\sqrt{145}}{2}$, and $u_0 = \sinh^{-1}\left(\frac{48}{7\sqrt{145}}\right)$. Assuming the acoustic pressure p to be harmonic in time $p = P(u, v, z)e^{i\omega t}$, which upon substitution into Eq. (8.1.12) produces the Helmholtz equation:

$$\nabla^2 P + k^2 P = 0, \tag{8.1.14}$$

where $k = \omega/c$ with ω the angular frequency, and c the phase speed. Using separation of variables $P(u, v, z) = U(u)V(v)Z(z)$. Substituting this expression into the Helmholtz equation

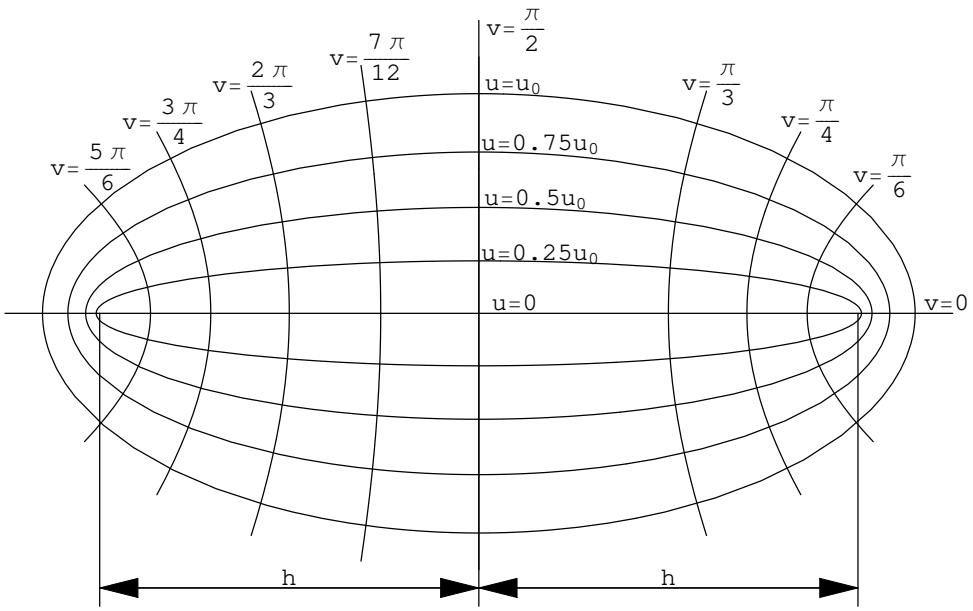


Figure 8-4. – The elliptic cylindrical coordinate system.

and dividing the result by UVZ gives

$$\frac{1}{h^2(\sinh^2(u)+\sin^2(v))}\left(\frac{1}{U}\frac{d^2U}{du^2}+\frac{1}{V}\frac{d^2V}{dv^2}\right)+\frac{1}{Z}\frac{d^2Z}{dz^2}+k^2=0. \tag{8.1.15}$$

Equating the term containing U and V to the separation constant $-m^2$ and the term containing Z to $-k_z^2$ gives the system dispersion relation

$$k^2=k_z^2+m^2. \tag{8.1.16}$$

The differential equation for Z,

$$\frac{d^2Z}{dz^2}+k_z^2Z=0, \tag{8.1.17}$$

has solution

$$Z(z)=A_n\cos(k_zz)+B_n\sin(k_zz). \tag{8.1.18}$$

Simplifying the differential equation for U and V gives:

$$\left[\frac{1}{U}\frac{d^2U}{du^2}+m^2h^2\sinh^2(u)\right]+\left[\frac{1}{V}\frac{d^2V}{dv^2}+m^2h^2\sin^2(v)\right]=0. \tag{8.1.19}$$

The first term is independent of v and the second term is independent of u, therefore each term must equal a constant. Letting c represent this constant:

$$\frac{1}{U}\frac{d^2U}{du^2}+m^2h^2\sinh^2(u)=c\rightarrow\frac{d^2U}{du^2}-\left[c-m^2h^2\sinh^2(u)\right]U=0 \tag{8.1.20}$$

$$\frac{1}{V}\frac{d^2V}{dv^2}+m^2h^2\sin^2(v)=-c\rightarrow\frac{d^2V}{dv^2}+\left[c+m^2h^2\sin^2(v)\right]V=0 \tag{8.1.21}$$

The trigonometric relations

$$\sinh^2(u) = \frac{1}{2}\left(\cosh(2u) - 1\right) \tag{8.1.22}$$

$$\sin^2(v) = \frac{1}{2}\left(1 - \cos(2v)\right) \tag{8.1.23}$$

are used to Eq. (8.1.20) and Eq. (8.1.21). Substitution of these relations into the differential equations for U and V gives:

$$\frac{d^2U}{du^2} - \left[\left(c + \frac{m^2h^2}{2}\right) - \frac{m^2h^2}{2}\cosh(2u)\right]U = 0 \tag{8.1.24}$$

$$\frac{d^2V}{dv^2} + \left[\left(c + \frac{m^2h^2}{2}\right) - \frac{m^2h^2}{2}\cos(2v)\right]V = 0. \tag{8.1.25}$$

Letting $a \equiv c + \frac{m^2h^2}{2}$ and $q \equiv \frac{m^2h^2}{4}$ gives:

$$\frac{d^2U}{du^2} - \left[a - 2q\cosh(2u)\right]U = 0 \tag{8.1.26}$$

$$\frac{d^2V}{dv^2} + \left[a - 2q\cos(2v)\right]V = 0 \tag{8.1.27}$$

These are the canonical forms of the differential equations Mathieu obtained solving for the vibration of an elliptical membrane. The solution to the differential equation for V is given by

$$V = C_r ce_r(a, q, v) + D_r se_r(a, q, v), \tag{8.1.28}$$

where the Mathieu function of the first kind ce has been termed the 'cosine-elliptic' and the Mathieu function of the first kind se has been termed the 'sine-elliptic' by E. T. Whittaker. The solution to the differential equation for U is

$$U = E_r Ce_r(a, q, u) + F_r Se_r(a, q, u), \tag{8.1.29}$$

where Ce and Se are termed the modified Mathieu functions of the first kind. The following relates the Mathieu functions to the modified Mathieu functions:

$$Ce_r(a, q, z) = ce_r(a, q, iz) \tag{8.1.30}$$

$$Se_r(a, q, z) = -ise_r(a, q, iz), \tag{8.1.31}$$

where $i = \sqrt{-1}$.

For the majority of the physical problems encountered, the solution in v is periodic by either π or 2π. This periodicity requires that a relationship exist between q and a for each ce_r and se_r, such that for each non-zero value of q a characteristic value of a exists allowing for a periodic solution in v. Common among authors today is to denote the characteristic values for ce_r by a_r, and the characteristic values for se_r by b_r. Methods for determining a_r and b_r based on q are presented in McLachlan, and Gradshteyn and Ryzhik, with formulas for r up to 8 given in Abramowitz and

Stegun. For the two cases described here, *Mathematica* was used to determine the characteristic values.

For both boundary condition configurations considered, the ends of the elliptical tank were rigid, i.e., acoustic velocity is zero at $z = 0$ and $z = L$. The solution obtained above gives the acoustic pressure in the tank. To apply the zero velocity boundary condition, the momentum equation was used to relate acoustic pressure to acoustic velocity. The momentum equation is

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\vec{\nabla} p, \tag{8.1.32}$$

where $\vec{u} = \vec{e}_u u_u + \vec{e}_v u_v + \vec{e}_z u_z$. The gradient operator in elliptic cylindrical coordinates takes the form

$$\vec{\nabla} = \frac{1}{h\sqrt{\sinh^2(u) + \sin^2(v)}} \left(\vec{e}_u \frac{\partial}{\partial u} + \vec{e}_v \frac{\partial}{\partial v} \right) + \vec{e}_z \frac{\partial}{\partial z}. \tag{8.1.33}$$

Substitution of the z component of pressure in Eq. (8.1.18) into Eq. (8.1.32), and applying the $u_z = 0$ boundary condition gives

$$Z(z) = \sum_{n=0}^{\infty} A_n \cos(k_z z), \tag{8.1.34}$$

where $k_z = \frac{n\pi}{L}$.

The boundary condition configuration having rigid boundaries on all sides of the elliptic cylindrical tank requires the $\vec{e}_u$ acoustic velocity component be 0 at $u = u_0$. Substitution of Eq. (8.1.29) into Eq. (8.1.32) and applying this boundary condition gives

$$\sum_{r=1}^{\infty} \left[E_r \frac{\partial Ce_r(a,q,u)}{\partial u}\bigg|_{u=u_0} ce_r(a,q,v) + F_r \frac{\partial Se_r(a,q,u)}{\partial u}\bigg|_{u=u_0} se_r(a,q,v) \right] = 0, \tag{8.1.35}$$

where $\frac{\partial Ce_0(a,q,v)}{\partial u} = 0$. To satisfy this equation requires each term of the series equal zero, giving

$$\frac{\partial Ce_r(a_r,q_r,u)}{\partial u}\bigg|_{u=u_0} = 0 \tag{8.1.36}$$

$$\frac{\partial Se_r(b_r,q_r,u)}{\partial u}\bigg|_{u=u_0} = 0, \tag{8.1.37}$$

where the resonance frequencies are determined from the values of q which satisfy Eqs. (8.1.36) and (8.1.37). The complete set of resonance frequencies for the elliptic cylindrical tanker having all boundaries rigid is determined from the dispersion relation using the values of k_z in Eq. (8.1.34) and $m = \frac{4\sqrt{q_r}}{h}$ obtained from Eqs. (8.1.36) and (8.1.37)

$$f = \frac{c}{2\pi} \sqrt{\left(\frac{n\pi}{L}\right)^2 + \frac{4q_r}{h^2}}, \tag{8.1.38}$$

where $c = 58724$ in/s. Table 8-9 compares the first 24 resonance frequencies between the exact determination and the **Sierra/SD** prediction for the case of rigid boundary conditions.

Exact (Hz)	**Sierra/SD**	Percent Error
248.832	248.832	0
361.1	361.1	0
438.532	438.533	2.28e-4
497.664	497.665	2.00e-4
614.868	614.87	3.25e-4
659.152	659.156	6.07e-4
687.876	687.879	4.36e-4
704.556	704.56	5.68e-4
731.499	731.503	5.47e-4
746.497	746.501	5.36e-4
825.925	825.932	8.48e-4
829.247	829.253	7.24e-4
849.025	849.035	1.18e-3
900.831	900.843	1.33e-3
934.566	934.58	1.50e-3
950.48	950.495	1.58e-3
982.512	982.529	1.73e-3
995.329	995.346	1.71e-3
995.861	995.878	1.71e-3
1015.1	1015.12	2.00e-3
1029.16	1029.18	1.94e-3
1058.81	1058.83	1.89e-3
1072.88	1072.91	2.80e-3
1130.71	1130.74	2.65e-3

Table 8-9. – Comparison between the exact analytical resonance frequencies and **Sierra/SD** predictions for the elliptic cylindrical tank with rigid boundary boundaries.

The boundary condition configuration having pressure release boundaries $p = 0$ on the sidewall of the elliptic cylindrical tank (and rigid end caps) requires the acoustic pressure be zero at $p(u_0, v, z)$. Applying this condition to Eq. (8.1.29) gives

$$\sum_{r=0}^{\infty} [E_r Ce_r(a, q, u_0) ce_r(a, q, v) + F_r Se_r(a, q, u_0) se_r(a, q, v)] = 0. \tag{8.1.39}$$

As before, to satisfy this condition each term of the series must equal zero, giving

$$Ce_r(a_r, q_r, u_0) = 0 \tag{8.1.40}$$

$$Se_r(b_r, q_r, u_0) = 0, \tag{8.1.41}$$

where the resonance frequencies are obtained from the values of q which satisfy Eqs. (8.1.40) and (8.1.41). The complete set of resonance frequencies for the elliptic cylindrical tanker having rigid end caps and pressure release sidewalls is determined from Eq. (8.1.38) with $c = 58724$ in/s. Table 8-10 compares the first 24 resonance frequencies between the exact determination and the **Sierra/SD** prediction for this boundary condition configuration. Note that since $Ce_0 \neq 0$ the modes cut-on at a higher frequency compared to the rigid boundaries configuration.

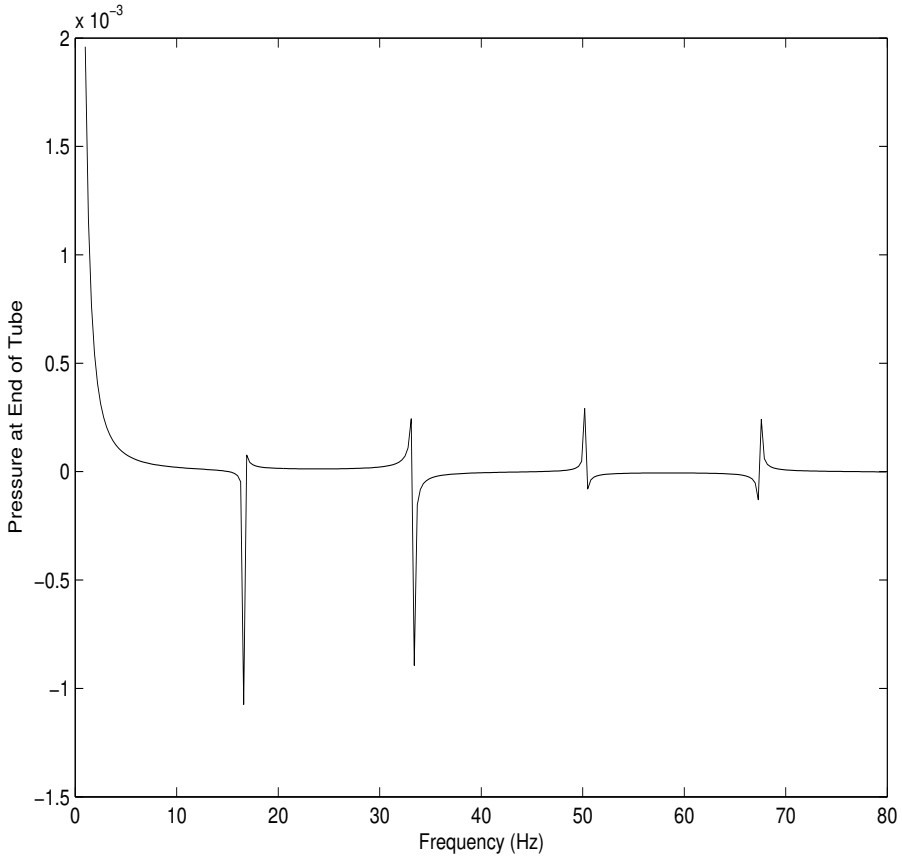


Figure 8-5. – Direct frequency response of an acoustical waveguide with rigid end cap.

Exact (Hz)	**Sierra/SD**	Percent Error
733.807	733.811	5.45e-4
774.849	774.853	5.16e-4
886.647	886.657	1.13e-3
970.884	970.898	1.44e-3
1002.26	1002.28	2.00e-3
1046.77	1046.8	2.86e-3
1224.69	1224.75	4.90e-3
1225.4	1225.45	4.08e-3
1236.59	1236.65	4.85e-3
1250.41	1250.47	4.80e-3
1322.61	1322.68	5.29e-3
1332.8	1332.89	6.75e-3
1355.83	1355.92	6.64e-3
1390.43	1390.53	7.19e-3
1422.68	1422.81	9.14e-3
1434.88	1434.99	7.67e-3
1444.44	1444.57	9.00e-3
1491.07	1491.19	8.05e-3
1511.69	1511.82	8.60e-3
1527.61	1527.8	1.24e-2
1550.06	1550.23	1.10e-2
1569.9	1570.08	1.15e-2
1571.93	1572.09	1.02e-2
1578.15	1578.34	1.20e-2

Table 8-10. – Comparison between the exact analytical resonance frequencies and **Sierra/SD** predictions for the elliptic cylindrical tank having rigid end caps and pressure release boundary conditions on the sidewall.

8.1.4. Direct Frequency Response

Next direct frequency response is determined for the previous configuration. The boundary condition is either the rigidly capped configuration of the previous example (a Neumann boundary condition), or a pressure release condition (a Dirichlet condition). For the two types of boundary conditions on the right end,[39] gives the exact resonance frequencies. When the tube is rigidly capped, they are

$$f_n = \frac{nc}{2L} \quad n = 0, 1, 2, 3, \ldots \tag{8.1.42}$$

and when the tube is open (pressure release) they are

$$f_n = \frac{(n + \frac{1}{2})c}{2L} \quad n = 0, 1, 2, \ldots \tag{8.1.43}$$

where f_n is in Hz, c is the speed of sound, and L is the length of the tube. In this example, $c = 332.0m/s$, and $L = 10.0m$, which results in the frequencies

$$f_n = 0.0, 16.6, 33.2, 49.8, \ldots \tag{8.1.44}$$

and

$$f_n = 8.3, 24.9, 41.5, \ldots \tag{8.1.45}$$

Figures 8-5 and 8-6 show the direct frequency response computations, and it is seen that the peaks in these plots correspond to the natural frequencies given above, for both types of boundary conditions.

The pressure at the piston, as a function of frequency, is given in[32] as

$$p = -j\rho c V_0 \cot(kL).$$

In Figure 8-7, we plot the computed and exact pressure at the piston, as a function of frequency. The two curves are virtually identical, except at the point of resonance. At resonance, however, the computed solutions are known to be inaccurate, and thus some difference there is expected.

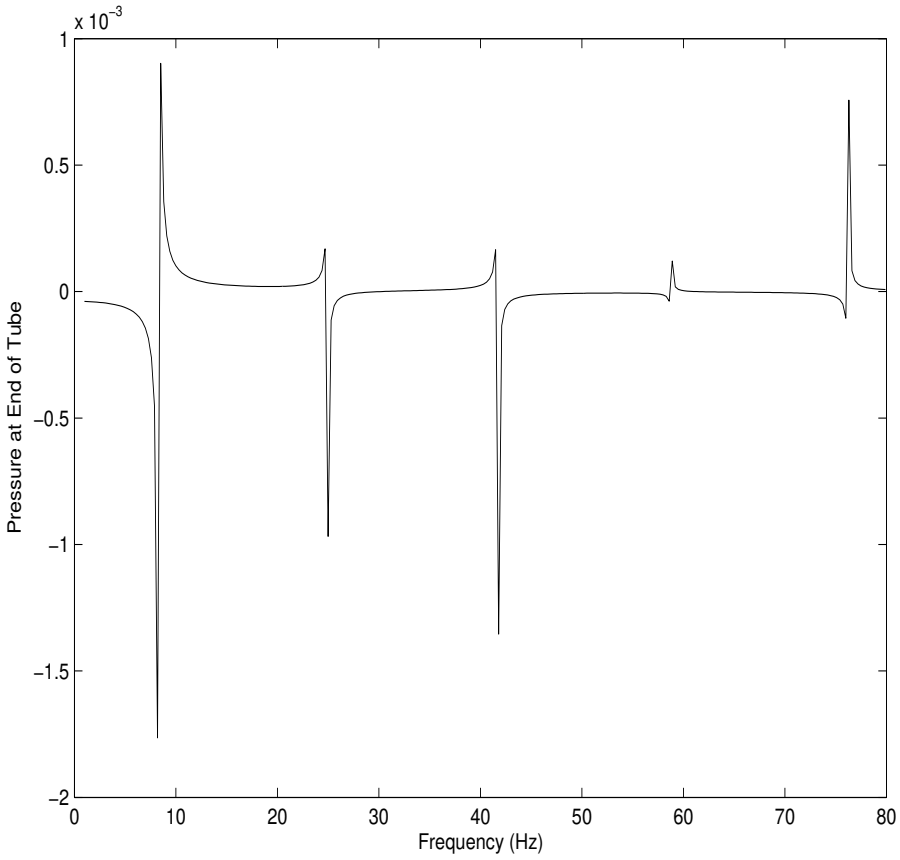


Figure 8-6. – Direct frequency response of an acoustical waveguide with pressure release end.

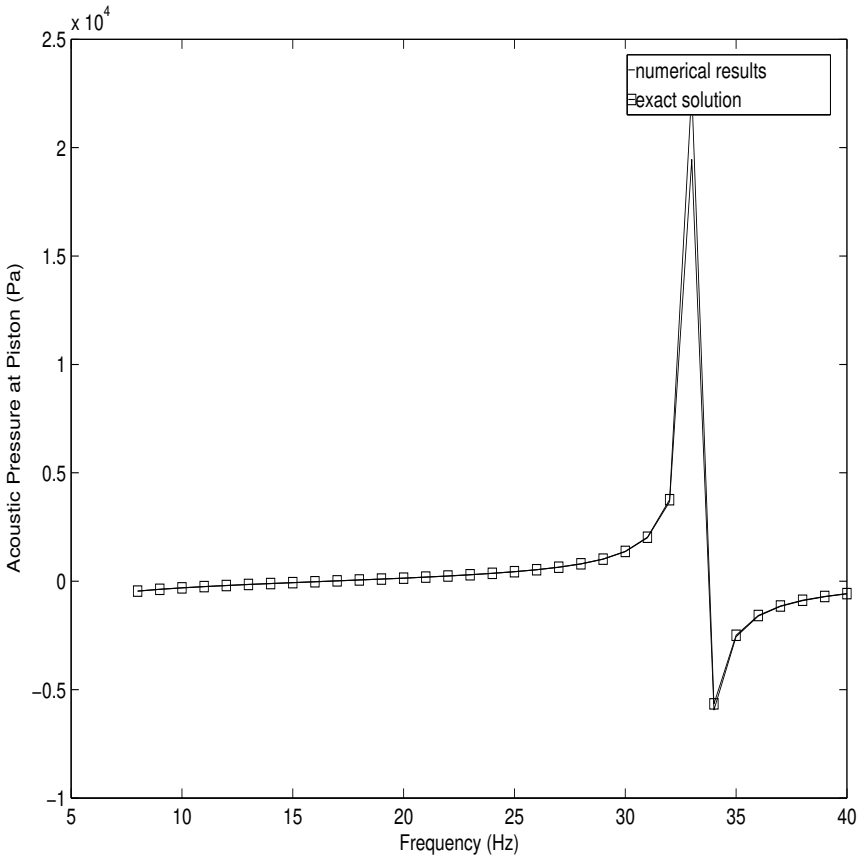


Figure 8-7. – Direct frequency response of an acoustical waveguide with rigid end cap. A comparison of computed and exact acoustic pressure at the piston.

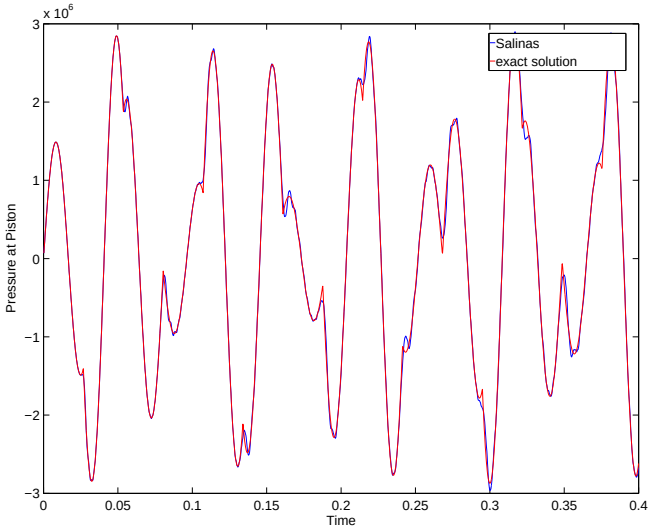


Figure 8-8. – Transient simulation of an acoustical waveguide with pressure release end condition.

8.1.5. Transient Acoustics with Pressure Release

This example was similar to the previous case, except that the far end of the tube was assigned a pressure release boundary condition. Also, in this case the velocity of the piston was assigned as

$$v(0,t) = v_p(t) = \sin(\omega t) \tag{8.1.46}$$

where $\omega = 60\pi$. The exact solution is given in[39] as

$$p(0,t) = \rho c \left[v_p(t) + 2 \sum_{n=1}^{\infty} (-1)^n v_p(t - \frac{2nL}{c}) \right]. \tag{8.1.47}$$

The terms in the summation become nonzero if their arguments are positive. This behavior was implemented in MATLAB using Heaviside functions, and the results were compared with **Sierra/SD**. Figure 8-8 shows the results. Excellent agreement between exact and computed solutions is observed.

8.1.6. Nonconforming Acoustic-Acoustic Discretizations

In this example, we test our simple method for coupling two acoustic domains that have mismatched meshes on the interface between them. In this case we chose an acoustic eigenvalue analysis, since the resulting eigen frequencies can be conveniently used in a convergence analysis. A three-dimensional example consisting of two adjacent acoustic domains with different discretization densities was investigated, as shown in Fig. (8-9). The nearly cubic volume having dimensions $L_x = 5$ m., $L_y = 10\sqrt{2}/3$ m., and $L_z = 15/(2\sqrt{2})$ m. was used to avoid repeated eigenvalues. The model was divided in half by an xy-plane located at $L_z/2$, as shown in Fig. (8-9), and the two halves were connected together using the inconsistent tied contact approach described in the previous section. This configuration was chosen to investigate the convergence of inconsistent tied contact for mode shapes having pressure variations in the plane of the interface. The fluid in both regions had sound speed $c = 343$ m/s and fluid density $\rho = 1.20$ kg/m^3. The boundary condition is a rigid wall (Neumann). The equations in subsubsection Coupled Equations and Their Discretizations, subsection Coupled Structural Acoustics, section Acoustics and Structural Acoustics[44] were solved with zero forcing on the right-hand side, thus corresponding to the acoustic eigenvalue problem with mismatched meshes on subdomains.

Four element size ratios at the interface were investigated: 2:3, 2:4, 3:4, and 4:5. Problems with convergence can arise in inconsistent tied contact when the face-surface is more finely discretized than the node-surface, see for example.[17,18] In all cases, the face-surface was chosen as the side with the coarser discretization. The convergence study consisted of uniformly refining the meshes several times, while keeping these discretization ratios (and hence element size ratios) at the nonconforming interface fixed. Only linear hexahedra were considered. The eigenvalues of the first thirty modes in the model were compared to the theoretical eigenvalues given by

$$f = \frac{c}{2} \sqrt{\frac{N_x^2}{L_x^2} + \frac{N_y^2}{L_y^2} + \frac{N_z^2}{L_z^2}}, \tag{8.1.48}$$

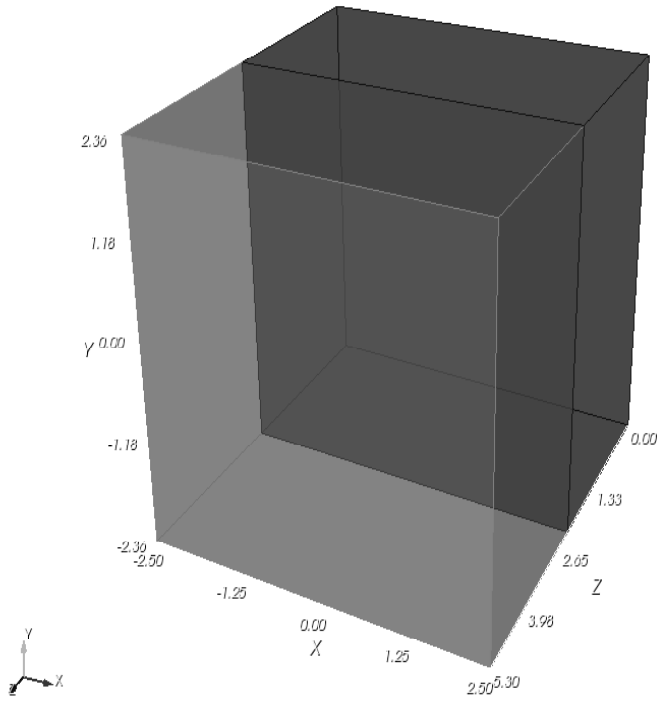


Figure 8-9. – Three-dimensional model.

where N_x, N_y, and N_z are non-negative integers. For comparison of the convergence rates, the eigenvalues of a conforming model were also obtained. In Figs. (8-10)-(8-12) the convergence plot for the four discretization ratios are shown along with the conforming case. The horizontal axis is the common logarithm of the largest dimension of the face-surface side elements. The eigenvalue error is given by $100(\lambda^h - \lambda)/\lambda$. Figures (8-10), (8-11), and (8-12) illustrate convergence for an axial, tangential and oblique mode, respectively. For the conforming case, theory predicts that the eigenvalues will converge at a rate of 2.0 for linear elements. For comparison purposes, an additional line with a slope of 2.0 is added to the three previous figures, using the triangle symbol. For all the cases presented, the convergence rates for the nonconforming meshes are close to those of the conforming meshes. The exceptional the 2:3 case, in which the nonconforming meshes convergence rate is greater than 2, is believed to be an abnormality. The theoretical convergence rate of 2.0 is based on conforming theory, and thus does not apply in the nonconforming case.

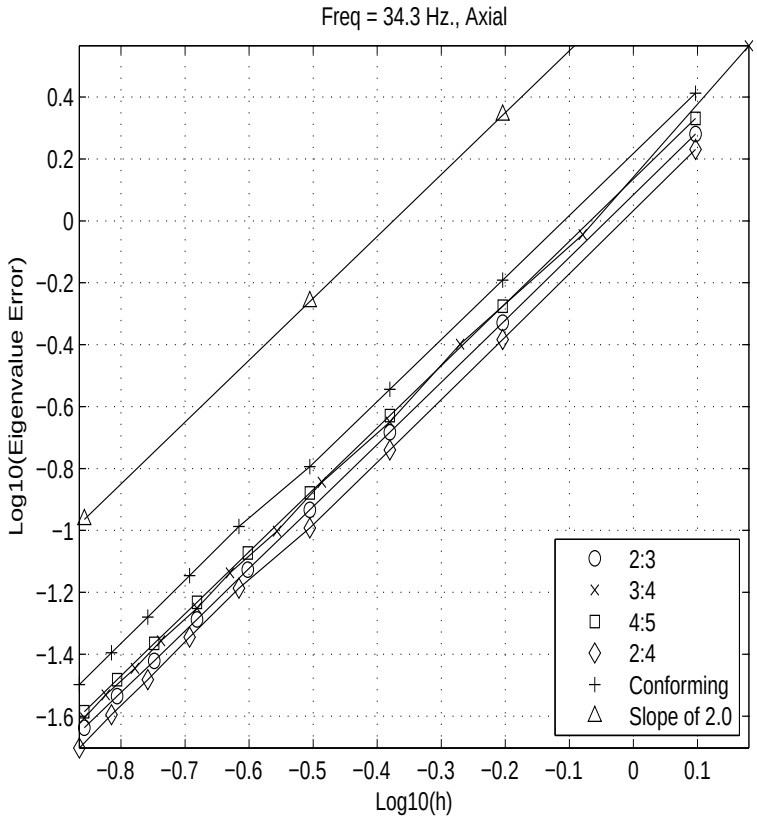


Figure 8-10. – Convergence plot for an axial mode ($N_x = 1, N_y = N_z = 0$).

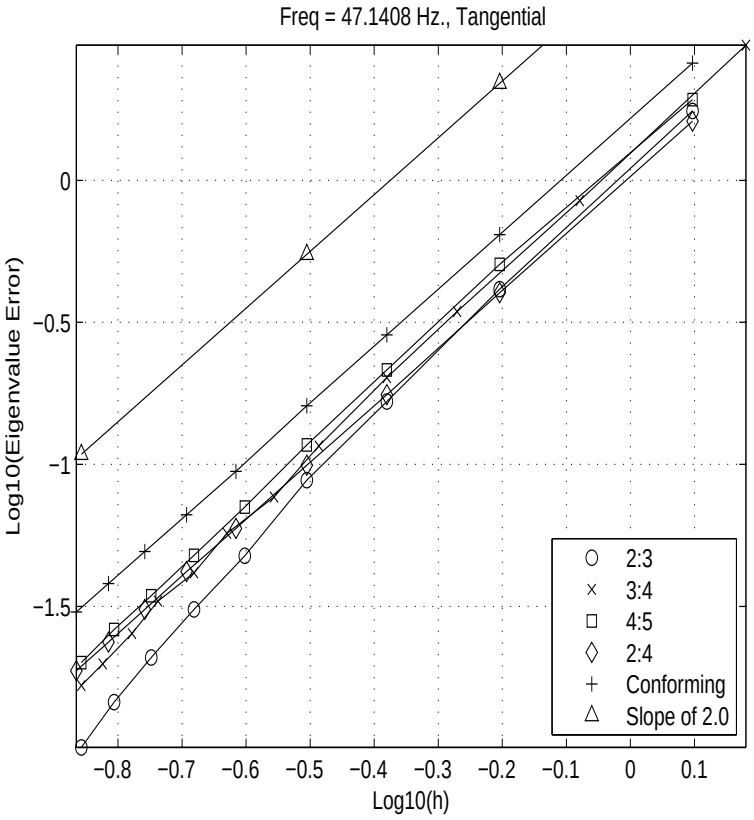


Figure 8-11. – Convergence plot for a tangential mode ($N_x = 1, N_y = 0, N_z = 1$).

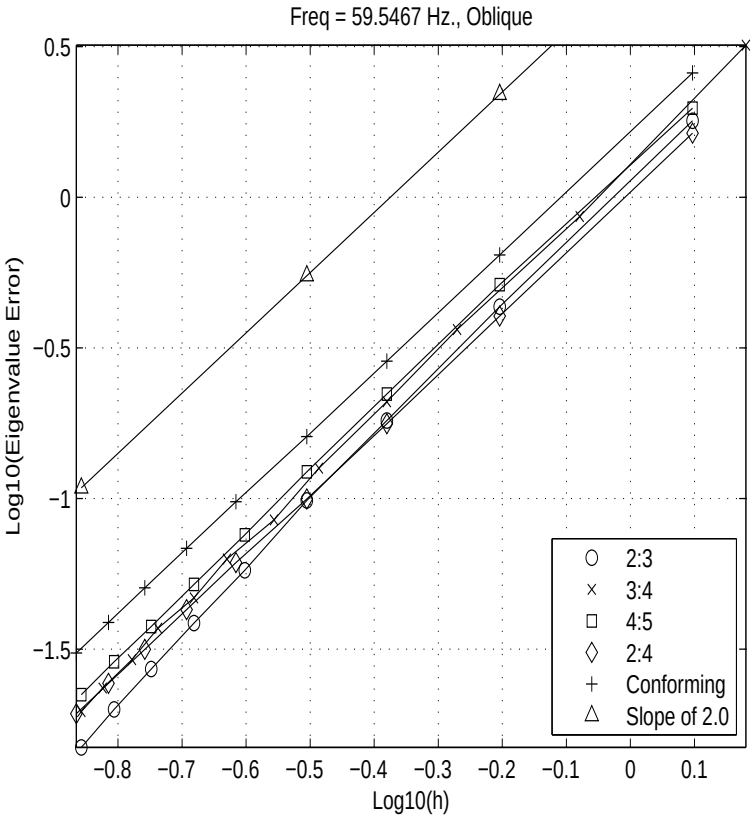


Figure 8-12. – Convergence plot for an oblique mode ($N_x = N_y = N_z = 1$).

8.1.7. Direct Frequency Response of Structural Acoustics with Tied Surfaces

This section written by Jerry Rouse.
In this case, the acoustic pressure and structural particle displacement of a one dimensional structural acoustic model is compared with theory. The model consists of a waveguide of square cross-section, 0.25 meters on a side, having an overall length of 20 meters. The length is equally divided between fluid and structure, each of length 10 meters. To the free end of the fluid is applied a harmonic particle velocity (forcing), and the free end of the structure is fixed. Inconsistent tied contact is used at the solid-fluid interface, where the fluid is treated as the independent surface. A **Sierra/SD** direct frequency response solution is used. The **Sierra/SD** prediction was verified at the tied interface between the fluid and solid regions.

System response was modelled using the wave equation for longitudinal wave propagation in the solid and acoustic wave propagation in the fluid. The two solutions were coupled at the solid-fluid interface through the continuity of elastic stress and pressure, and the continuity of structural particle displacement and acoustic particle displacement. The longitudinal wave equation for the solid is given by,

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c_s^2}\frac{\partial^2 u}{\partial t^2} = 0, \tag{8.1.49}$$

where u is the particle displacement, the phase velocity $c_s = \sqrt{\frac{E}{\rho_s}}$, E is Young's modulus, and ρ_s is the material density. For the solid the coordinate system was aligned such that the x_s-axis was the center of the waveguide, with $x_s = 0$ at the fixed end of the solid and $x_s = -L_s$ at the solid-fluid interface. The fixed end boundary condition for the solid is expressed $u(x_s = 0, t) = 0$. Application of this boundary condition to the general solution of Eq. (8.1.49), expressed in terms of left and right traveling waves, gives

$$u = A \sin(k_s x_s) e^{i\omega t}, \tag{8.1.50}$$

where the wave number $k_s = \omega / c_s$, $i = \sqrt{-1}$ and A is a frequency dependent coefficient which shall be determined from the continuity conditions at the solid-fluid interface.

The acoustic wave equation is given by

$$\frac{\partial^2 p}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0, \tag{8.1.51}$$

where p is the acoustic pressure, the phase velocity $c = \sqrt{\frac{\gamma P_0}{\rho_0}}$, where P_0 and ρ_0 are the undisturbed atmospheric pressure and density, respectively, and γ is the ratio of specific heats, here equal to 1.4. The coordinate system for the fluid was aligned such that the x_a-axis was the center of the waveguide, with $x_f = 0$ at the forcing end of the fluid and $x_f = L_f$ at the solid-fluid interface. The forcing boundary condition at the free end of the fluid in terms of the applied particle velocity V_0 is expressed

$$V_0 = \frac{i}{\omega \rho_0}\frac{\partial p}{\partial x}\bigg|_{x_f = 0}. \tag{8.1.52}$$

Application of this boundary condition to the general solution of Eq. (8.1.51) gives

$$p = \left[\frac{V_0 \, \omega \, \rho_0}{k_f} e^{-ik_f x_f} + B \cos(k_f \, x_f) \right] e^{i\omega t}, \tag{8.1.53}$$

where the wave number $k_f = \omega/c$ and B is a frequency dependent coefficient which shall be determined from the continuity conditions at the solid-fluid interface.

The coupling conditions at the solid-fluid interface ensure no net pressure and no net velocity across the interface. The continuity condition on pressure is given by

$$E \frac{\partial u}{\partial x}\bigg|_{x_s=-L_s} = -p\bigg|_{x_f=L_f}, \tag{8.1.54}$$

where tensile stress in the solid is considered positive, and the continuity condition on velocity is given by

$$\frac{\partial u}{\partial t}\bigg|_{x_s=-L_s} = \frac{i}{\omega\rho_0} \frac{\partial p}{\partial x}\bigg|_{x_f=L_f}. \tag{8.1.55}$$

Substitution of Eqs. (8.1.50) and (8.1.53) into Eqs. (8.1.54) and (8.1.55), and solving for the frequency dependent coefficients A and B finds

$$A = \frac{iV_0\omega\rho_0}{\omega^2\rho_0 \sin(k_s L_s)\cos(k_f L_f) + E k_s k_f \cos(k_s L_s)\sin(k_f L_f)}, \tag{8.1.56}$$

and

$$B = \frac{-V_0 c \rho_0 \sin(k_f L_f) e^{-ik_f L_f} \left[\omega^2\rho_0 + iE k_s k_f \cot(k_s L_s)\right]}{\omega^2\rho_0 \cot(k_f L_f) + E k_s k_f \cot(k_s L_s)}. \tag{8.1.57}$$

Given these coefficients, the structural particle displacement is

$$u = \frac{iV_0\omega\rho_0 \sin(k_s x_s) e^{i\omega t}}{\omega^2\rho_0 \sin(k_s L_s)\cos(k_f L_f) + E k_s k_f \cos(k_s L_s)\sin(k_f L_f)}, \tag{8.1.58}$$

and the acoustic pressure given by

$$p = \frac{iV_0 c \rho_0 \sin(k_f L_f) e^{i\omega t} \left[\omega^2\rho_0 \sin(k_f(L_f - x_f)) - E k_s k_f \cot(k_s L_s)\cos(k_f(L_f - x_f))\right]}{\omega^2\rho_0 \cot(k_f L_f) + E k_s k_f \cot(k_s L_s)}. \tag{8.1.59}$$

The Sierra/SD verification was performed with the following properties for the system. The fluid was modeled as air: $c = 343$ m/s and $\rho_0 = 1.2$ kg/m^3. The solid was modeled as steel: $E = 200$ GPa., $\rho_s = 7850$ kg/m^3, and Poisson's ratio $\nu = 0$. The value of Poisson's ratio was intentional. In Figure 8-13 the **Sierra/SD** prediction of structural particle displacement at the solid-fluid interface is compared to the theoretical result given by Eq. (8.1.58) evaluated at $x_s = -L_s$. The **Sierra/SD** prediction was obtained over the frequency range 1 to 60 Hz. using a frequency step of 1 Hz. In Figure 8-14 the **Sierra/SD** prediction of acoustic pressure at the solid-fluid interface is compared to the theoretical result given by Eq. (8.1.59) evaluated at $x_f = L_f$. In both figures the **Sierra/SD** prediction shows excellent agreement with the theoretical result.

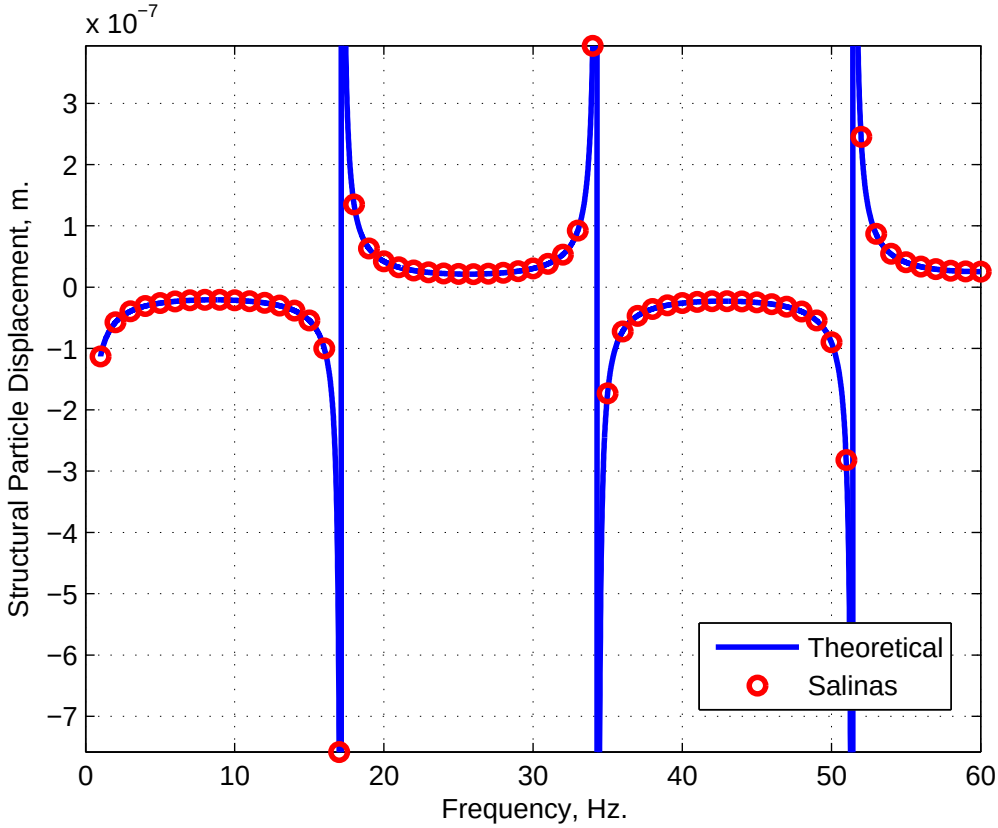


Figure 8-13. – Comparison of the **Sierra/SD** prediction of structural particle displacement at the solid-fluid interface with the theoretical result.

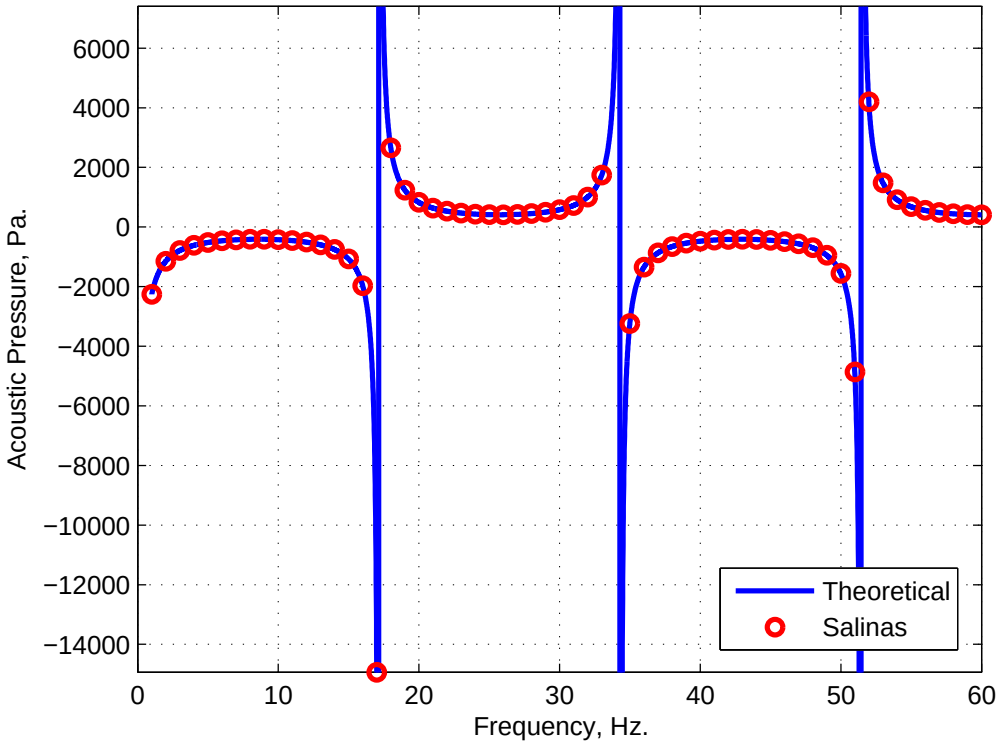


Figure 8-14. – Comparison of the **Sierra/SD** prediction of acoustic pressure at the solid-fluid interface with the theoretical result.

8.1.8. Radiation from a uniformly-driven spherical shell

In this example, we considered a spherical shell that was surrounded by an infinite acoustic fluid. The shell was composed of tria3 elements, and the acoustic fluid was modeled with tet4 elements. On the wet interface, the shell/acoustic meshes were conforming. The radius of the spherical shell was 1.0(m), and the radius of the truncated acoustic domain was 5.0(m). An absorbing boundary condition was applied to the exterior surface of the truncated acoustic domain, to simulate the infinite fluid.

A uniform, periodic pressure was applied to the inside surface of the spherical shell, and the resulting shell displacements and acoustic pressures were measured in the frequency domain. The analytic solution to this problem was derived in.[20] First we define some physical quantities. The impedance of the shell structure is given as

$$Z_s = \frac{i}{\omega}(\omega^2 m_s - k_s) \tag{8.1.60}$$

where $m_s = 4\pi a^2 h$, $k_s = \frac{8\pi E h}{1-\nu}$, h is the thickness of the shell, a is the radius of the shell, E is Young's modulus, and ν is Poisson's ratio. The impedance of the infinite fluid (as seen by the spherical surface that defines the shell) is

$$Z_f = \frac{i\omega\rho 4\pi a^3}{1+ika} \tag{8.1.61}$$

where $k = \frac{\omega}{c}$ is the wavenumber, ρ is the fluid density.

With the above quantities defined, the exact expression for the complex-valued radial displacement is

$$d = \frac{4\pi a^2 p_0}{i\omega(Z_s + Z_f}) \tag{8.1.62}$$

Figure 8-15 shows the comparison of the numerical results and analytic solution, for the real and imaginary components of radial displacement of the shell. The results show good agreement.

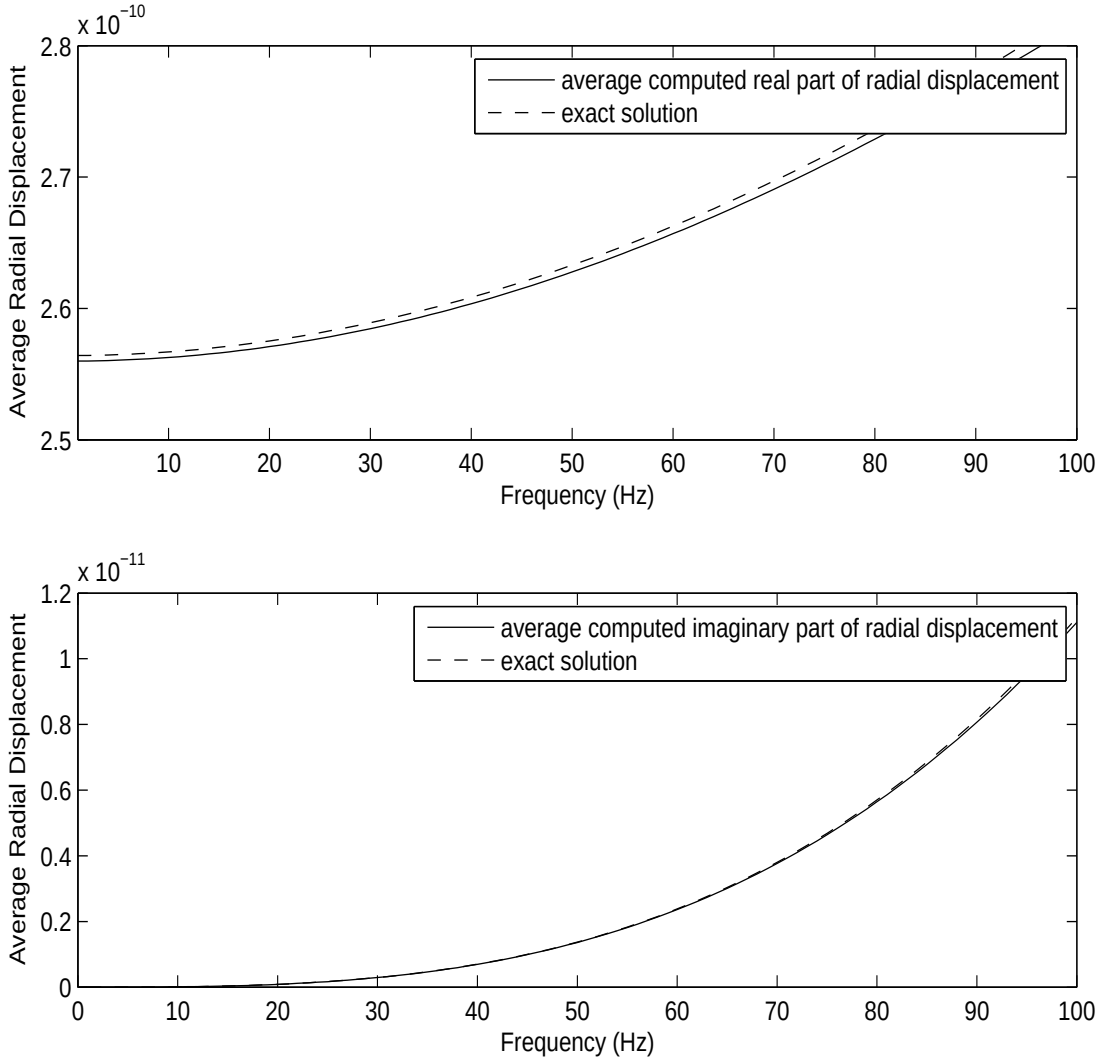


Figure 8-15. – Direct frequency response of a spherical shell immersed in an infinite fluid. The real and imaginary parts of the analytical solution are compared against **Sierra/SD**. The results show good agreement.

8.1.9. Radiation from a uniformly driven spherical acoustic surface

This example is similar to the previous example, except that the shell is removed, and we instead apply a uniform, periodic particle velocity to the inside surface of the spherical acoustic space. As in the previous example, an absorbing boundary condition is applied to the exterior surface of the truncated acoustic space, to simulate the infinite fluid. Once again, the radius of the inner spherical void is 1.0(m).

In this case, the analytic solution for the acoustic pressure on the driven surface is given by[39]

$$P = \frac{iv_0\omega\rho a^2}{r(1+ika)}e^{ik(r-a)} \tag{8.1.63}$$

where v_0 is the amplitude of the imposed particle velocity on the driven surface.

Figure 8-16 shows the comparison of the numerical results and analytic solution, for the real and imaginary components of the acoustic pressure. The results show good agreement.

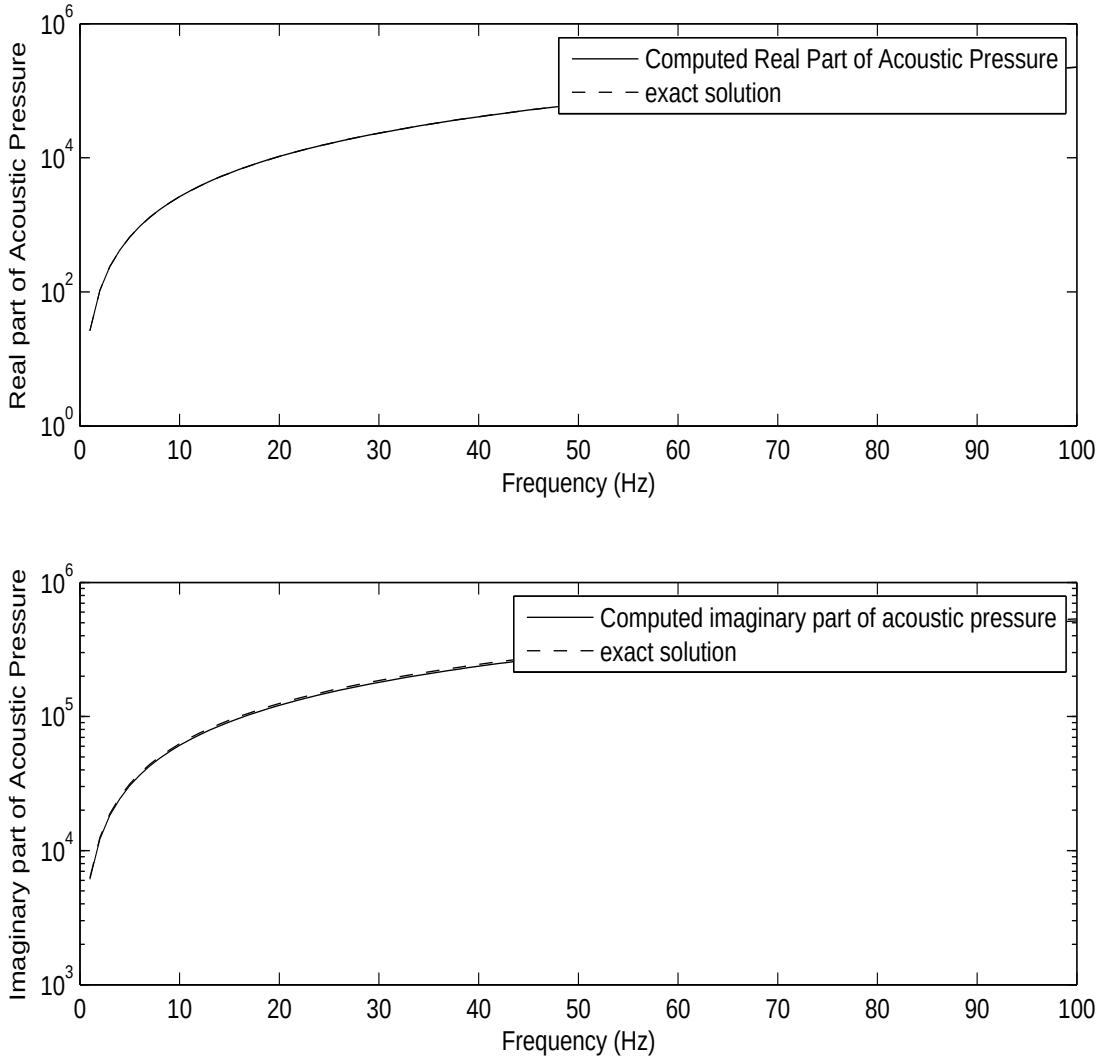


Figure 8-16. – Direct frequency response of a spherical shell immersed in an infinite fluid. The real and imaginary parts of the analytical solution are compared against **Sierra/SD**. The results show good agreement.

8.1.10. Scattering from a Flat Plate

This example involves scattering from a flat plate. The geometry consists of a uniform, acoustic tube of length 10(m), which is terminates by a flat plate. The acoustic tube is discretized with 3D acoustic elements, and the flat plate is discretized with quad shell elements. Plane waves are initiated inside of the acoustic tube, which then scatter off of the flat plate.

There is no analytical solution to this problem. However, we can still verify that the resonances of both the acoustic tube and the plate are excited at the correct excitation frequencies. This checks that the structural acoustic coupling between the plate and acoustic fluid is working correctly.

In the first example, we consider the fluid to be air, and the plate to be composed of steel, with a thickness of 0.1(m). In that case, the plate is a rigid surface to the fluid, and hence the resonance frequencies of the tube should match exactly that of a tube with rigid end caps. Figure 8-17 shows the acoustic pressure in the tube as a function of frequency. It is seen that the first resonance is predicted correctly, which according to theory should be 16.6Hz.

In the second example, we consider a light fluid that has a high speed of sound (ρ = 1.0, c = 1500.0). We also consider a thin plate, with thickness of 0.001(m). This lowers the natural frequencies of the plate well below those of the previous example. In this case, the fluid imparts no added mass effect onto the plate, since its density is low. Also, due to the high speed of sound, the natural frequencies of the tube are much higher than those of the plate. Consequently, the resonances of the plate should be the first observed resonances of the overall system. The first two exact resonances of the plate are at 3.5Hz, and 4.7Hz. Figure 8-18 shows the displacement of a corner point on the plate as a function frequency. The numerical results correctly predict the first two resonances of the plate.

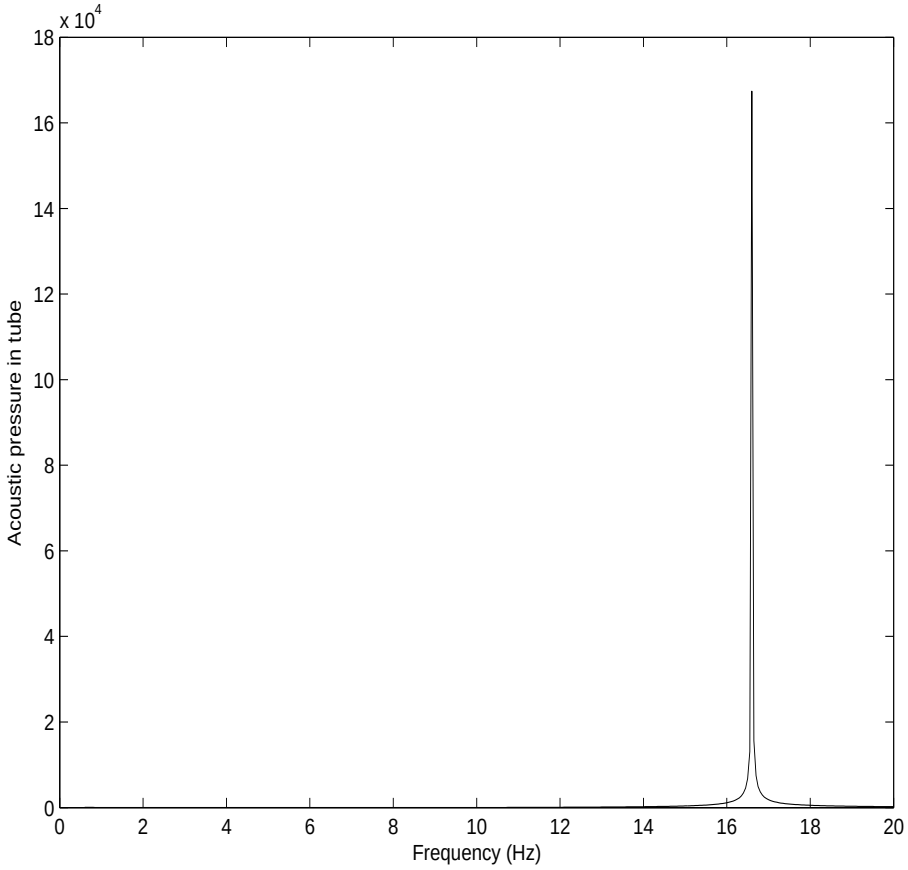


Figure 8-17. – Acoustic scattering from a plate. In the case when the plate is rigid compared with the fluid, the first resonance of the fluid tube, 16.6Hz, is reproduced well.

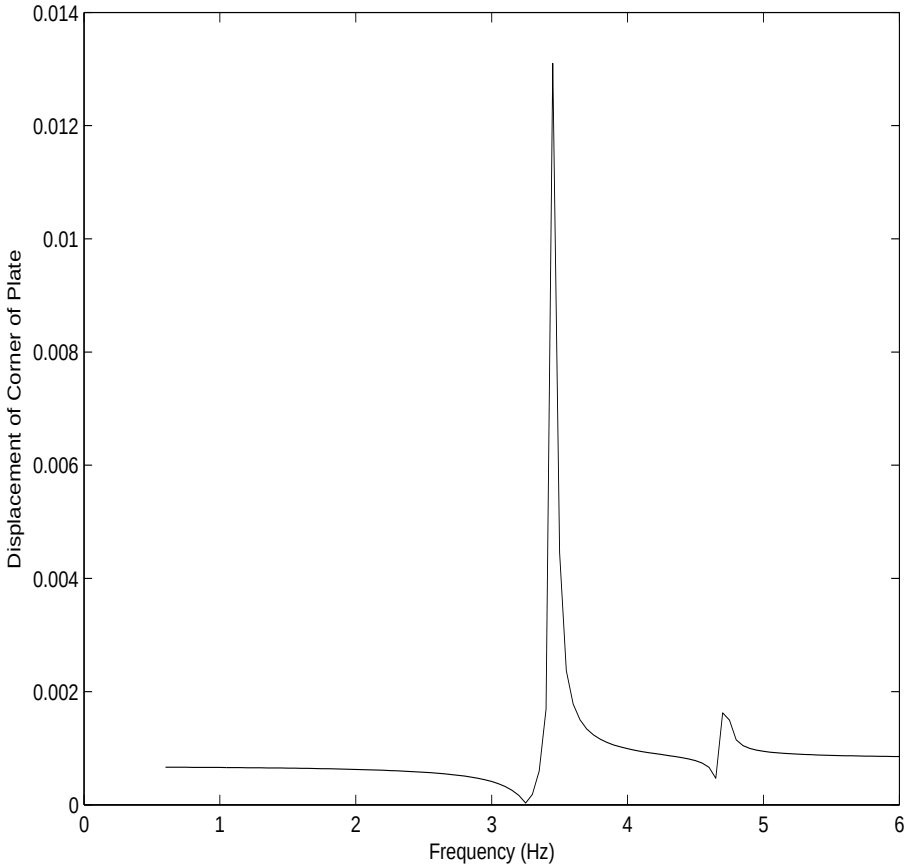


Figure 8-18. – Acoustic scattering from a plate. In the case when the fluid is given a low density and high speed of sound, the first resonance of the plate appears before the acoustic tube resonances. In that case, the first two resonances of the plate, 3.5Hz and 4.7Hz, are reproduced well.

8.1.11. Transient Scattering from a Flat Plate

In this example, we evaluate transient scattering from a flat plate. The test consists of an acoustic domain that is a perfect cube of dimensions $1 \times 1 \times 1$, which is attached with tied surfaces to a flat plate of dimension 1×1. The acoustic domain is given properties of air, and the flat plate is made of steel. Given the material property mismatch between the structural and acoustic domains, the coupling between these domains is negligible. This allows us to test the effect that the scattering waves have on the acoustic and structural components separately, without having to consider coupling.

The structural acoustic system is subjected a harmonic plane wave with frequency of 10Hz. The wet surface is located at the origin, and thus the incident pressure at the wet surface is given by

$$p(t) = \cos(\omega t) \tag{8.1.64}$$

The corresponding velocity input on the acoustic domain is given by

$$v(t) = \frac{1}{\rho c} \cos(\omega t) \tag{8.1.65}$$

An absorbing boundary condition is placed at the far-end of the acoustic domain, and thus the acoustic response should resemble that of an infinite tube. In that case, the acoustic pressure response should be equal to the input velocity times ρc. Figure 8-19 shows a comparison of the analytical and computed acoustic pressure on the wet surface. Excellent agreement is observed.

In the case of the structural response, we can use a simple force balance to determine the acceleration response of the plate, since we are ignoring coupling between the structural and acoustic components. In this case, the total pressure on the plate is equal to the sum of the incident and scattered pressures. The area of the plate is 1.0, and thus the force is equal to the pressure. Thus, we can compute the acceleration of the plate as follows

$$a = \frac{F}{m} = \frac{2\cos(\omega t)}{770} \tag{8.1.66}$$

Figure 8-20 shows the comparison of the analytical and computed acceleration of the plate.

This test case can be found at

`Salinas_rtest/verification/acoustic/hexplane.test`

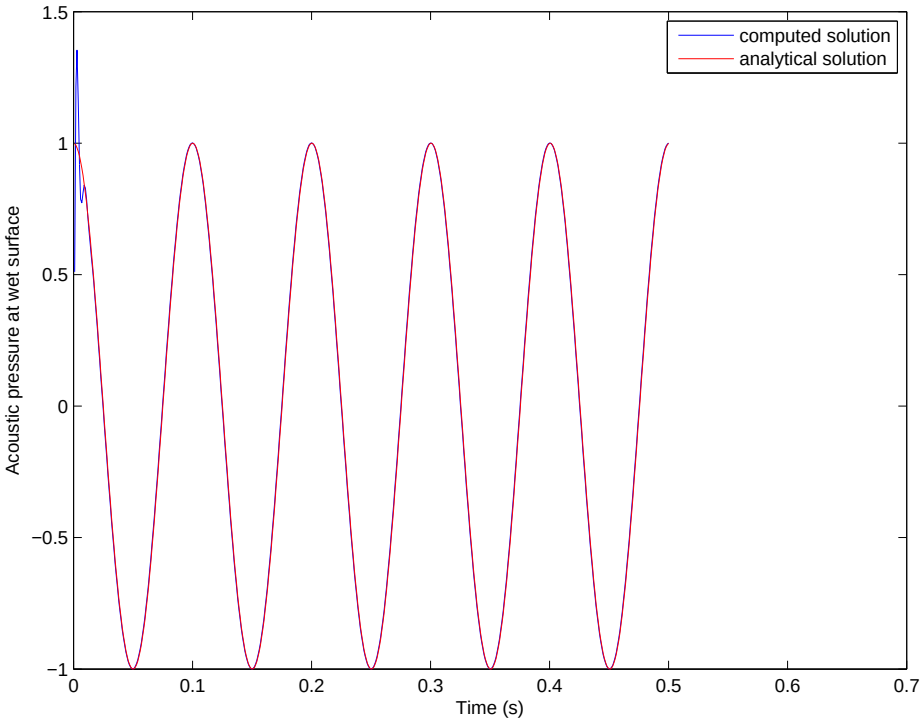


Figure 8-19. – Comparison of **Sierra/SD** result with analytical solution of the scattered acoustic pressure for a simple 1D problem.

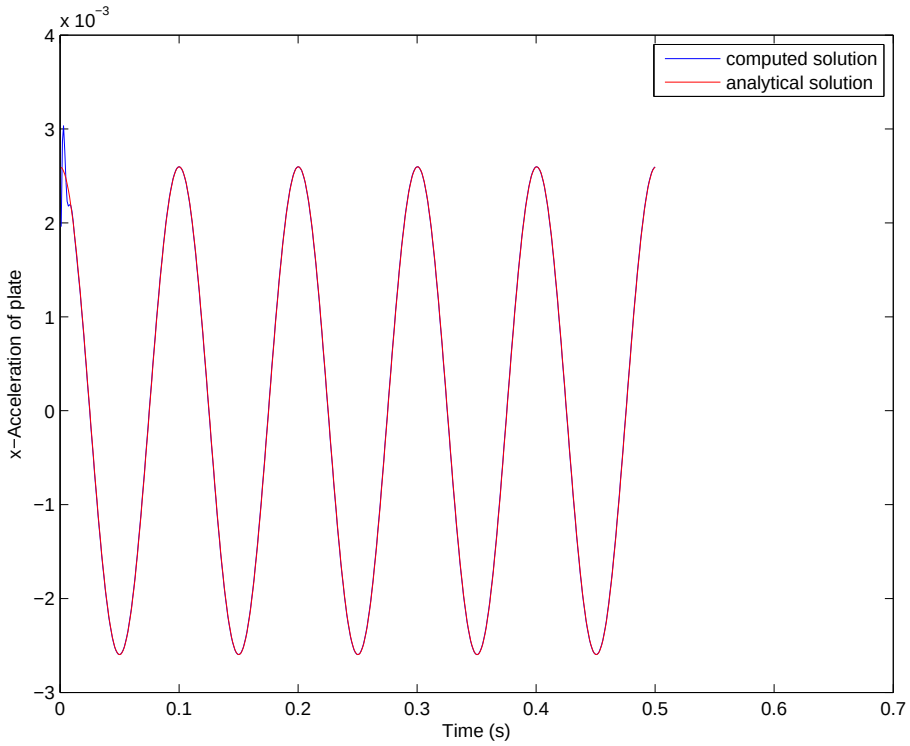


Figure 8-20. – Comparison of **Sierra/SD** result with analytical solution of the acceleration for a simple 1D scattering problem.

8.1.12. Transient Scattering of a Plane Step Wave from a Spherical Shell

Acoustic analysis often includes the concepts of a "scattering" solution. By this, we mean an analysis where it is easy to specify the incident wave at all points in space, and we solve for the reflected wave. Such scattering solutions are useful in a variety of contexts. For example, a submarine in the ocean may be struck by an incident "ping" from a neighboring ship. Such a ping is nearly a plane wave, and calculation of the outbound wave is the item of interest. Because the incident wave is known, we do not need to model the vast region of space between the incident source and the scattering object. This reduces the cost of the computation.

The theory manual details the formulation. Here we address verification of a simple sphere in an infinite medium using a problem from a LS-Dyna Verification Manual that is no longer available. The model includes a steel sphere of radius 10 inches and thickness 0.1 inches immersed in sea water. The parameters of the problem are given in Table 8-11.

parameter	value
shell radius	10.0 in
shell thickness	0.1 in
shell modulus	$0.29e+08\frac{lb}{in^2}$
shell density	$0.732e-03\frac{lb-sec^2}{in^4}$
water density	$0.96e-04\frac{lb-sec}{in^4}$
water speed of sound	$60000\frac{in}{sec}$
step wave amplitude	$100\frac{lb}{in^2}$
hit point	$z=-10in$

Table 8-11. – Parameters from Verification Model of Spherical Shell Subjected to Plane Step Wave

The solution is shown in Figure 8-21. There are discrepancies. The FEM solution excites higher order modes not seen in the analytic solution. There may be reflections from the boundaries of the fluid mesh. The nightly test (located in Salinas rtest/verification/acoustic/scattering/sphere plane step), is "quarter-sphere".

We note that the quarter-sphere model described utilized the standard absorbing boundary condition for the exterior surface of the acoustic mesh. Identical results are obtained using infinite elements. The location of the test is

```
Salinas_rtest/verification/acoustic/scattering/quarter_sphereIE.inp
```

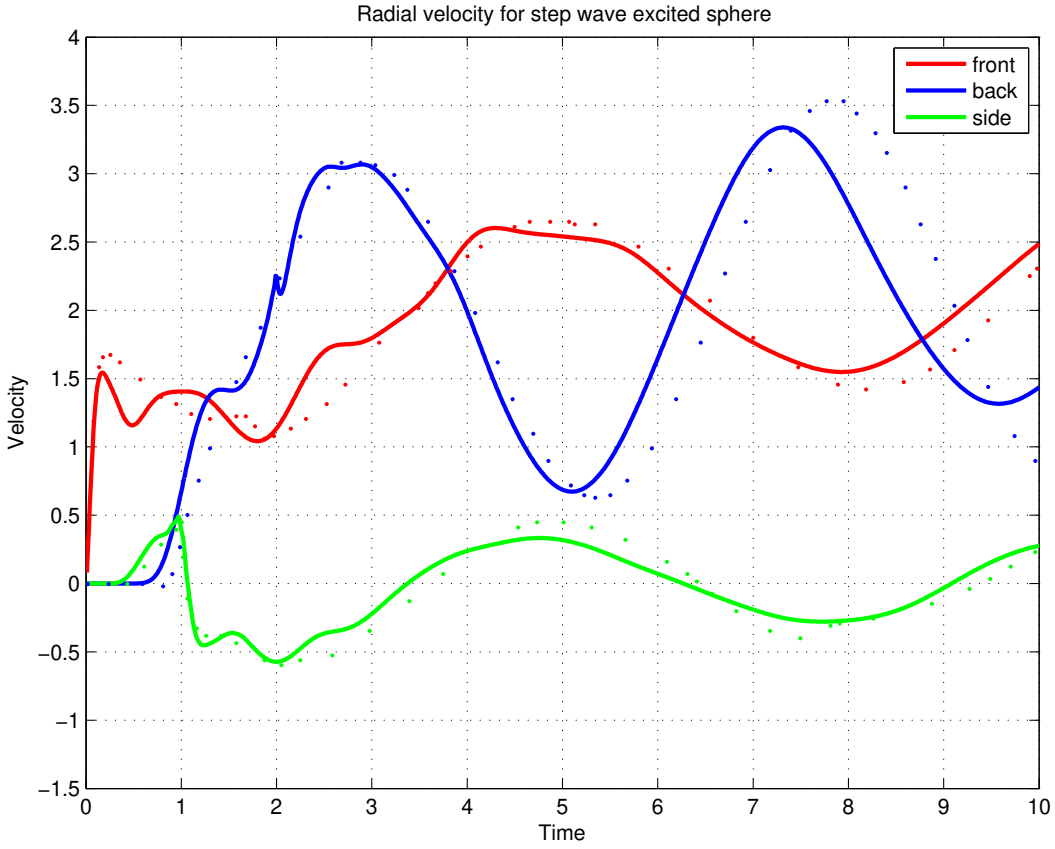


Figure 8-21. – Sphere Impacted by Step Wave. The incoming step wave arrives from the $-Z$ direction. Dashed lines are the analytic solution.

8.1.13. Infinite Elements on an Ellipsoidal Surface - Transient Scattering

It is often advantageous to mesh the area about a structure with an ellipsoidal (or prolate spheroid) mesh, and use infinite elements on the ellipsoidal boundary to model the effects of an infinite fluid. This is the case if a submarine is modeled. A spherical mesh about this long cylindrical structure is larger than an ellipsoidal mesh. To verify the behavior of the infinite elements on this boundary, we use the spherical structure of section 8.1.12 and compare with the known closed form solutions.[30] This problem came to our attention through an LS-Dyna Verification Manual that is no longer available.

The standard formulation of infinite elements is built on radial basis functions. In the case of a sphere, these basis functions can be defined using a common source location at the origin of the sphere. When the infinite element surface is an ellipsoid, a common source location yields basis functions that are not orthogonal to the infinite element surface, resulting in poor performance and spurious reflections. To alleviate this, the basis functions for an ellipsoidal can be defined using a variable source location, such that each element (each node on the surface) has its own source point for expansions of the basis functions. This ensures that the basis is orthogonal to the ellipsoidal surface.

To evaluate the reflection of the infinite elements, several meshes were composed. Details of the meshes are shown in Table 8-12. All meshes are quarter symmetry models. A representative mesh is shown in Figure 8-22. Results from the analyses are shown in Figures 8-23 through 8-25.

Name	Eccentricity	Acoustic Elements
sphere-m1	1:1	672
sphere-m2	1:1	5088
sphere-m3	1:1	40128
sphere-m4	1:1	323856
ellipse-m1	3:1	672
ellipse-m2	3:1	5088
ellipse-m3	3:1	40128
ellipse-m4	3:1	323856

Table 8-12. – Mesh Parameters of Infinite Elements on Ellipsoidal Surfaces.

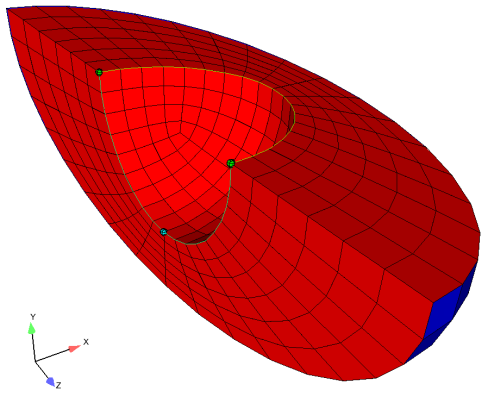


Figure 8-22. – Representative Mesh of Quarter Symmetry Sphere in Ellipse.

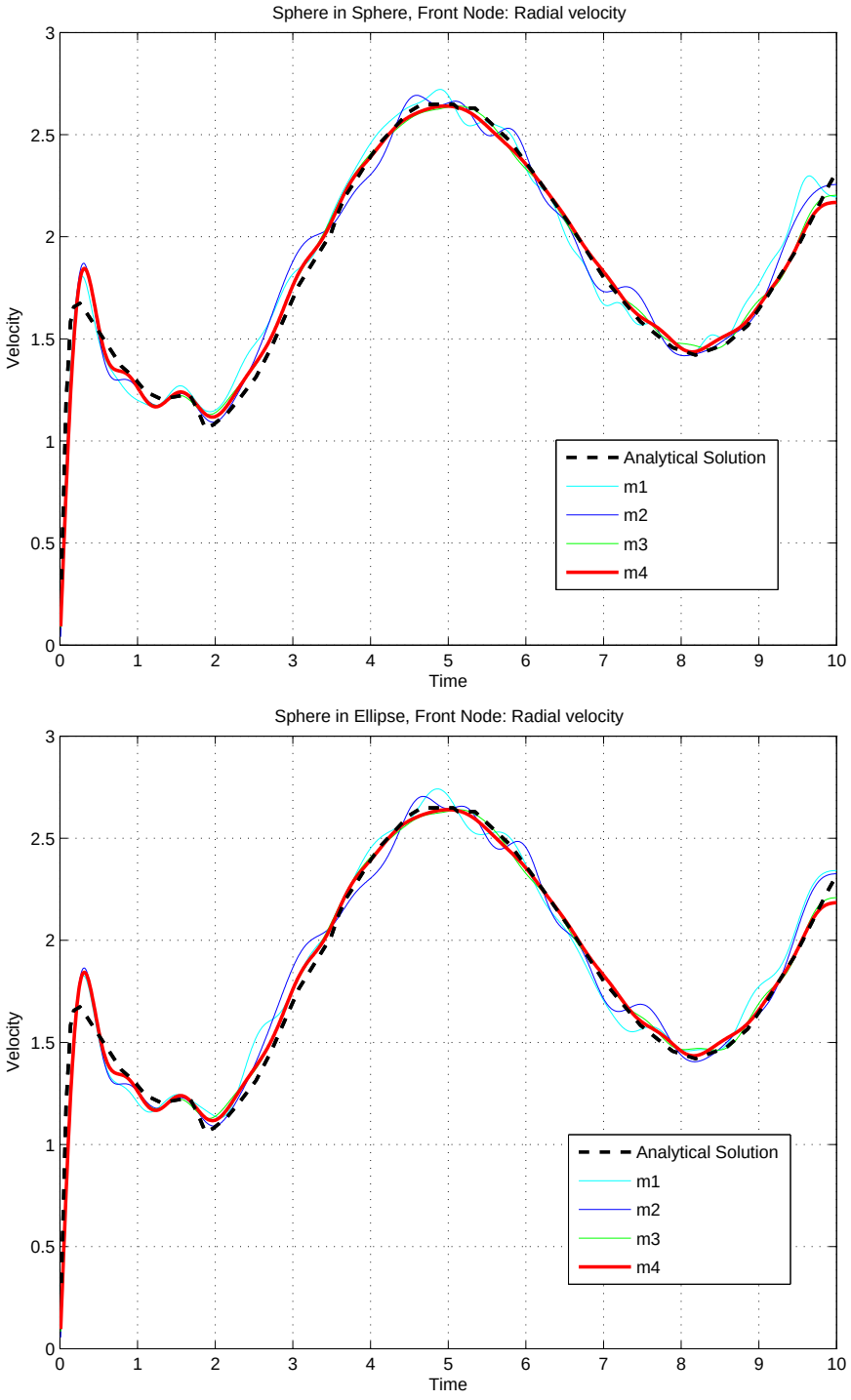


Figure 8-23. – Filtered Front Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

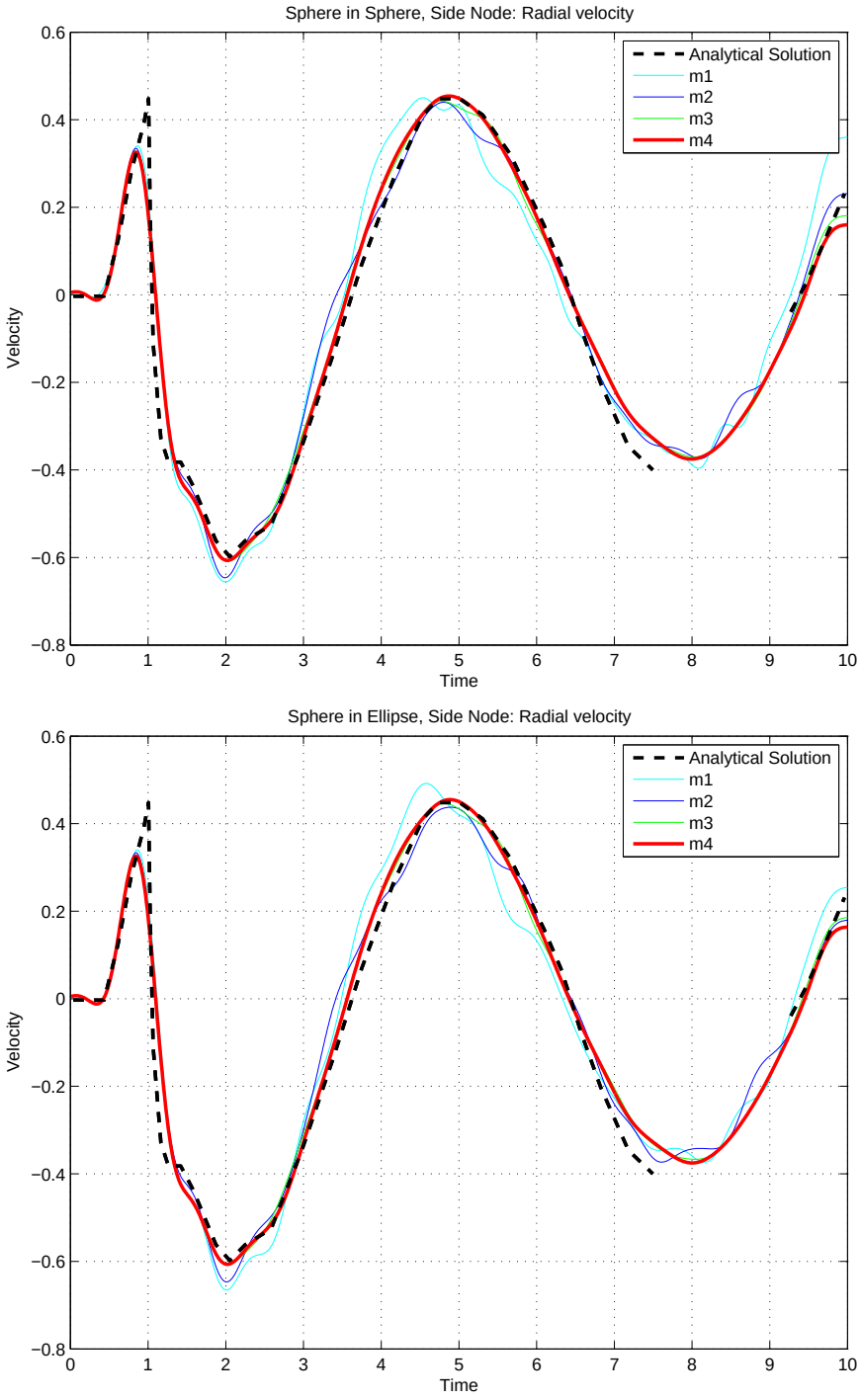


Figure 8-24. – Filtered Side Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

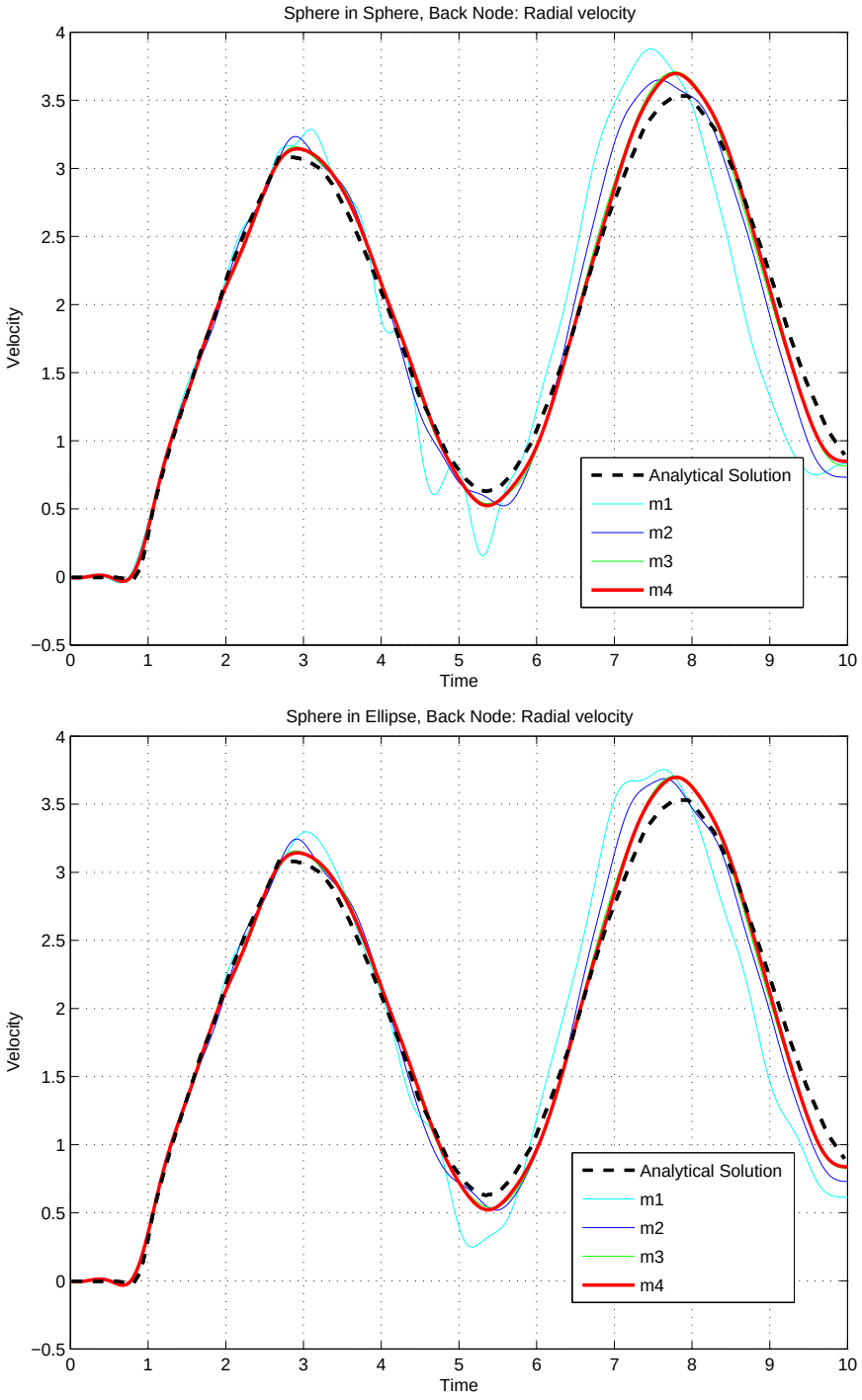


Figure 8-25. – Filtered Back Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

8.1.13.1. High Frequencies

There are two reasons why it is necessary to eliminate high frequencies from the comparison. First, the analytic solution is a series summation (see equation 17[30]). It contains the lowest frequency modes in the solution, and filters the higher frequency solution. Second, high frequencies are introduced through of the mesh discretization. We observe that while the frequency of these spurious solutions increases with mesh density, the amplitude typically decreases. It is impractical to refine the mesh sufficiently to eliminate all such mesh dependent responses.

The higher frequencies could be eliminated in a variety of ways. The input loading can be filtered to "smooth" the step function and eliminate high frequency excitation. The integrator could introduce artificial numerical damping which removes high frequency energy during the computation. Or, the signal could be post-processed by filtering. We use post-process filtering in this case because it is straightforward to implement and does not introduce unknown phase shifts. We use the MATLAB "filtfilt" function on a Butterworth low pass filter of order 6. The cutoff frequency is 10 kHz.

The radial response of an unfiltered and filtered responses is shown in Figure 8-26. Even with increasing mesh density, high frequency oscillations continue to dominate the response.

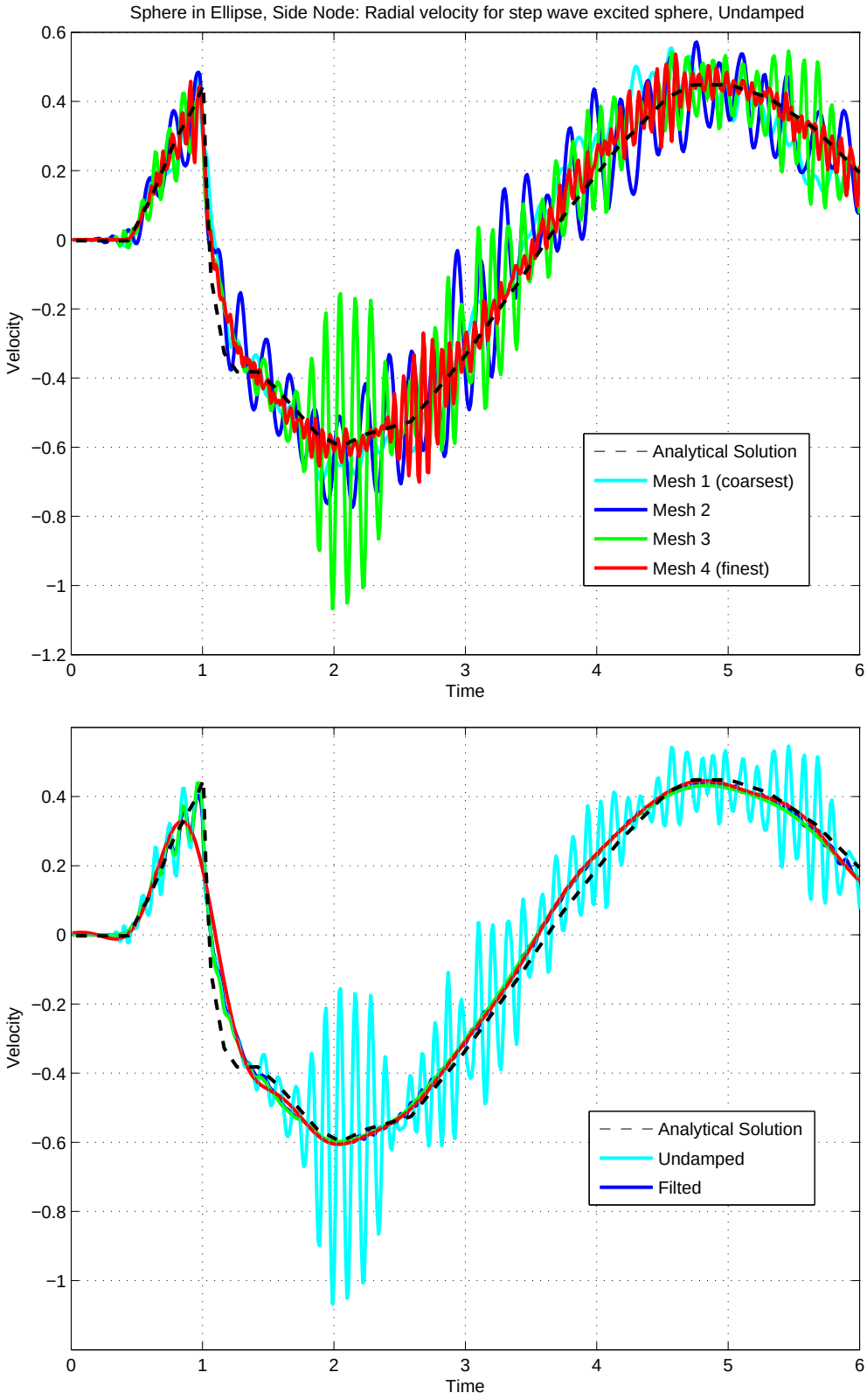


Figure 8-26. – Sphere in Ellipsoid. Unfiltered response at 90^o location.

8.1.13.2. Dependence on Loading Decay

The analytic solution loadings include an exponential decay following a step wave response. [1] The previous analysis was analysis performed with no decay. Figures 8-27 and 8-28 show the response for various decay factors as observed on the leading and trailing edges of the sphere. The analytical solutions for this case were taken from Sprague and Geers.[48]

The purposes of these plots is to determine the dependence of the solution on the decay parameter "beta". This dependence is in general well represented, but the phase error is significant. Figure 8-29 compares numeric solution with the analytic solution of Geers and the results published in the USA verification manual for the case of $\beta = 0$. The numeric results are much closer to the USA prediction. There are some issues here that have not been identified. The two analytical solutions should be identical, but differ. We can guess that a different number of terms were retained in the series expansion. The USA solution is available for $\beta = 0$ only.

[1] The pressure can be written as,

$$P = H(t - \tau) \exp(-\beta[t - \tau])$$

where $H()$ is the Heavyside step function, t is the measurement time, τ represents the travel time from the source to measurement location and β is the decay constant.

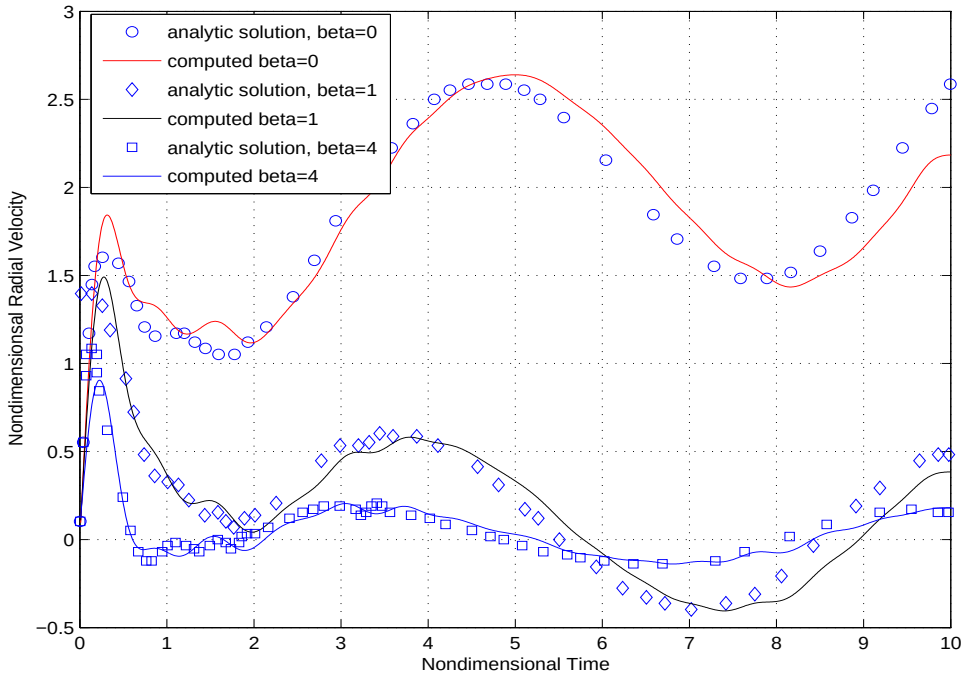


Figure 8-27. – Comparison of **Sierra/SD** result with analytical solution of the scattered acoustic pressure on the leading surface of a sphere. Mesh=m4.

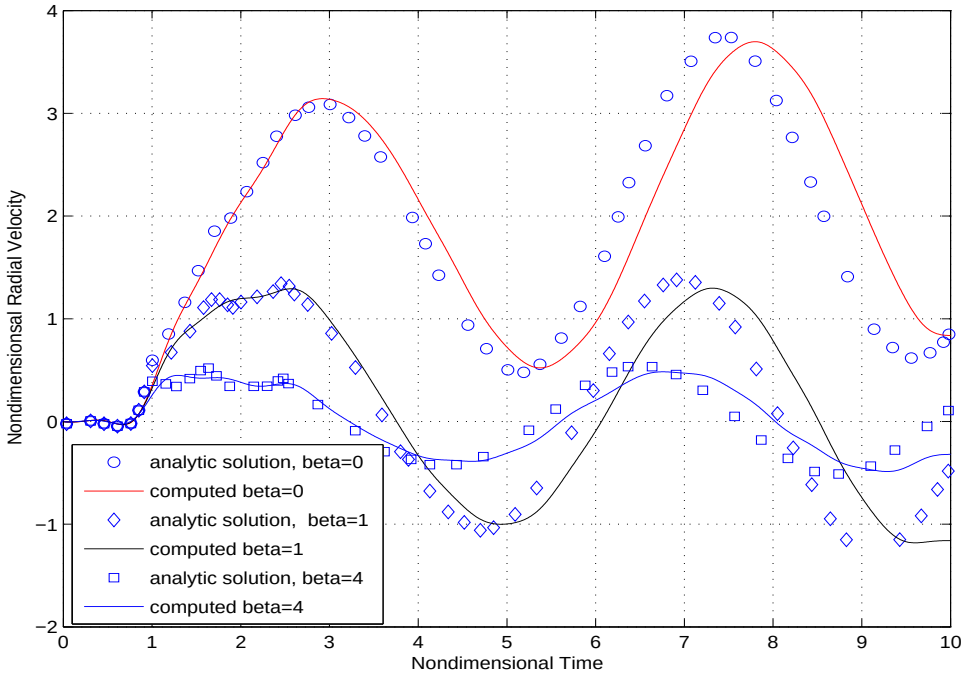


Figure 8-28. – Comparison of **Sierra/SD** result with analytical solution of the scattered acoustic pressure on the back surface of a sphere. Mesh=m4.

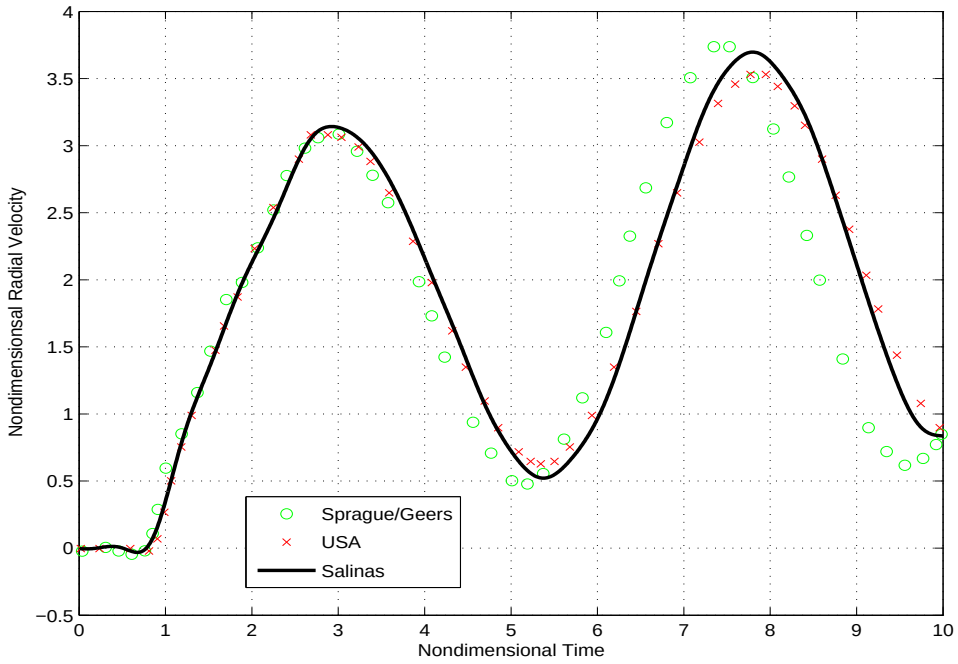


Figure 8-29. – Comparison of **Sierra/SD** numerical result with two differing analytic solutions. Mesh=m4. Prediction on the back surface

8.1.14. A comparison of spherical and ellipsoidal infinite elements on a model problem

In this section we examine the results of a simple test problem designed to compare the results of infinite elements on spherical and ellipsoidal meshes. For the purposes of these comparisons, we will use the results on the spherical meshes as the truth model, and the goal will be to show that for sufficiently fine acoustic meshes and sufficiently high infinite element order, the results on the spherical and ellipsoidal meshes are the same.

Figures 8-30 and 8-31 show the geometry of the test case. In the case of the ellipse, two different aspect ratios were studied, 10 : 1 and 3 : 1. Figure 8-31 shows the aspect ratio of 10 : 1. An acoustic mesh is defined on a spherical (Figure 8-30) and ellipsoidal (Figure 8-31) geometry. In both cases a cylindrical hole is cut out from the mesh, and an applied acoustic velocity is applied to the outermost surface of the cutout. The applied velocity is the same on the entire surface, and consists of the hat function shown in Figure 8-32.

Figure 8-33 shows the results of acoustic pressure along a 45^o angle relative to the major axis, for a spherical mesh and an ellipsoidal mesh of aspect ratio 3 : 1. For the ellipsoidal meshes, results are shown using two different source location algorithms of the plane-line intersect method, and the constant offset method. The results from a previous **Sierra/SD** release that involved a fixed source location is also shown. Both the plane-line intersect and constant offset ellipse algorithms replicate the results produced on the sphere, but the fixed source location algorithm from the previous **Sierra/SD** release shows significant differences. This is expected, since that algorithm

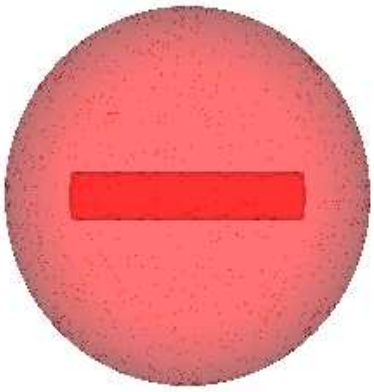


Figure 8-30. – Spherical acoustic mesh for cylindrical cutout problem.

Figure 8-31. – Ellipsoidal mesh with aspect ratio 10:1 for cylindrical cutout problem.

required a zero mass matrix even when the mass matrix was non-zero, as in this case. Figure 8-34 shows the same results, but for an ellipsoidal mesh of aspect ratio 10 : 1. Similarly, the plane-line intersect and constant offset source location algorithms for the ellipsoidal meshes yield identical results to the sphere.

Figure 8-35 shows the results of acoustic pressure along the major axis, for a spherical mesh and an ellipsoidal mesh of aspect ratio 3 : 1. For the ellipsoidal meshes, results are shown using the two different source location algorithms of the plane-line intersect method, and the constant offset method. The results involving a fixed source location that was implemented in a previous **Sierra/SD** release are also shown. Both the plane-line intersect and constant offset ellipse algorithms replicate the results produced on the sphere, but the fixed source location algorithm shows significant differences. This is expected, since that algorithm required a zero mass matrix even when the mass matrix was non-zero, as in this case. Figure 8-36 shows the same results, but for an ellipsoidal mesh of aspect ratio 10 : 1. In this case, the initial behavior of the results on ellipsoidal meshes are identical to that of the sphere, but later times show some small discrepancies. Further increases in infinite element order did not resolve these discrepancies. Additional acoustic mesh refinements are necessary for the results to converge.

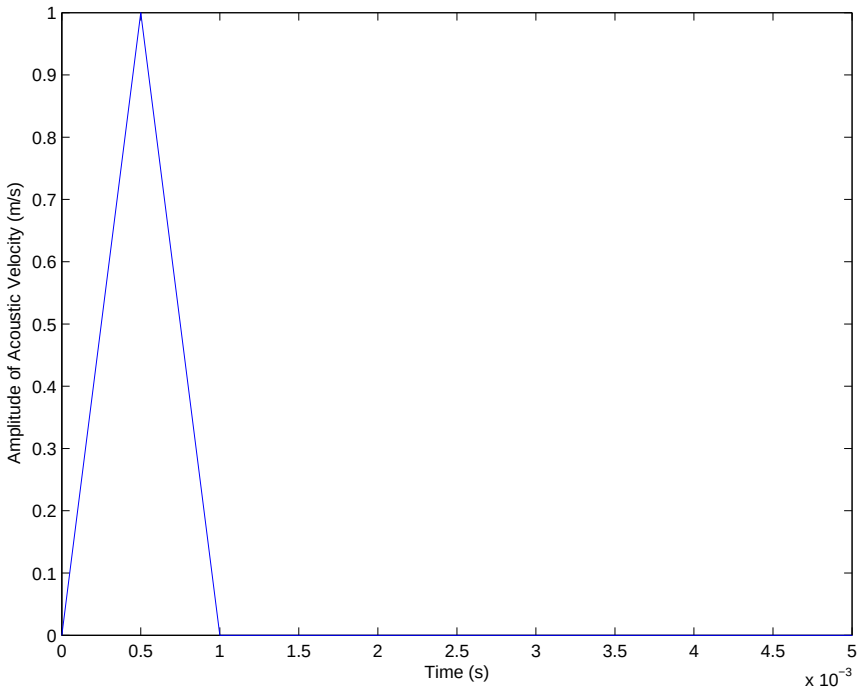


Figure 8-32. – Amplitude function used to scale input acoustic velocity for cylindrical cutout problem.

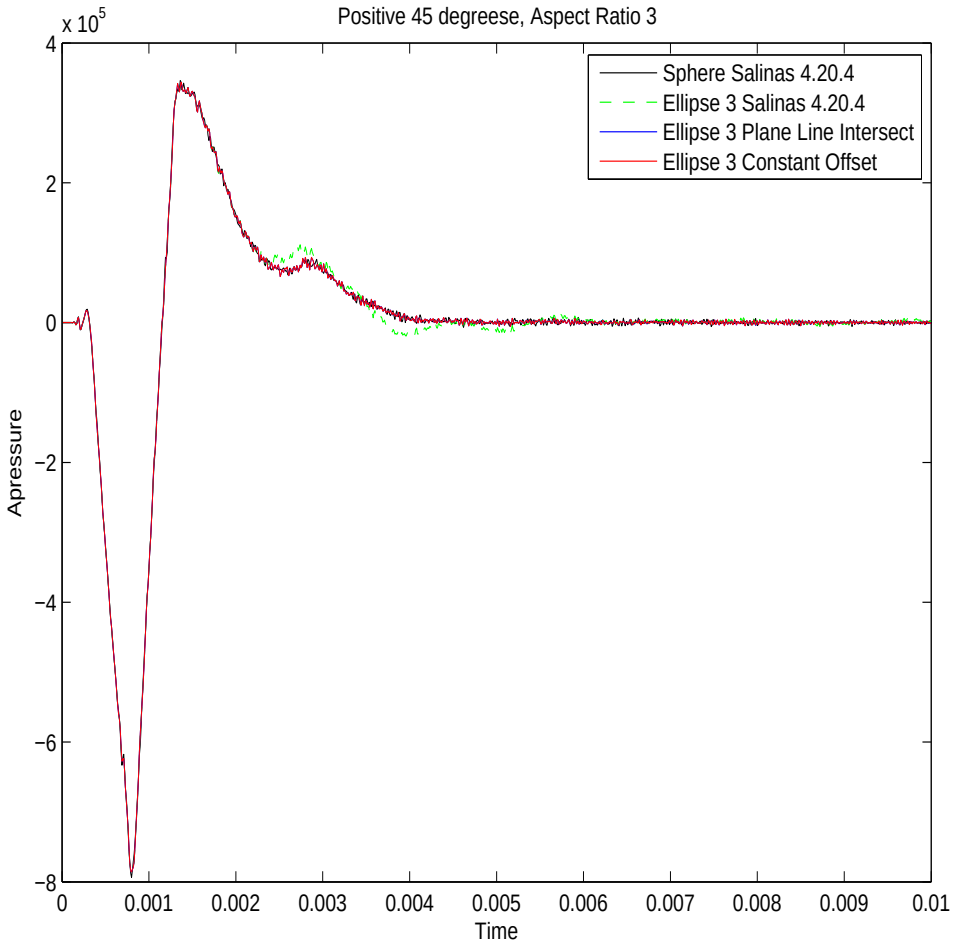


Figure 8-33. – A comparison of results along a 45^o angle from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 3:1

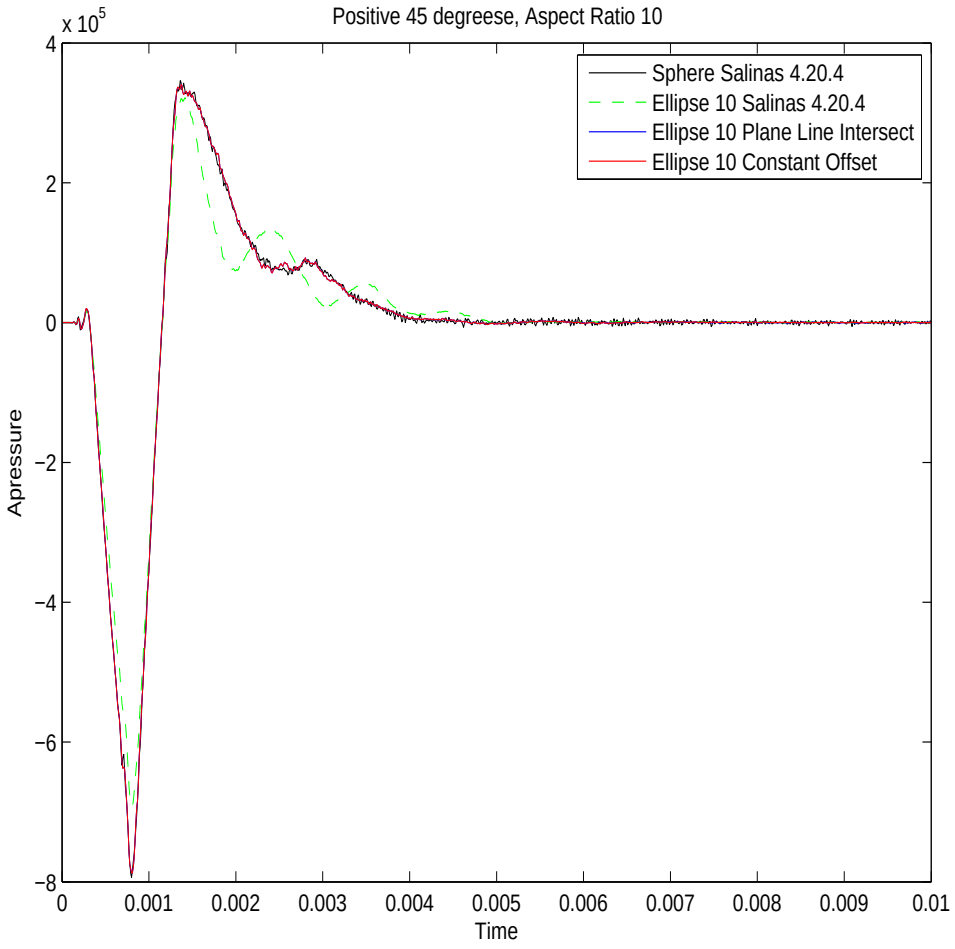


Figure 8-34. – A comparison of results along a 45^o angle from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 10:1

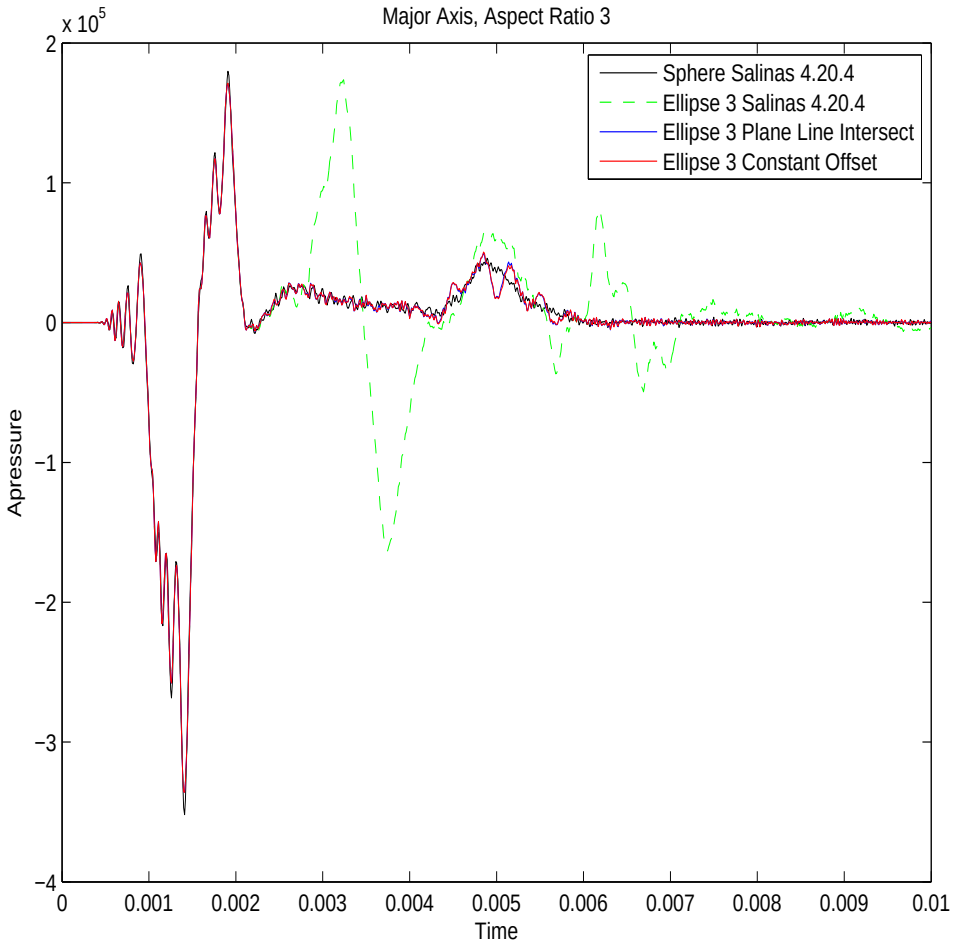


Figure 8-35. – A comparison of results along the major axis from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 3:1

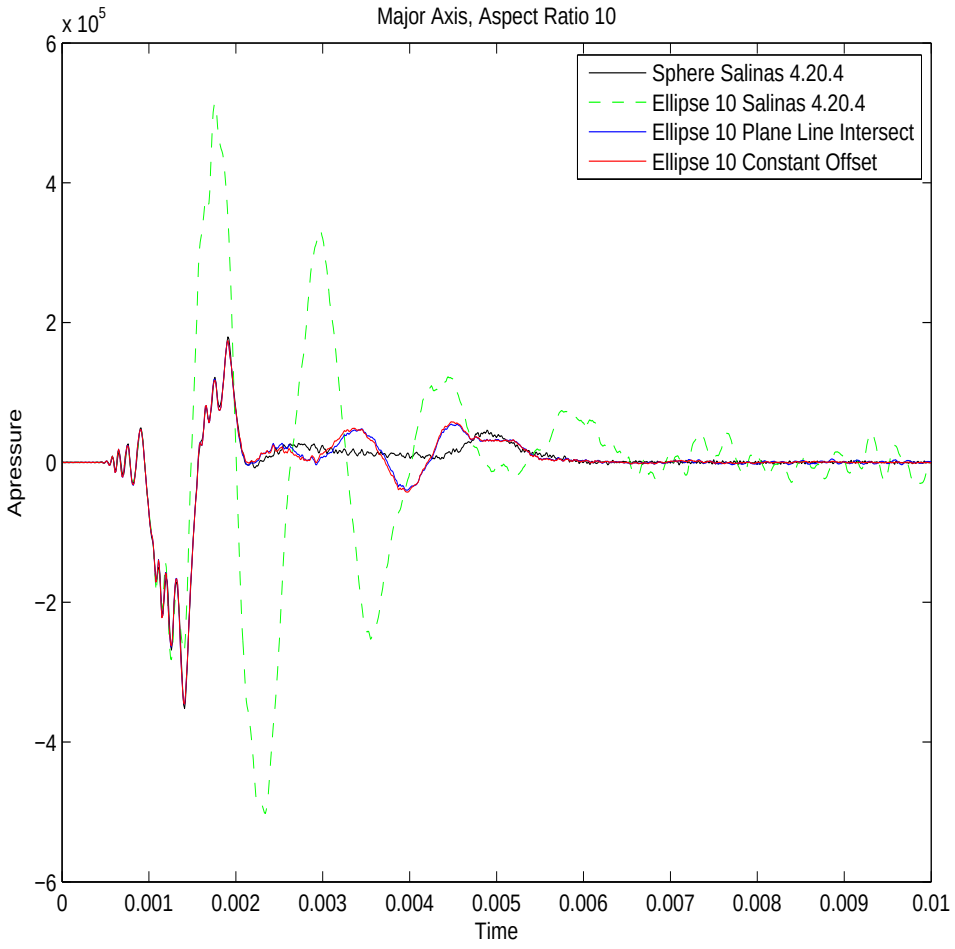


Figure 8-36. – A comparison of results along the major axis from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 10:1

8.1.15. Absorbing Boundary Conditions for Infinite Elastic Spaces.

In this example we consider a perfect cube, of dimensions $1 \times 1 \times 1$, which is subjected to a pressure wave and a shear wave along one of its faces. The opposing face is designated to be an absorbing boundary condition. In both cases, we apply the loads in the frequency domain, since we have analytical solutions for the corresponding particle displacements. We note that for the shear wave loading, we needed to constrain the motion of the space to be zero in the orthogonal directions in order to match the analytical solution. This is expected, since this solution assumes no rigid body rotation of the space. We note that these tests can be found at

```
Salinas_test/verification/acoustic/infinite_elastic_space_frf_test
Salinas_test/verification/acoustic/infinite_elastic_space_frf2_test
```

In the case of a pressure wave, the amplitude of the particle displacement at the forcing boundary is given by

$$u = \frac{P}{\omega \rho c} \tag{8.1.67}$$

where P is the pressure wave amplitude, ω is the circular frequency, ρ is the material density, and c is the dilatational wave speed in the material. The solution is for the infinite space. It will test the accuracy of the absorbing boundary condition for pressure waves. Figure 8-37 shows the comparison of this exact solution with the displacements obtained by **Sierra/SD**. The results are indistinguishable.

In the case of a shear wave, the amplitude of the particle displacement at the forcing boundary is given by

$$u = \frac{T}{\omega \rho c_s} \tag{8.1.68}$$

where T is the traction wave amplitude, ω is the circular frequency, ρ is the material density, and c_s is the shear wave speed in the material. The solution is for the infinite space. It hence will test the accuracy of the absorbing boundary condition for shear waves. Figure 8-38 shows the comparison of this exact solution with the displacements obtained by **Sierra/SD**. The results are indistinguishable.

We also test the verification of the far-field evaluation. In the frequency domain, the exact solution for an outwardly propagating spherical wave is given by

$$P = \frac{A}{r} e^{-ikr}.$$

If we prescribe the value $P = P_a$ at some value of a, as in the time-domain example described above, then we have

$$P_a = \frac{A}{a} e^{-ika}.$$

This implies that $A = P_a a e^{ika}$, and thus

$$P = P_a \frac{a}{r} e^{-ik(r-a)} \tag{8.1.69}$$

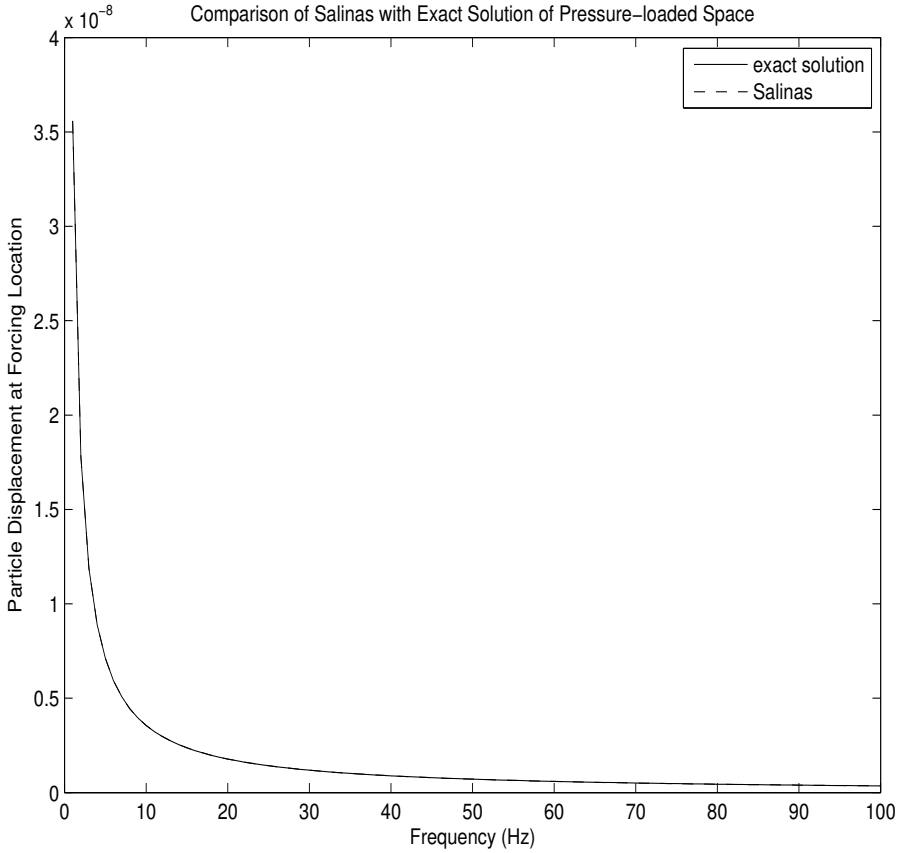


Figure 8-37. – This plot shows the comparison of **Sierra/SD** prediction with the analytical solution of particle displacement at the forcing boundary, for a perfect cube subjected to a pressure load at one end and an absorbing boundary condition at the opposite end.

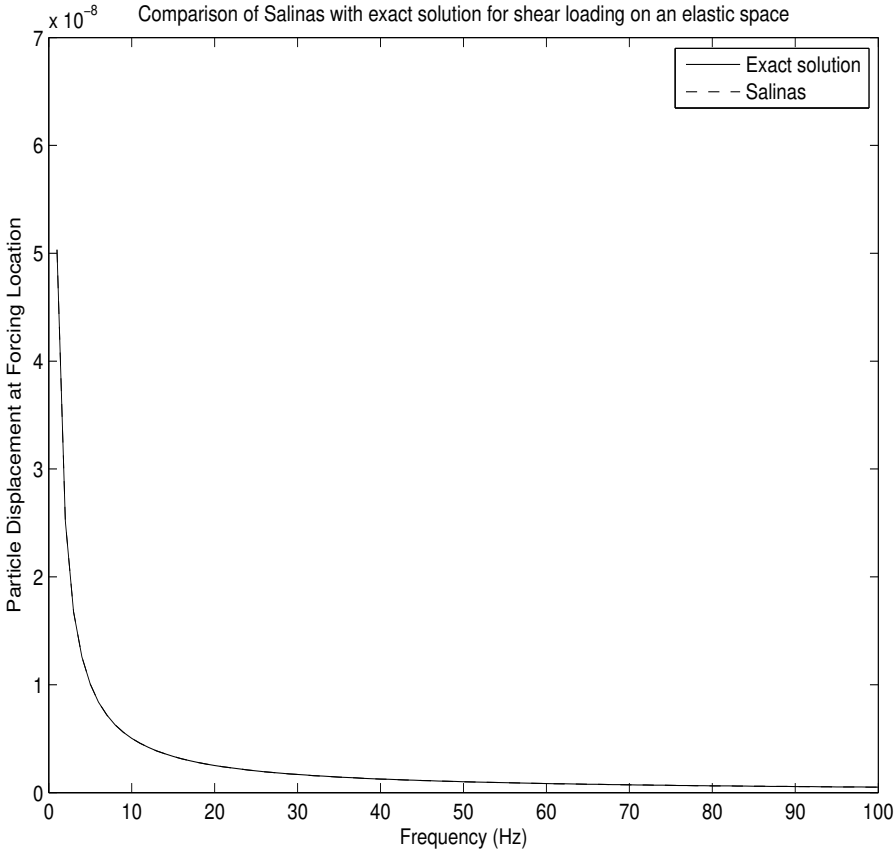


Figure 8-38. – This plot shows the comparison of **Sierra/SD** prediction with the analytical solution of particle displacement at the forcing boundary, for a perfect cube subjected to a shear load at one end and an absorbing boundary condition at the opposite end.

Equation 8.1.69 was used to compute the far-field solution to the frequency-domain version of the

8.1.16. Impedance Boundary Conditions

A simple impedance boundary condition has been implemented in **Sierra/SD**. This boundary condition relates the acoustic pressure and particle velocity on the surface. In the implementation, it results in a damping matrix with a multiplicative coefficient that depends on the impedance. For more details, we refer to the theory notes.

We consider an air-filled acoustic waveguide of length L. At the left end, we apply a prescribed particle velocity V, and at the right end, we apply an impedance boundary condition with an impedance of Z. The exact solution to this problem is given by Kinsler[32] as

$$p = V\rho c * \frac{\frac{Z}{\rho c} + jtan(kL)}{1 + j\frac{Z}{\rho c}tan(kL)} \tag{8.1.70}$$

where p is the acoustic pressure at the left end, ρ is the density, c is the speed of sound, $k = \frac{\omega}{c}$ is the wave number, and j is the imaginary number.

We consider an example with the following properties: $L = 5$, $c = 332.0$, $\rho = 1.293$, and $Z = 0.5\rho c$. Given these parameters, we ran a directfrf analysis in **Sierra/SD** and compared in Figure 8-39 the **Sierra/SD** results against the analytic solution in equation 8.1.70. An excellent agreement is observed.

The corresponding regression test uses the input deck

`Salina_rtest/verification/acoustic/waveguide_impedance.inp`

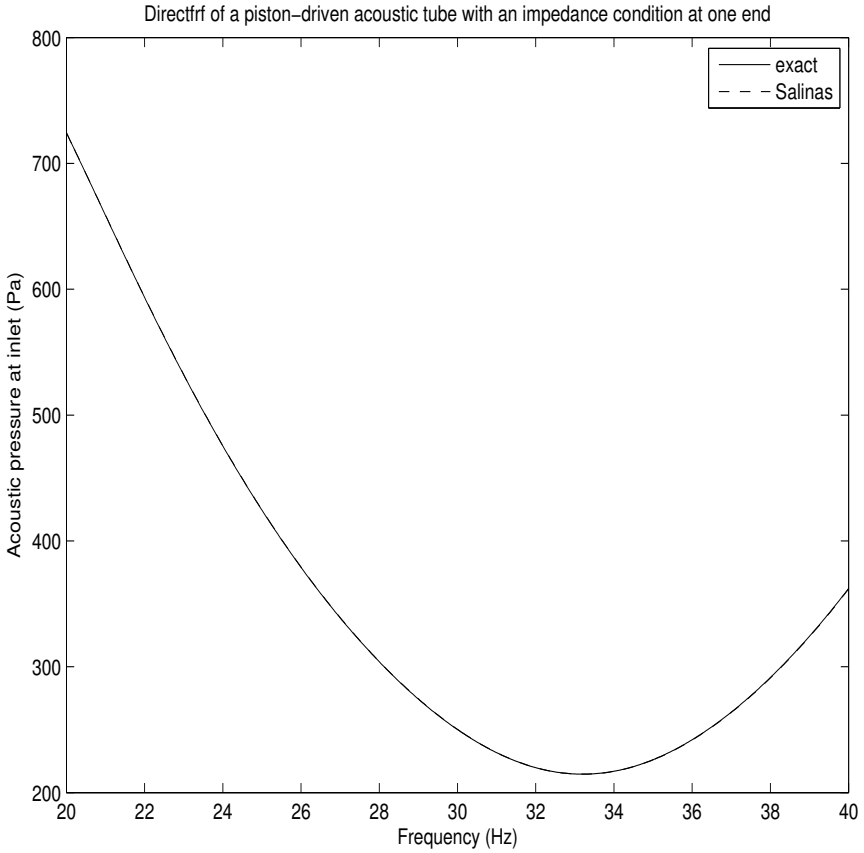


Figure 8-39. – This plot shows the comparison of **Sierra/SD** prediction with the analytical solution of acoustic pressure, for a piston-driven acoustic wave tube with an impedance boundary condition at the opposite end.

8.1.17. Moving Point Source

In this section, we study a similar example as the previous one, except that the point source has a translation superimposed on the sinusoidal volume change. For simplicity, we assume that the point source is moving in a straight line with velocity V. The exact solution for this problem is given as[36]

$$p(R,t) = \frac{\rho}{4\pi}\frac{\dot{Q}(t-\frac{R}{c})}{R(1-M\cos\theta)^2}H(t-\frac{R}{c}) + \frac{\rho Q(t-\frac{R}{c})}{4\pi}\frac{(\cos\theta - M)V}{R^2(1-M\cos\theta)^2}H(t-\frac{R}{c}) \qquad (8.1.71)$$

where Q is the same as the preceding example, $M = \frac{V}{c}$ is the Mach number of the point source, R is a vector going from the field point of interest to the source location, and θ is the angle between the direction of motion of the source and the vector R.

We note that in the case when the velocity $V = 0$ of the source is zero, we have that $M = 0$. In that case, the second term in equation 8.1.71 is zero and equation 8.1.71 reduces to equation 5.10.1. Also, we note that equation 8.1.71 is derived by assuming that the point source is moving subsonically, i.e., that the Mach number $M < 1$. In the case $M > 1$, a similar equation can be derived (see,[36] but we will not consider it here.

Figure 8-40 shows the geometry for the test problem in this case. It consists of a single hex element that moves in the x direction, along the center line of an acoustic half-space. The second time derivative of the volume of this hex element is mapped to the acoustic space, creating an image of a moving source. The hex element moves with a constant velocity. Its volume is given by the equation

$$Q(t) = \frac{8}{3\sqrt{3}}\left(r_0 + \Delta\sin(\omega t)\right)^3 \qquad (8.1.72)$$

where $r_0 = 0.01 * \sqrt{(3)}$, $\Delta = 0.01$, and $\omega = 100 \times 2\pi$. Two subsequent time derivatives of this function give the necessary expressions for $\dot{Q}$ and $\ddot{Q}$ for the time derivatives of volume that are mapped to the acoustic space. Given these, equation 8.1.71 can be used to compute the exact solution.

Figure 8-41 shows the comparison of computed and analytical solutions for the case when the hex is given a velocity of $20\frac{m}{s}$, and the measurement point is at the bottom of the acoustic hemisphere. Generally the agreement is good, with both solutions showing increasing amplitude as the hex approaches the measurement point (at $t = 0.025$), and decreasing amplitude as the hex passes and travels away from the measurement point ($0.025 < t < 0.05$). Better agreement could likely be obtained by refining both the acoustic and hex meshes, but that is not pursued here. We note that this example can be found in the performance test suite (it was too large to be placed in the verification suite) at

```
Salinas_rtest/performance/moving_source.inp
```

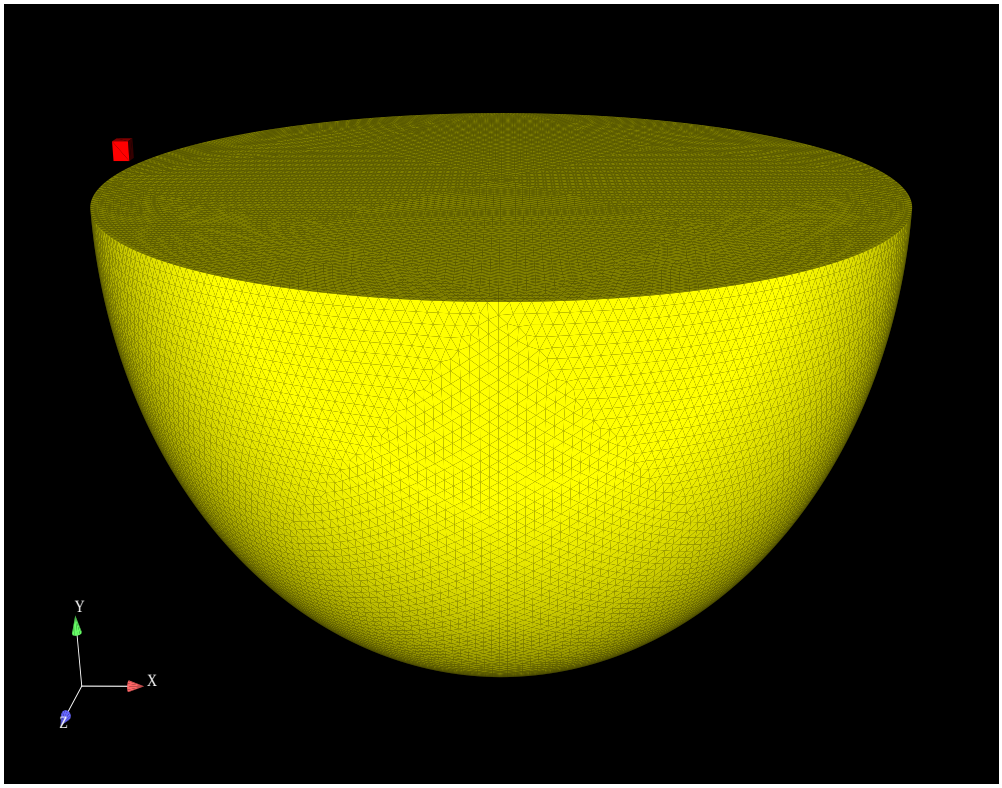


Figure 8-40. – Geometry for verification example of moving point acoustic source in an infinite medium.

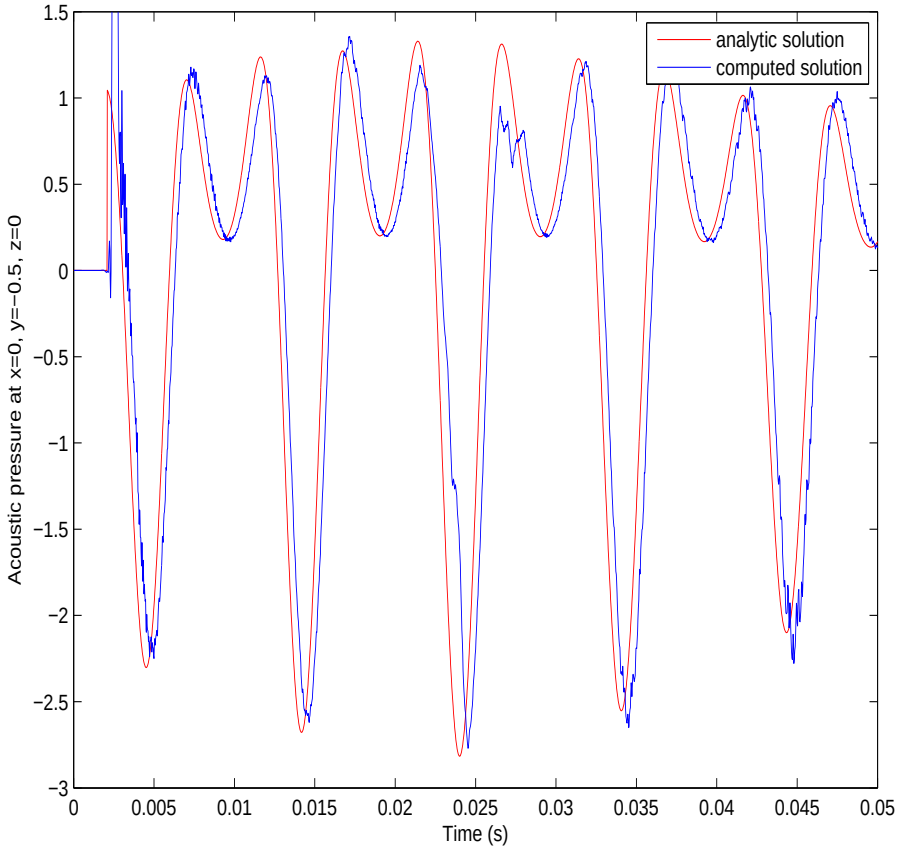


Figure 8-41. – Comparison of computed and analytic solutions for verification example of moving point acoustic source in an infinite medium.

8.1.18. Infinite Elements for Transients

The infinite element implementation was verified on a single element transient example. This element was a hex element that was aligned with a spherical surface of radius $a = 100m$. A surface acceleration excitation of $\sin(2\pi t)$ was applied to the free face of the hex element, and a third order infinite element was defined on the opposite face. Since this element was aligned with a spherical coordinate system, its exact solution should be the same as that of the sound pressure radiated from a pulsating sphere of the same radius. This exact solution is given in[39] as

$$\phi(t) = a \int_{-\infty}^{t} e^{-(c/a)(t-\tau)} v_S(\tau) d\tau \tag{8.1.73}$$

where a is the radius of the sphere, c is the speed of sound, and $v_S(t)$ is the applied surface velocity on the inner surface of the sphere. Once $\phi(t)$ is found, the acoustic pressure can be recovered as follows

$$p(r,t) = \frac{\rho c \dot{\phi}}{r} \tag{8.1.74}$$

If we define an input surface acceleration as

$$a_S(t) = \sin(2\pi t) \tag{8.1.75}$$

Then we have an implied input velocity of

$$v_S(t) = \frac{-1}{2\pi}\cos(2\pi t) + \frac{1}{2\pi} \tag{8.1.76}$$

Substituting this into equation 8.1.73, we obtain

$$\phi = \frac{-a}{2\pi} \int_{-\infty}^{t} e^{-(c/a)(t-\tau)} \left[\frac{-1}{2\pi}\cos(2\pi t) + \frac{1}{2\pi}\right] d\tau \tag{8.1.77}$$

Simplifying, and using the identity

$$\int e^{c_1 x}\cos(c_2 x) = \frac{e^{c_1 x}}{c_1^2 + c_2^2}\left(c_1\cos(c_1 x) + c_2\sin(c_2 x)\right) \tag{8.1.78}$$

we obtain

$$\phi(t) = \frac{-a}{2\pi}\frac{1}{(\frac{c}{a})^2 + (2\pi)^2}\left[\frac{c}{a}\cos(2\pi t) + (2\pi)^2\sin(2\pi t)\right] + \frac{2\pi a^2}{c\left(\frac{c}{a}\right)^2 + c(2\pi)^2} e^{\frac{-ct}{R}} \tag{8.1.79}$$

Inserting this expression into equation 8.1.74, we obtain the exact solution on the surface of the sphere (R=a)

$$p(r,t) = \frac{\rho c}{\left(\frac{c}{a}\right)^2 + (2\pi)^2}\left[2\pi e^{\frac{-ct}{a}} + \frac{c}{a}\sin(2\pi t) - 2\pi\cos(2\pi t)\right] \tag{8.1.80}$$

We note that there is both a transient and a steady-state component to the solution in equation 8.1.80. The transient term dies out after sufficient time, and then the steady terms persist.

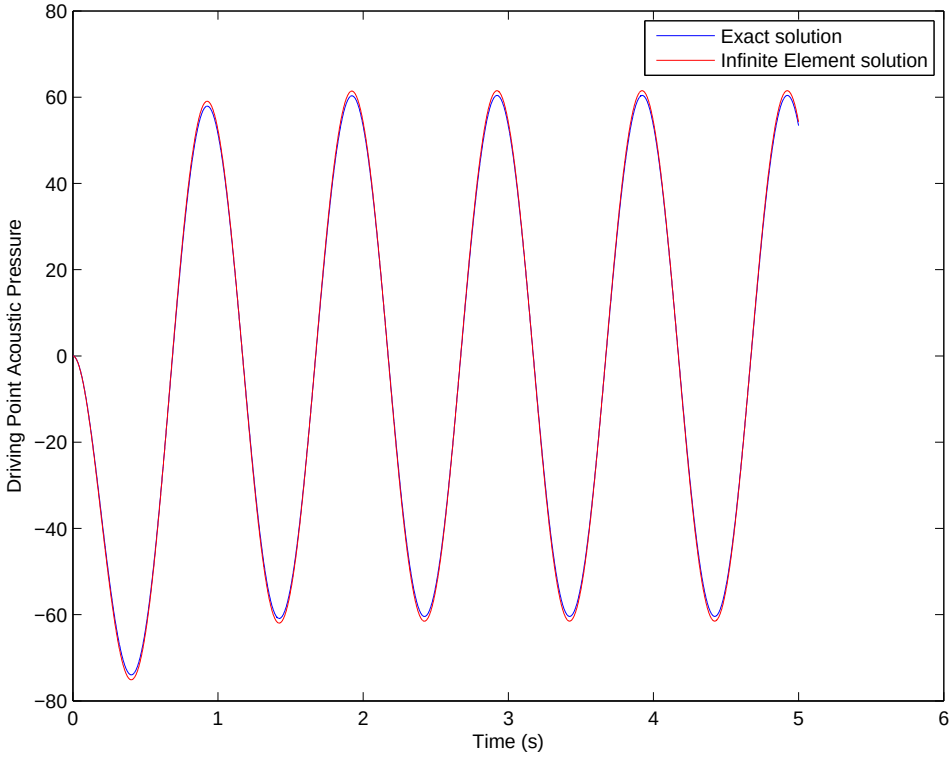


Figure 8-42. – A comparison of an exact solution for spherical wave radiation and the **Sierra/SD** computation using transient infinite elements.

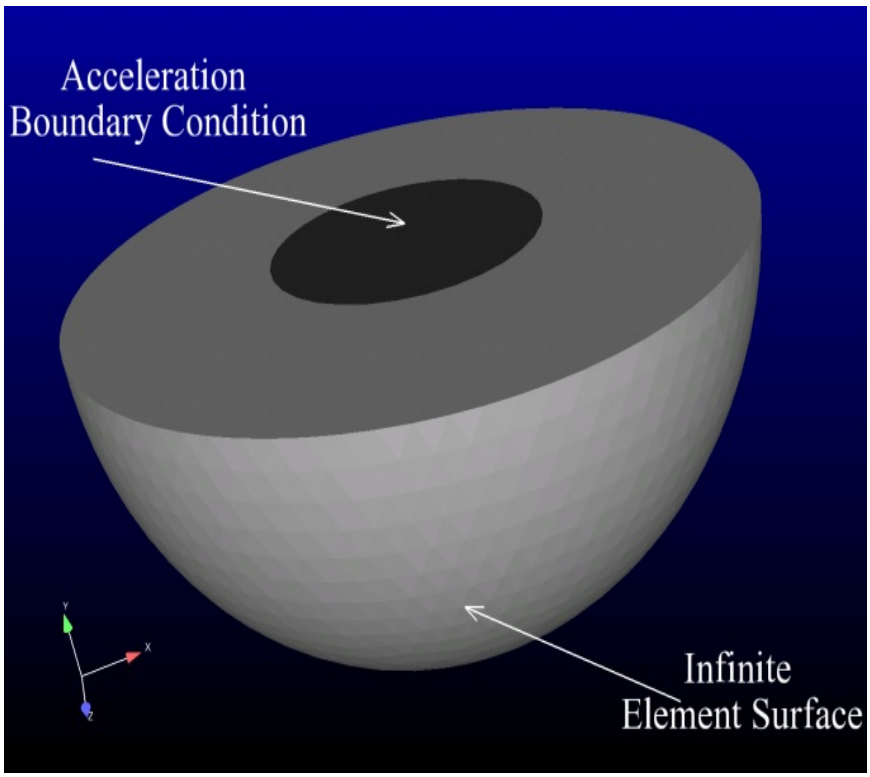


Figure 8-43. – A schematic of the geometry of a piston mounted on an infinite baffle for verification of transient infinite elements.

Figure 8-42 shows the comparison of the exact solution of equation 8.1.80 and the computed solution using **Sierra/SD**. Excellent agreement is seen between the curves.

A second verification example was considered that consisted of a piston mounted on an infinite baffle. Figure 8-43 shows a schematic of the geometry. A 3D hemispherical domain of radius $0.5(m)$ was constructed and meshed with tetrahedrons. A normal acceleration boundary condition was applied to a circular portion of the flat face, of radius $0.25(m)$. The flat plane of the hemisphere was set at $y = 0$, as shown in Figure 8-43. The remaining part of the flat surface was treated as acoustically rigid (zero particle acceleration). Infinite elements were then applied to the curved surface, thus making the geometry appear to be a semi-infinite space with a piston mounted on the (rigid) baffle.

The analytical solution to this problem is given as[39]

$$p(x,t) = \frac{\rho}{2\pi}\int_S \frac{a_n(x_s, y_x, t - R/c)}{R} dS \tag{8.1.81}$$

where $p(x,t)$ is the acoustic pressure at an arbitrary point x in space and time t, ρ is the fluid density, $a_n(x_s, y_x, t - R/c)$ is the normal acceleration on the piston surface, x_s and y_s are points on the piston used in the surface integration, $R = \sqrt{\left[(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2\right]}$ is the distance from a point on the piston surface to the point x where the solution is desired, and c is the speed of

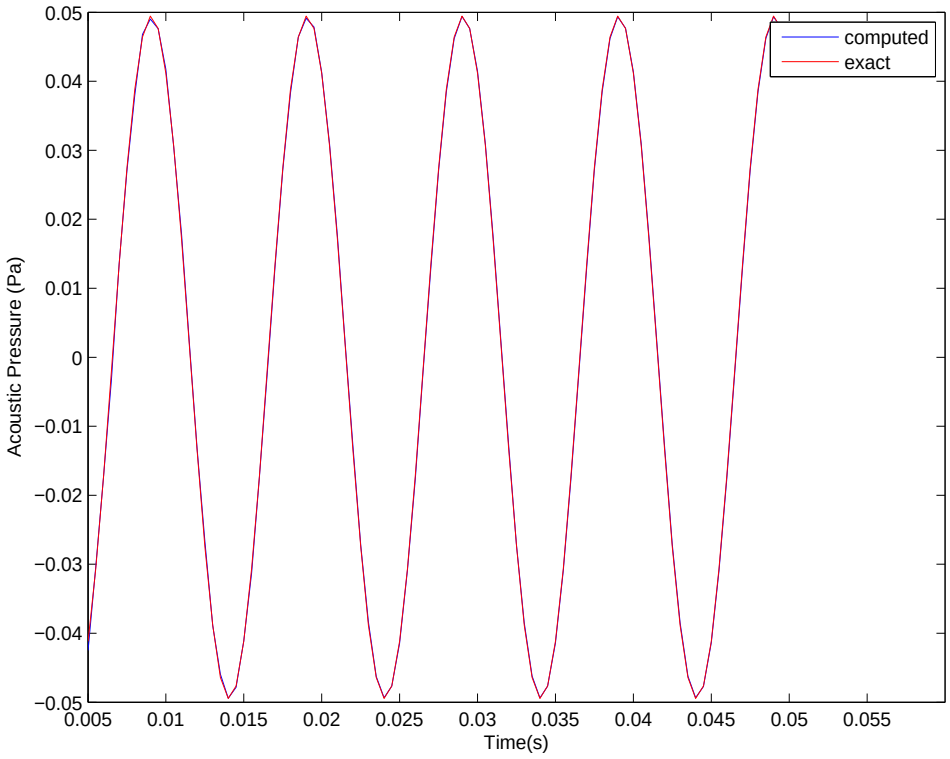


Figure 8-44. – A comparison of computed vs. analytic solution for a piston mounted on an infinite baffle. Field point is at $x = 0$, $y = -0.5$, $z = 0$.

sound. Thus, we see that for an arbitrary point in space x, and an arbitrary time history of accelerations a_n, the integral in equation 8.1.81 must be carried out numerically.

We consider 2 points in space for the comparison with analytical solution. The first point (point A) is located along the axis of the piston at $x = 0$, $y = -0.5$, and $z = 0$. The second point (point B) is located off-axis as $x = 0.5$, $y = 0$ and $z = 0$. Figures 8-44 and 8-45 show comparisons of the analytical and computed solutions for the case when $a_n(t) = \sin(200\pi t)$, which corresponds to the case when the piston is rigid and moving harmonically at a frequency of 100Hz.

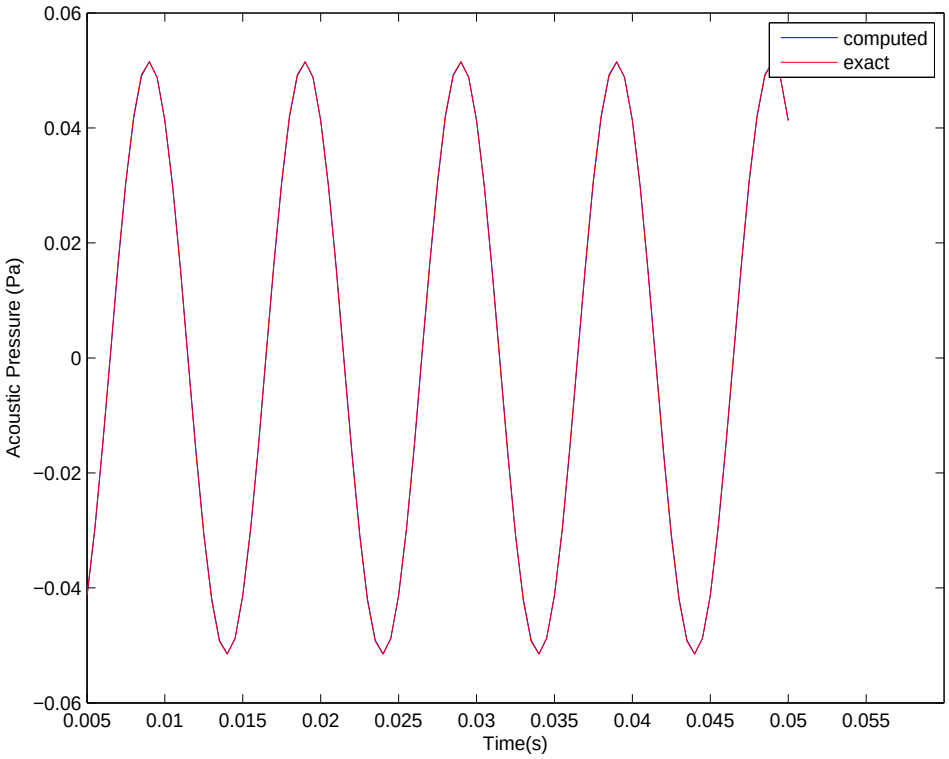


Figure 8-45. – A comparison of computed vs. analytic solution for a piston mounted on an infinite baffle. Field point is at $x = 0.5$, $y = 0.0$, $z = 0$.

8.1.19. Variable Order Infinite Element Implementation

Before making comparisons of the infinite element and Kirchhoff integral approaches, we first examine the dependence of the infinite element approach on the order of the radial expansion used in the approximation. If the implementation is correct, the computed solution should converge to the analytical solution for sufficiently high order of radial expansion in the infinite element approximation.

Figure 8-46 shows the geometry of the mesh used for the baffled piston. It consists of a hemispherical geometry with a circular surface defining the area over which the piston makes contact with the air. An applied acceleration time history is given to the piston, which acts as a Neumann boundary condition. The flat face of the hemisphere is a subset of the infinite baffled plane. The infinite elements are placed on the curved part of the hemispherical surface. The piston is given a uniform, time-dependent acceleration in the direction of its surface normal. We denote this acceleration as $a_P(t)$, and the exact form of the time dependence will take two different forms, as described below.

The exact solution to this problem can be computed from the Kirchhoff integral

$$p(\boldsymbol{x},t) = \frac{\rho}{2\pi}\int_S \frac{a_P(\boldsymbol{x_S}, t-\frac{R}{c})}{R} dS \tag{8.1.82}$$

where $p(\boldsymbol{x},t)$ is the acoustic pressure at point $\boldsymbol{x}$ and time t, ρ is the density of the fluid, S is the surface area over which the piston interacts with the fluid, $a_P(\boldsymbol{x_S}, t-\frac{R}{c})$ is the normal acceleration of the piston at the point $\boldsymbol{x_S}$, and at the delayed time $t-\frac{R}{c}$, $R = |\boldsymbol{x}-\boldsymbol{x_S}|$ is the distance from the surface point $\boldsymbol{x_S}$ to the far field point $\boldsymbol{x}$, and c is the speed of sound. The evaluation of equation 8.1.82 was carried out numerically, and this provided the exact solution for comparison with the computations.

In the following examples, we consider standard conditions for the air surrounding the piston, $\rho = 1.293$, $c = 332.0$. The piston has a radius of 0.25(m). The mesh consists of $1,800,000$ linear tetrahedral acoustic elements with an approximate element diameter of 0.0026 m. For a wave at 2000 Hz, the wavelength is about 0.166 m, and thus this consists of about 50 elements per wavelength. The time step for the transient analysis was taken at $5.0\text{x}10^{-6}$ s, which is much finer than needed to resolve a frequency of 2000 Hz. Thus, we expect both spatial and temporal resolution to be sufficient to capture the wave response, and thus allow the infinite element and Kirchhoff solutions for far-field pressures to be easily compared.

Figure 8-47 shows a comparison of the exact vs. computed transient response at the particular point $x = -0.25$, $y = 0$, $z = 0$ for increasing order of the infinite element approximation. In this case, the piston was given an acceleration of the form $a_P(t) = \sin(2\pi f t)H(t)$, $f = 2000$(Hz). As expected, the infinite element solution converges to the exact solution as the order is increased. For the examples that follow, a similar approach was taken in that the order was increased until subsequent increases in the order of the infinite elements made no difference in the obtained results.

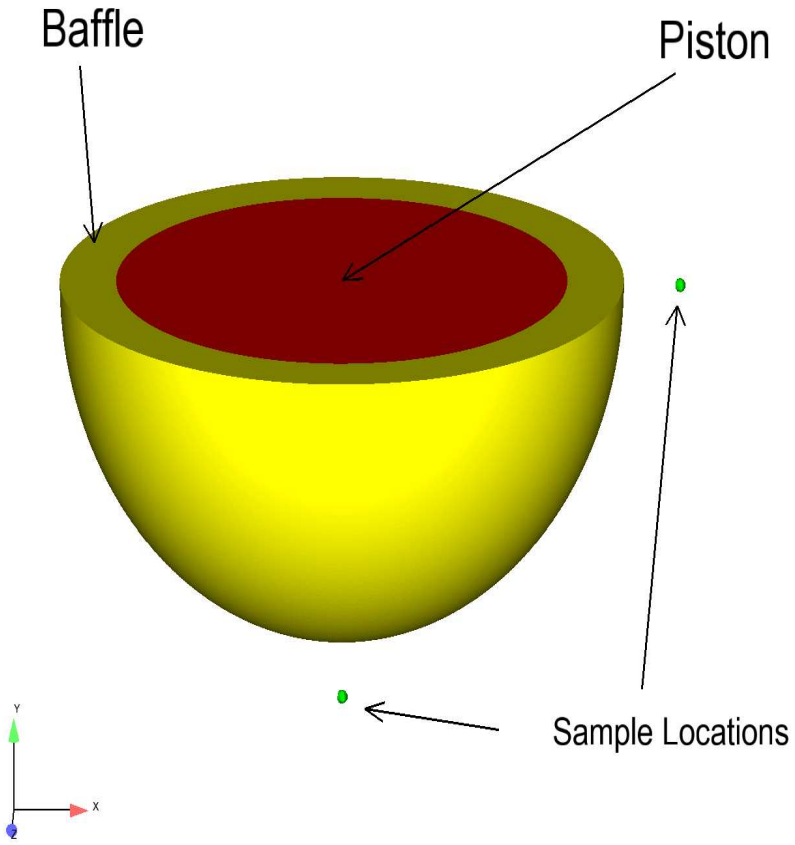


Figure 8-46. – The geometry and mesh of the baffled piston problem.

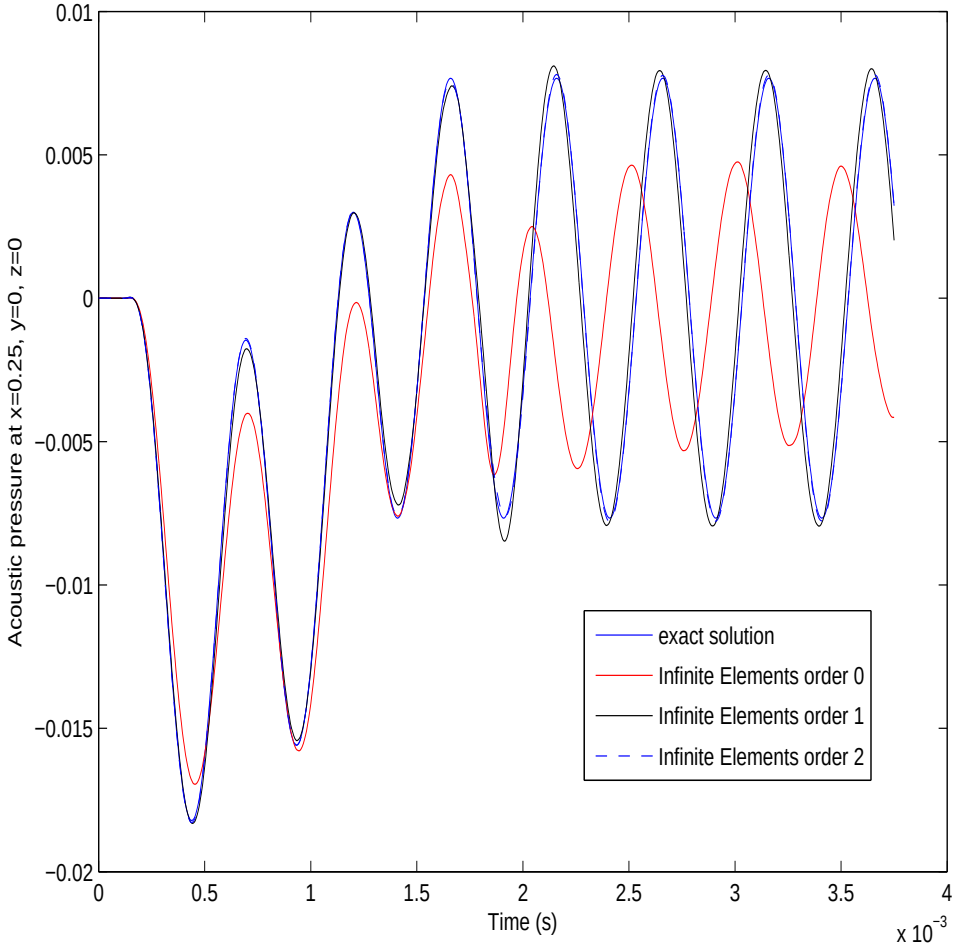


Figure 8-47. – A convergence study for infinite element order, demonstrated on the baffled piston problem

8.1.20. Coupled Acoustic-Structure Directfrf with Viscoelastic Material

This example compares the solution from ABAQUS with that of **Sierra/SD** for a coupled acoustic-structure interaction directfrf problem with a viscoelastic material. The problem consists of a thick plate fixed on the edges and loaded on one face. The opposite side of the solid is coupled a prism with a prescribed acoustic pressure equal to zero on the opposite face. A sketch of the problem domains is shown in Figure 8-48. The pressure contours for both the **Sierra/SD** and ABAQUS outputs are shown in Figures 8-50 and 8-49, respectively, while a comparison of peak values are shown in Table 8-13.

	Peak Pressure (Pa)	Peak Uy (m)
ABAQUS	-10811.5	1.031e-6
Sierra/SD	-10818.16	-1.030e-6

Table 8-13. – . Peak pressure and displacement for coupled acoustic-structure interaction problem with viscoelastic material.

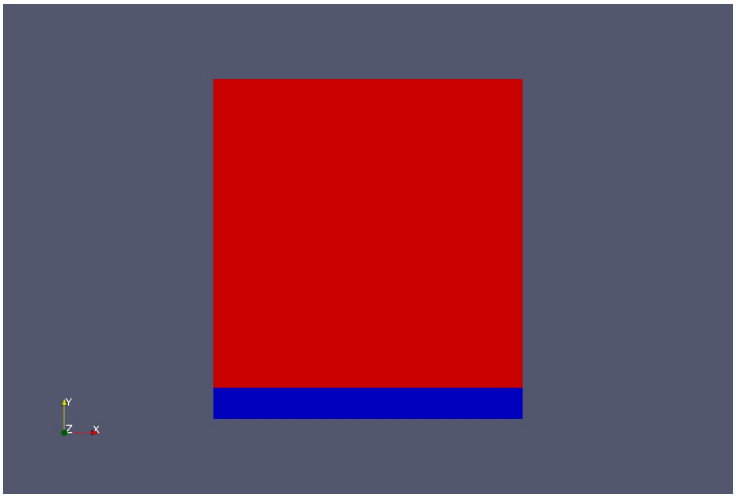


Figure 8-48. – The bottom part is the solid, the top part is the fluid

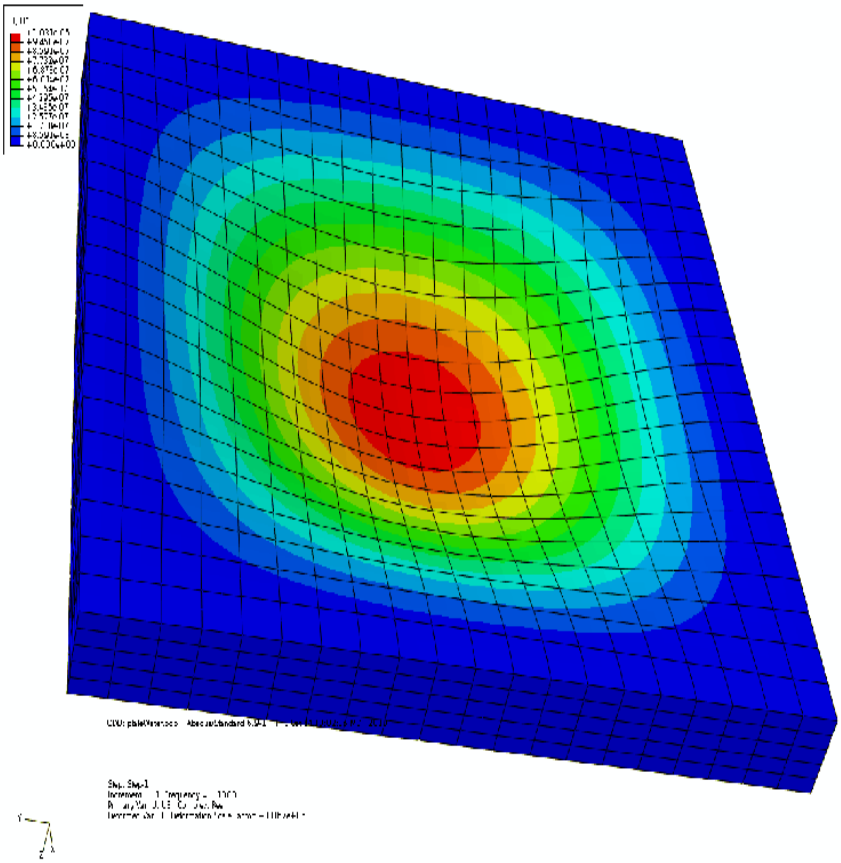


Figure 8-49. – Vertical displacement distribution from ABAQUS.

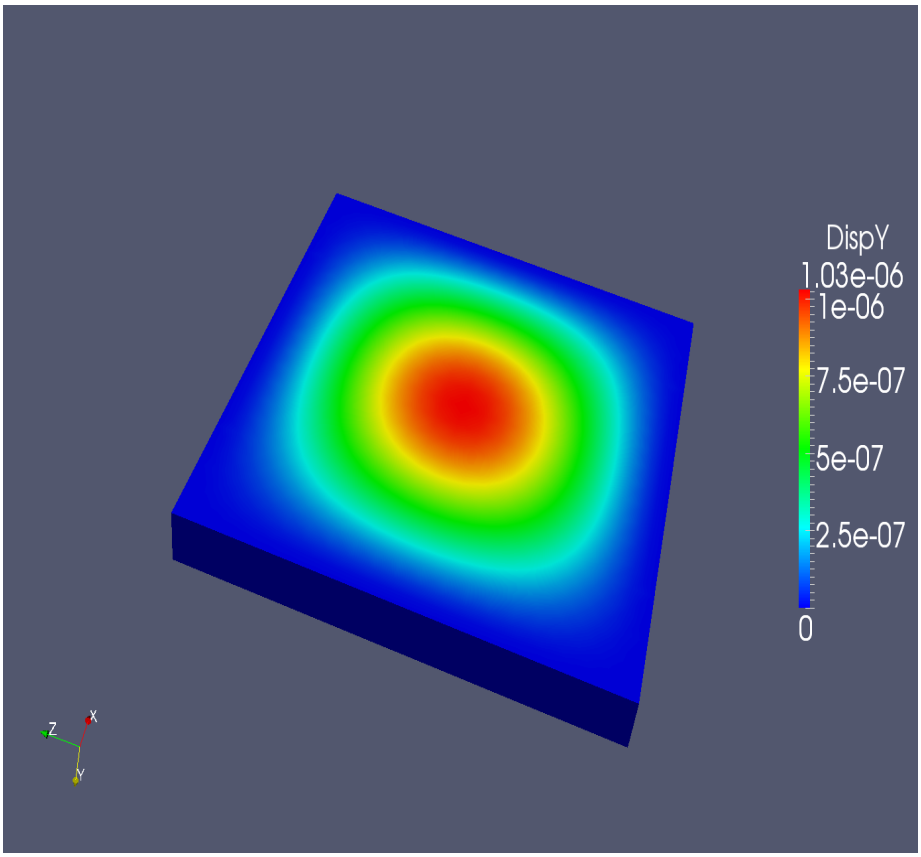


Figure 8-50. – Vertical displacement distribution from **Sierra/SD**.

8.2. Nonlinear Acoustics

In Sierra SD nonlinear acoustics is modeled using the Kuznetsov Equation. For verification purposes, we consider the same sequence of simulations given in[29,28] involving a piston-radiation problem. This example is shown in Figure 8-51. It consists of a long air-filled tube that has a sinusoidal boundary condition at the left end. This boundary condition can either be in the form of a pressure (Dirichlet) condition or a velocity (Neumann) condition, which are given as

$$p(0,t) = p_0 \sin(\omega t) \tag{8.2.1}$$

$$v(0,t) = v_0 \sin(\omega t) \tag{8.2.2}$$

To simulate the infinite condition at the right end of the tube, an absorbing boundary condition is used. The exact solution to this problem is given by the Fubini solution (see Section 11.2 of[39]) in the pre-shock regime and by the Fay solution in the post-shock regime.

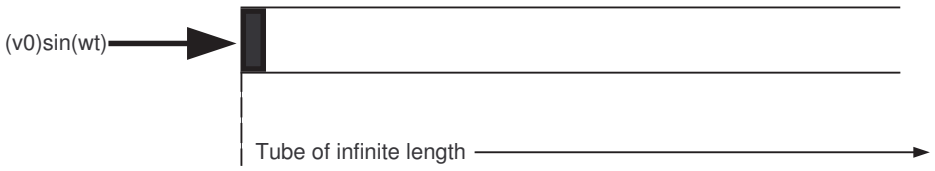


Figure 8-51. – A wave tube example for verification.

In the case of a plane wave, the distance to shock formation is given as

$$\sigma = \frac{c}{\left(1 + \frac{B/A}{2}\right) v_0 k} \tag{8.2.3}$$

where v_0 is the amplitude of the velocity of the source, and k is the wave number. As expected, for larger amplitude sources, and for more nonlinear fluids (larger B/A), the shock forms closer to the source. Interestingly, we see that the shocks also form closer to the source for high frequency waves, since k is in the denominator. In the numerical experiment, we chose $v_0 = 20\frac{m}{s}$, and $k = \frac{100}{332} = .3$, which resulted in a shock formation distance of $\sigma = \frac{332}{1.2*20*.3} = 46.1m$.

The Fubini solution[33,26] is given by

$$p(x,t) = p_0 \sum_{n=1}^{\infty} \frac{2}{n\bar{x}} J_n(n\bar{x}) \sin(n\omega\tau) \tag{8.2.4}$$

where $J_n(x)$ is the Bessel function of order n, $\bar{x} = \frac{x}{\sigma}$, and $\tau = t - \frac{x}{c_0}$. The Fay solution is

$$p(x,t) = p_0 \frac{2}{\Gamma} \sum_{n=1}^{\infty} \frac{\sin(n\omega\tau)}{\sinh\left[n(1+\sigma)\Gamma\right]} \tag{8.2.5}$$

where Γ is the ratio of the absorption length to the shock formation distance (see[26]). The Fubini solution assumes a lossless media, and is valid for $x < \sigma$. For the post-shock regime, $x > 3.5\sigma$,

the Fay solution must be used since it accounts for absorption. Transition solutions have been derived[7] that provide exact solutions for $\sigma < x < 3.5\sigma$, but we do not consider those here.

For all the results presented next, the fluid is air at ambient conditions, with $c = 332.0\frac{m}{s}$, $\rho = 1.293\frac{Kg}{m^3}$. Accounting for viscosity and thermal conductivity loss mechanisms, the absorption parameter can be calculated from the following equation[39]

$$\frac{b}{c^2} = \frac{1}{\rho c^2}\left[\frac{4}{3}\eta + (\gamma - 1)\frac{k}{C_p}\right] \tag{8.2.6}$$

$$= \frac{1}{1.293x332^2}\left[\frac{4}{3}1.846e^{-5} + (0.4)\frac{2.624e^{-2}}{1000}\right] \tag{8.2.7}$$

$$= 7.017e^{-6}x\left[2.461e^{-5} + 1.0496e^{-5}\right] = 2.46e^{-10} \tag{8.2.8}$$

$$\tag{8.2.9}$$

For air, $\frac{b}{c^2}$ is too small to affect the results. Note that this estimate neglects additional loss mechanisms such as molecular relaxation, and wall losses.

Figures 8-52, 8-53, and 8-54 show the solution at $x = 0$, $x = \sigma$, and $x = 4\sigma$, respectively. In all cases, the computed solution is compared with the exact solution, and convergence is obtained. In these results, three- dimensional linear finite elements were used, with element diameters of 0.125(m). The time steps were 1.0×10^{-3}, 2.5×10^{-4}, and 1.25×10^{-4} for Figures 8-52, 8-53, and 8-54, respectively.

To demonstrate the significant difference between linear and nonlinear solutions, in Figure 8-55 we show the results for the previous problem using linear and Kuznetsov wave equations. In this case, we plot acoustic pressure with distance along the tube, rather than with time. It is seen that linear theory is not sufficient for capturing the correct response.

Next, we examine the nonlinear convergence properties of the algorithm. Since we are using Newton's method to solve the nonlinear system of equations, we examine the number of iterations required for convergence. The criteria for convergence is based on a relative tolerance of 10^{-6}, e.g.

$$\frac{|Res_f|}{|F_{ext}|} \leq 10^{-6} \tag{8.2.10}$$

Also, we mention that the starting point for the Newton iterations is the value of velocity potential from the previous time step. Figure 8-56 shows the number of Newton iterations required to satisfy the inequality 8.2.10, for various levels of input velocities of the piston. As expected, for larger input velocities, more iterations are required for convergence. The highest level that was considered, $120\frac{m}{s}$, is beyond the limitations of the Kuznetsov equation, but we show it anyway to illustrate the divergence of the Newton scheme. For reasonable levels of piston velocities, $20\frac{m}{s}$, the Newton iterations converge rapidly, leveling off at about 4 iterations per time step. Interestingly, for source amplitudes that are within the range of validity of the Kuznetsov equation, the formation of shocks does not influence the number of iterations required for convergence.

A test case for the Fubini solution with the shock wave is currently in the verification test suite

`Salinas_rtest/verification/acoustics/shocktube_english.test for english units`

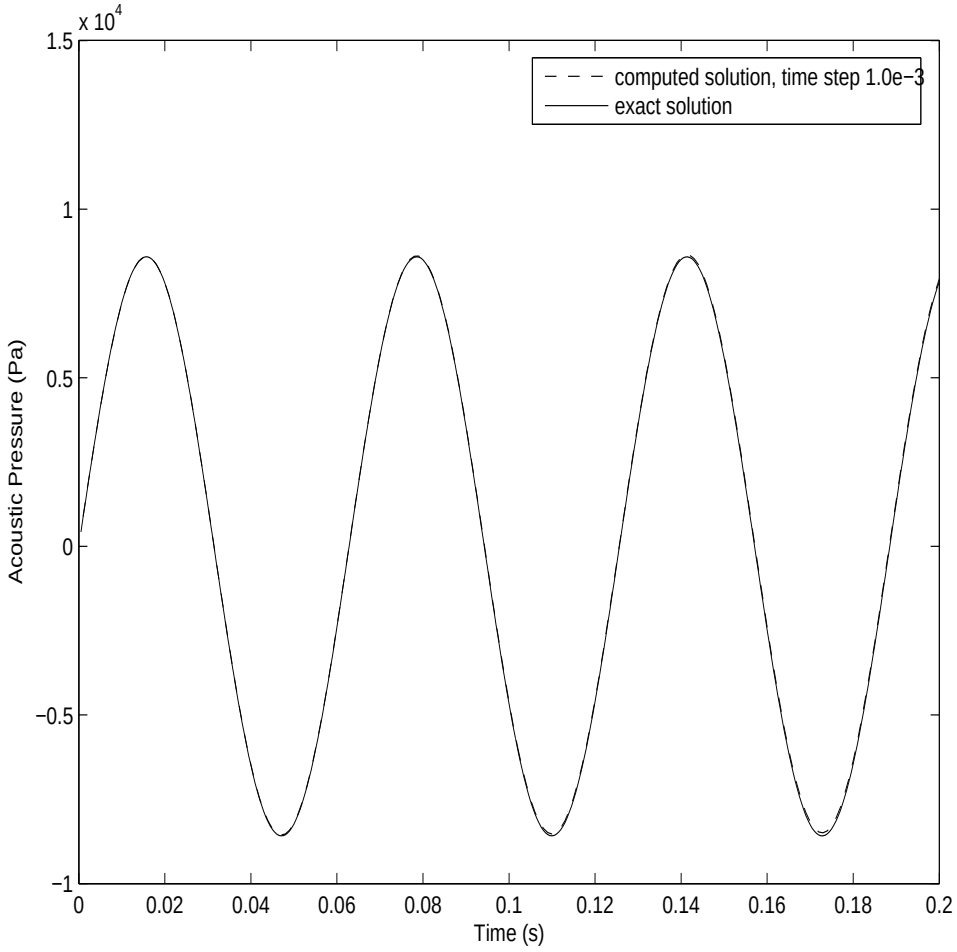


Figure 8-52. – Acoustic radiated pressure at $x = 0$

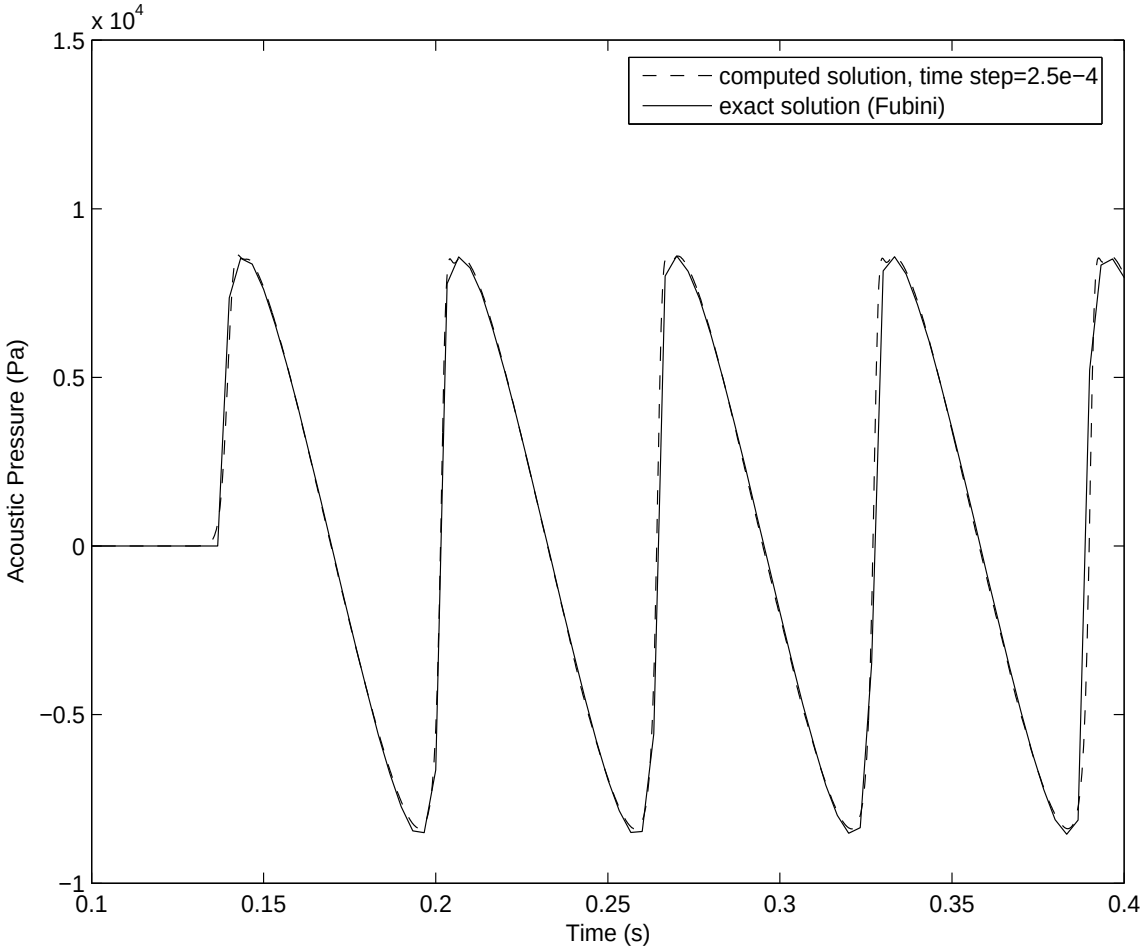


Figure 8-53. – Acoustic radiated pressure at $x = \sigma$.

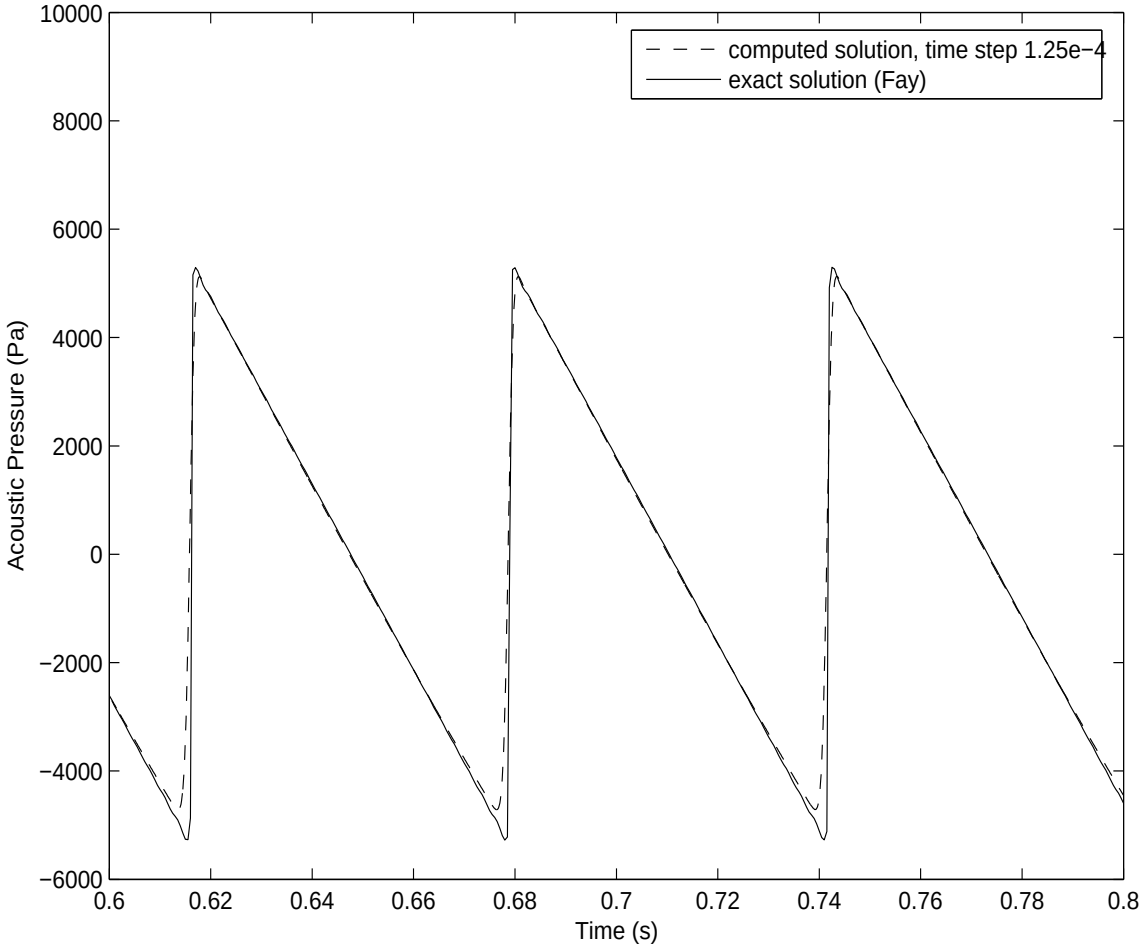


Figure 8-54. – Acoustic radiated pressure at $x = 4\sigma$.

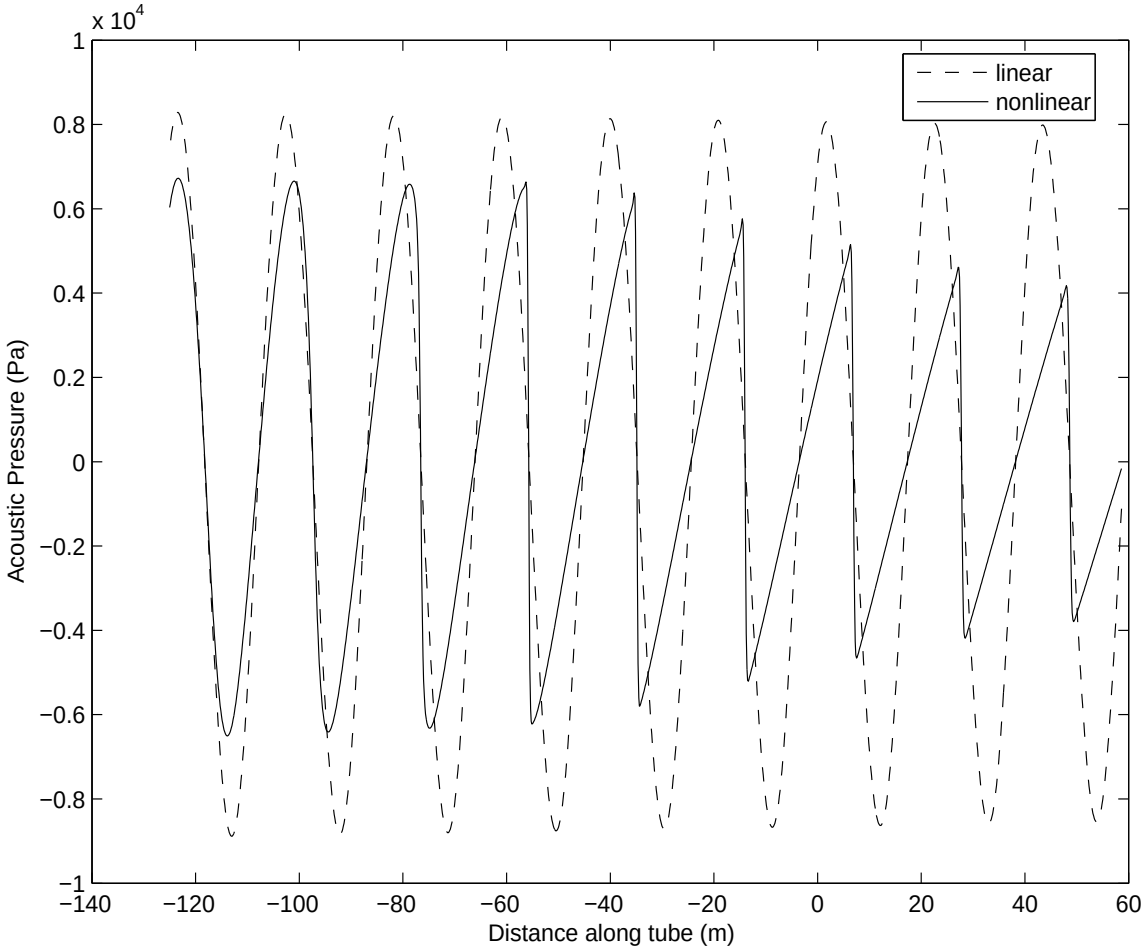


Figure 8-55. – A comparison of radiated pressure using linear and nonlinear acoustic formulations

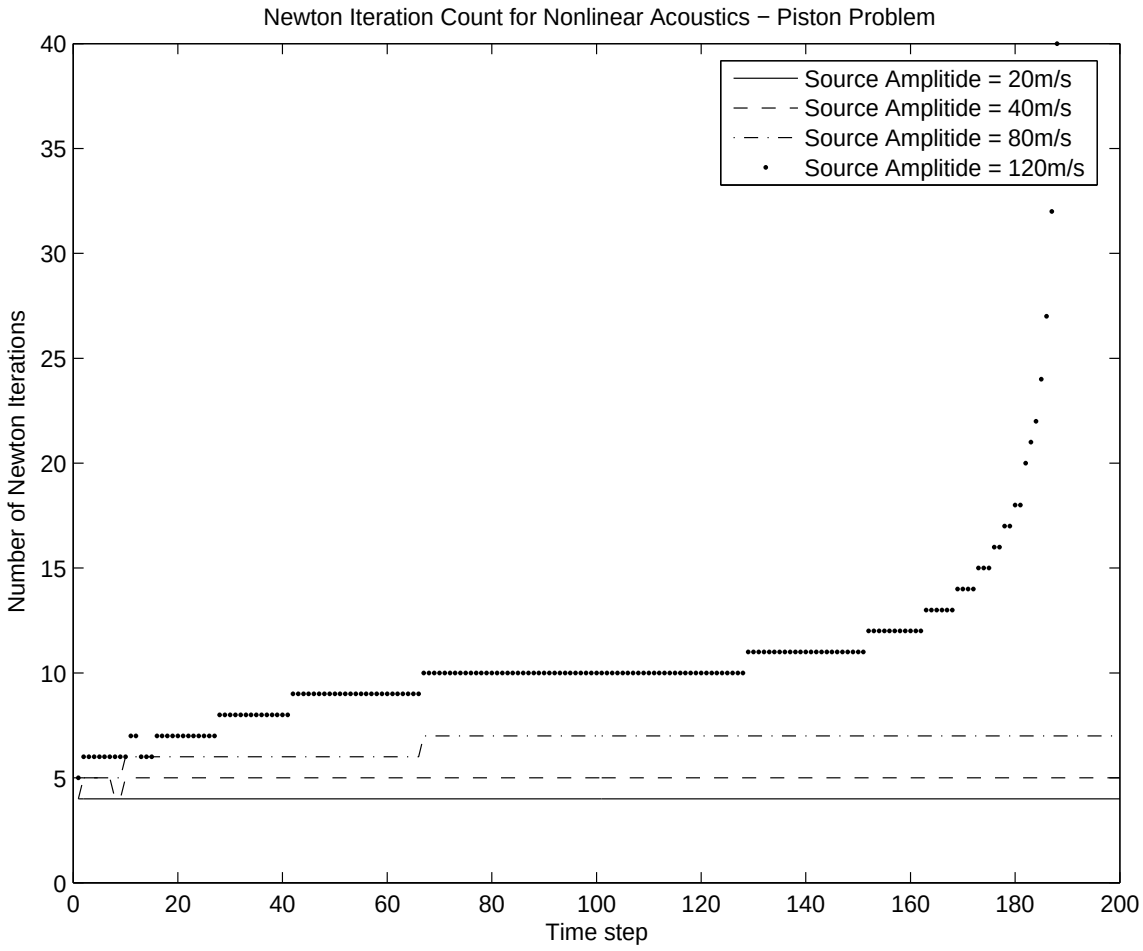


Figure 8-56. – The number of Newton iterations required for convergence of the piston radiation problem.

8.2.1. Eigen Analysis

Eigen analysis is performed as part of the verification of the element quantities. Practically speaking, it is difficult to verify the analysis independent of the element. For example, the hex20 and tet10 element convergence studies utilize eigen analysis for the convergence study. See Figures 4-49 and 4-50 for example.

8.2.2. Quadratic Eigen Analysis

There are several different solution approaches within the package that computes the solution to the quadratic eigenvalue problem. Each requires its own verification.

8.2.2.1. QEP – Proportionally Damped

The proportionally damped system is straightforward because the eigenvectors of the real system diagonalize the complex (or damped) solution. Consider

$$(K - \omega^2 M)\phi = 0 \tag{8.2.11}$$

For this system $\phi^T K \phi = \Lambda$ is diagonal, and $\phi^T M \phi = I$. The proportional damping matrix is given by $C = \alpha M + \beta K$. Also $\phi^T C \phi = \alpha I + \beta \Lambda$.

The solution to the j^{th} mode of the damped system is given by,

$$\Lambda_{jj} + \omega(\alpha + \beta \Lambda_{jj}) + \omega^2 = 0 \tag{8.2.12}$$

All quantities are known from the real eigenvalue analysis, and we can solve in terms of ω.

$$\omega_j = \frac{-(\alpha + \beta \Lambda_{jj}) \pm \sqrt{(\alpha + \beta \Lambda_{jj})^2 - 4\Lambda_{jj}}}{2} \tag{8.2.13}$$

Table 8-14 lists the eigenvalues and errors for a proportionally damped system with $\alpha = 0$ and $\beta = 0.001$. This is a small *Hex8* model for which the eigenvalues are known from real eigen analysis.

These solutions are within the expected round off. Notice that as the natural frequency increases, the fractional damping is increasing to almost 25%.

Table 8-14. – Eigenvalues of Proportionally Damped Model.

#	Λ	$\sqrt{\Lambda}/2\pi$	$\omega/2\pi$	error
1	5375.07	11.6684	(-0.427735,11.6606)	1.6e-6
2	108926	52.5275	(-8.66809,51.8074)	2.7e-6
3	219052	74.4893	(-17.4316,72.4209)	4.1e-7

8.2.2.2. QEP – Viscoelastically Damped

There are no verification tests yet for this solution.

8.2.2.3. QEP – Discrete Dampers

There are no verification tests yet for this solution.

8.2.3. SA_eigen

Verification of the SA_eigen solution is complicated by the model reduction inherent in the process. Kinsler[32] has a closed form expression for a coupled one dimensional structural acoustic system. The finite element solution will approach this solution as,

a the finite element mesh converges, and

b the modal truncation is eliminated.

Without both of these considerations, there will be no convergence of the solution. Unfortunately, while we can show a $1/h$ type convergence for the FE mesh, no such convergence can be expected for modal truncation. For some forms of basis functions the convergence will be rapid. In other cases, convergence may not be acceptable until the entire space has been spanned.

Because of model size issues, such convergence is demonstrated independently. Thus, we first show convergence of the mesh to the analytic solution. Then, with a coarse mesh, we demonstrate convergence as the number of modes in the basis is increased. Figure 8-57 shows the mesh convergence study. We note that for $1/H > 100$ the solution no longer appears to be converging. The *polyeig()* routine in MATLAB does a full factorization. Computing accurate modes with polyeig involves techniques that are beyond the scope of this document.

Figure 8-58 shows the convergence of the reduced model to the first coupled modal frequency when using 2 structural and 10 acoustic modes. Note that this mode converges from below to a value 1% higher than the mode of the solution without truncation.

Figure 8-59 shows the convergence of the modal frequency as the number of basis modes is increased. There is no damping for this system. Introducing radiation damping to the right side of the acoustic system impacts the modal convergence rate. As shown in Figure 8-60, radiation damping (or non-reflecting boundary conditions), delays convergence and degrades accuracy.

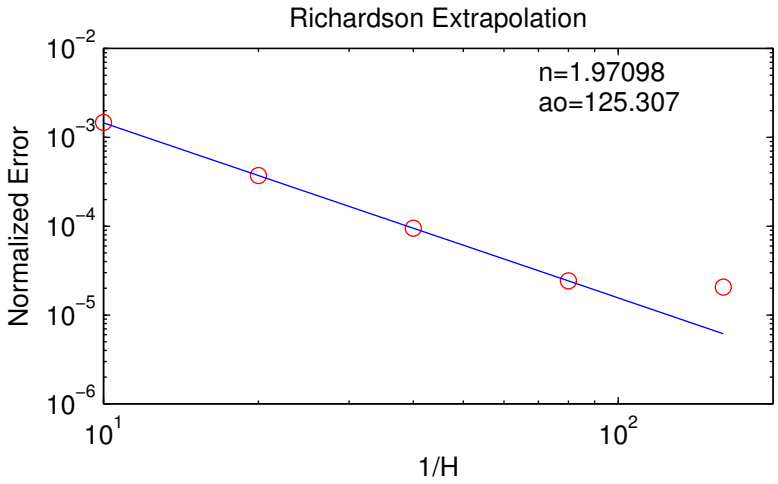


Figure 8-57. – Mesh convergence to 1D Structural Acoustics Example. The example, taken from Kinsler[32] uses $a = 1/25$ and $b = 8/3$, where a and b are defined in the reference. The eigen solution is found using MATLAB's `polyeig()` function. The analytical solution from equation 9.42a of Kinsler[32] is 125.2783.

To examine the dependence of this error on the coupling, we sweep through various structural mass quantities while holding all other parameters fixed. Sweeping the mass results in a change of structural resonant frequency. In addition, the type of coupling experienced by the acoustic cavity changes from approximately unbounded to fixed boundary conditions. Results shown in Figure 8-61, show variation as the parameter a of Kinsler is varied. The error is highest, and the coupling is greatest, when the structural and acoustic domains have similar resonant frequencies.

To examine the effects of impedance matching while maintaining the resonance frequencies, the structural mass and stiffness are varied together such that the resonance frequency is maintained at 160 Hz, below the acoustic resonance (166 Hz). Figure 8-62 provides the results. The error is largest when the impedance approximates an open acoustic termination.

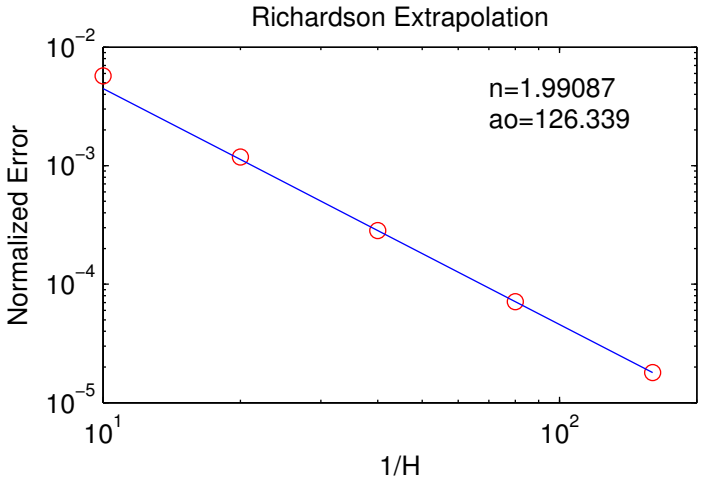


Figure 8-58. – Mesh convergence to 1D Structural Acoustics Example using a modal basis. The example is that of Figure 8-57. The quadratic eigen solution is computed using 2 structural and 10 acoustic modes in **Sierra/SD**.

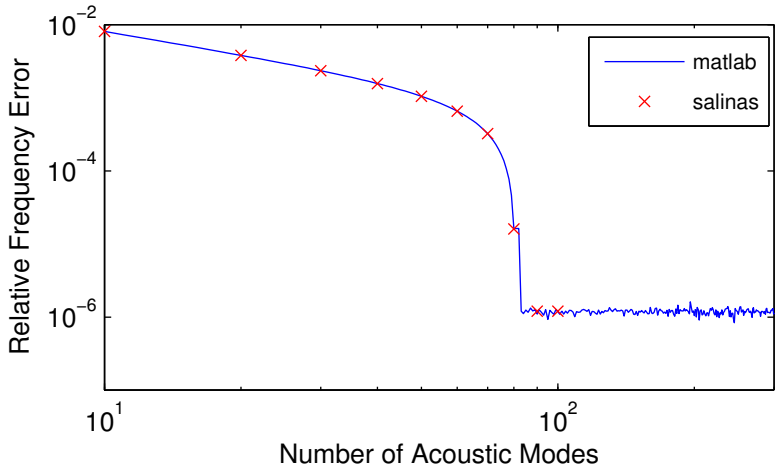


Figure 8-59. – Modal convergence to 1D Structural Acoustics Example using a modal basis. The example is that of Figure 8-57, with $1/h = 80$. The quadratic eigen solution is computed using 2 structural modes, while the number of acoustic modes varied. Computation is in MATLAB, with selective comparison to **Sierra/SD**. Convergence is not rapid as a solution requires components of all axial modes. After about 80 modes, no further improvement is obtained.

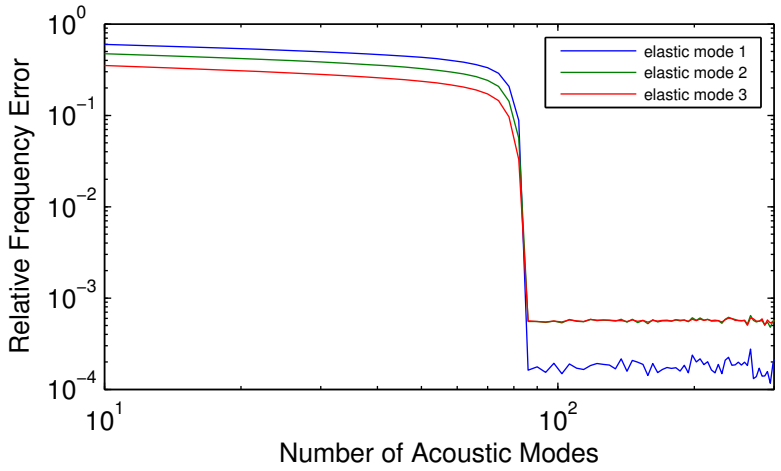


Figure 8-60. – Mesh convergence to *Damped* 1D Structural Acoustics Example using a modal basis. The model is unchanged from Figure 8-59 except that there is a non-reflecting boundary condition applied on the end opposite to the structure. MATLAB comparisons with `polyeig` truth model, with direct verification to **Sierra/SD** solution.

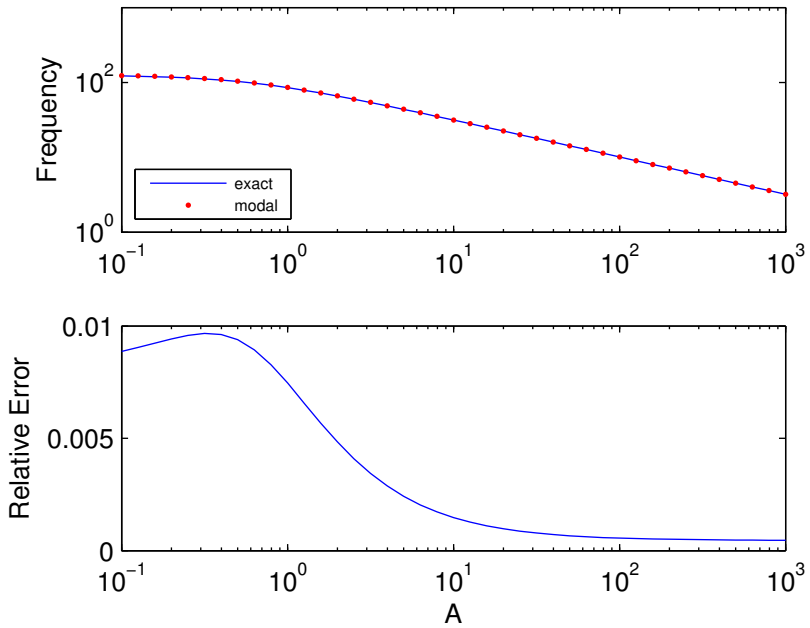


Figure 8-61. – Modal convergence of 1D Structural Acoustics Example using a modal basis. The example is that of Figure 8-57, with $h = 1/80$. The quadratic eigen solution is computed using 2 structural modes and 10 acoustic modes in **Sierra/SD**, while the mass parameter, a is varied.

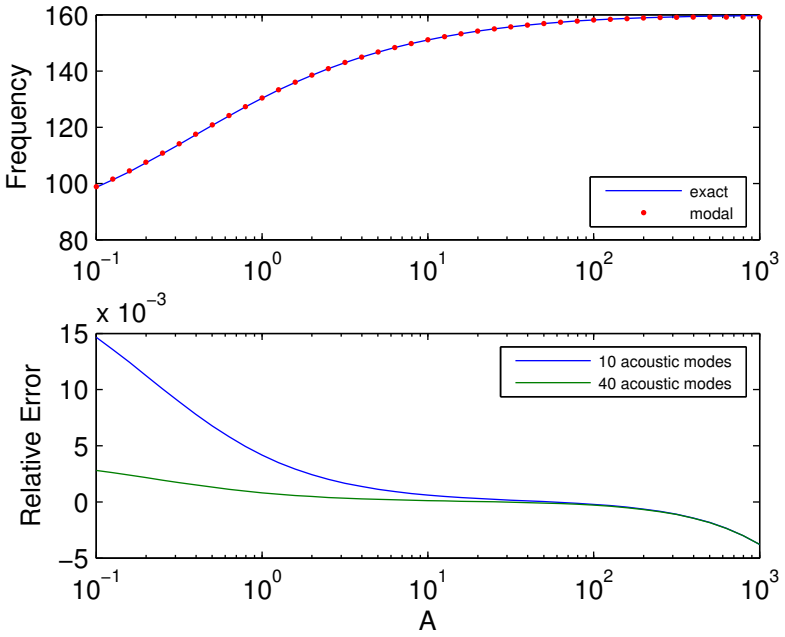


Figure 8-62. – Modal convergence of 1D Structural Acoustics Example using a modal basis as the impedance is swept. The example is that of Figure 8-57, with $h = 1/80$. The quadratic eigen solution is computed while both the mass parameter, a and the stiffness parameter, b, are varied. We maintain a structural resonance of 160 Hz.

8.2.4. Buckling of a Cantilever Beam

The buckling of a cantilever beam modeled using solid elements is verified. The geometry for this example consists of a cantilever beam with one end clamped, and with the other subjected to a compressive load P. Euler-Bernoulli beam theory predicts the critical buckling load to be

$$P_{cr} = \frac{2.4674EI}{L^2} \tag{8.2.14}$$

A simple mesh of this example was created, consisting of a $2 \times 2 \times 10$ hex elements. The critical buckling load is predicted to be

$$P_{cr} = \frac{2.4676 \times 30 \times 10^6 \times \frac{1}{12}}{10^2} = 61675 \tag{8.2.15}$$

The computed buckling load was 61370.1.

8.2.5. Verification With Respect to Semi-Analytical Eigen Analysis

Analytic and semi-analytic eigenvalue decompositions are known for most simple geometries. Summaries are available.[10] Note however, that these solutions are for idealized models. Euler's beam model is appropriate to thin beams where shear terms are negligible.

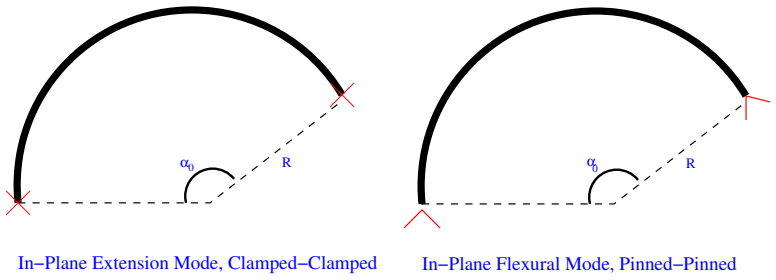


Figure 8-63. – Blevins Table 9-2.1 and 9-2.2 Geometries.

An eigenvector provides information about the global solution. A correct solution requires both a correct stiffness and mass matrix. Further, accuracy of the solution is easily determined by examination of the eigenvalues alone. On the other hand, the load vector is irrelevant, which simplifies the test matrix.

The "truth" model for these analyses are the eigenvalues obtained from analytic and semi-analytic solutions tabulated in *Blevins*. Note that the accuracy of the textbook solutions is limited to about 0.5% in most cases. Spring and Mass analysis matrix is detailed in Table 8-15. For beam elements, eigenvalues Tabulated in 8-16 through 8-18. Shell elements use Tables 8-19 through 8-21. Note that beams and shells have simplifying assumptions which may cause the solid based solutions to differ from the textbook solutions. For example, the "beams" built of solid elements will contain shear effects that are not present in a standard beam element. The geometry for these tests is illustrated in Figure 8-63.

The computational results represent the converged solution. In most cases a Richardson extrapolation has been performed to arrive at the minimum error due to discretization. Tables 8-15 through 8-21 use material properties for steel: $E = 3e7$ psi, $\nu = 0.30$ and $\rho = 0.288$ lbs/in^3 (7.4592x10^{-4} slugs/in^3).

Table 8-15. – Spring Mass Vibration.

Blevins Table	Description	Sol'n Type	Mode Number		
			1	2	3
6-2 2	two equal masses,	Exact	.0983632	.2575181	N/A
	two equal springs	FE	.0983632	.2575181	N/A
6-2 18	Three equal masses,	Exact	.159155	.3183100	.3183100
	six equal springs	FE	.159155	.3183100	.3183100

Note: The Lanczos solver (in ARPACK) cannot find all the modes of the system. Some modes were found by exporting the matrices and solving in MATLAB

Table 8-16. – Beam Mass Vibration.
Massless beam has square cross-section with $I_1 = 1$, L=20, 100 elements.

Table	Description	Sol'n	Mode 1
6-2 19	End mass on cantilever beam	Exact	16.88
		FE	16.88
6-2 20	Center mass, pinned-pinned beam	Exact	67.52
		FE	67.52
6-2 22	Center mass, clamped-clamped beam	Exact	135.05
		FE	135.05

Table 8-17. – Beam Vibration - Using Beam2.
The sample beam has a square cross-section with area=1, length=20. 100 elements. No torsion spring is yet available.

Blevins Table	Description	Sol'n Type	Mode Number			
			1	2	3	4
8-1 1	Free-free bending	Exact	515.36	1420.6	2785.0	4603.7
		FE	515.15	1419.6	2781.9	4596.9
8-1 2	Free-sliding bending	Exact	128.84	696.24	1719.3	3197.0
		FE	128.83	696.05	1718.5	3194.7
8-1 3	Clamped-free	Exact	80.99	507.56	1421.2	2784.9
		FE	80.98	507.44	1420.6	2783.2
8-1 5	Pinned-pinned	Exact	227.34	909.37	2046.1	3637.5
		FE	227.34	909.29	2045.7	3636.4

Table 8-18. – Uniform Shaft Torsional.
Note. The discrepancy in this table stems from a mismatch of geometry (which we intend to clear up soon). The analytic results are for circular cross-sections. We have a square cross-section in the FE results. It is clear that the frequencies should be ratios of 1,3,5,7, etc. This holds for the FE results.

Blevins Table	Description	Sol'n Type	Mode Number			
			1	2	3	4
8-19 2	Fixed-Free	analytic	1427.93	4283.78	7139.64	9995.5
		FE-Beam2	1554.68	4663.66	7771.49	10877.4
		FE-Hex8	1545.97	4642.1	7750.76	10880

Table 8-19. – Circular Arcs.

Blevins Table	Description	Sol'n Type	Mode Number			
			1	2	3	4
9-2 1	Extension Mode	analytic	52632	N/A	N/A	N/A
	Clamped-Clamped	FE Beam2	52693	N/A	N/A	N/A
9-2 2	In-Plane flexural mode	analytic	2579.35	13137.2	30989.4	56026.3
	Pinned-Pinned	FE Beam2	2587.73	13189.5	30671.7	54445.7
9-2 5	Out-of-Plane Flexural	analytic	1763.56	N/A	N/A	N/A
	Clamped-Clamped	FE Beam2	1741.11	N/A	N/A	N/A

Table 8-20. – Circular Plates - Bending.

Circular disk made of QuadT elements.

Blevins Table	Description	Sol'n Type	Mode Number			
			1	2	3	4
11-1.1	Free edge	Exact	126.84	219.35	295.32	495.50
		FE	129.31	217.25	300.16	493.72
11-1.2	Simply supported edge	Exact	120.18	336.61	619.37	718.61
		FE	119.20	335.69	618.69	718.64
11-1.3	Clamped edge	Exact	246.78	513.36	842.25	960.32
		FE	246.62	513.00	841.97	961.03
11-1.12	Clamped edge with point	Est.	25.98	N/A	N/A	N/A
	mass at center (M large)	FE	25.83	N/A	N/A	N/A

Table 8-21. – Rectangular Plates - Bending.

Using *Tria3* elements, aspect ratio $a/b = 1.5$ in all cases.

Blevins Table	Description	Sol'n Type	Mode Number			
			1	2	3	4
11-4.1	Free-free-free-free	Exact	864.14	927.25	2002.59	2158.85
		FE	862.61	919.15	1989.43	2142.13
11-4.21	Clamped-clamped-	Exact	2608.74	4029.22	6387.69	6428.04
	clamped-clamped.	FE	2608.29	4027.90	6387.04	6425.11
11-4.16	Simply supported	Exact	1377.13	2648.23	4237.00	4765.01
	(all 4 edges)	FE	1376.97	2648.01	4237.05	4766.57
11-4.6	Clamped-free-	Exact	652.94	1103.68	2127.08	2747.82
	simply supported-free	FE	648.82	1100.31	2113.90	2733.90

8.2.6. Code to Code Comparisons

Extreme care must be used when using code to code comparisons. They are no replacement for more rigorous verification techniques (see *Trucano*[51]). However, they may be useful when the following conditions are met.

- The "truth" model code has been adequately verified.
- The two codes can be determined to solve *exactly* the same differential equations.
- Comparisons are to the quantities to which the truth model code must ultimately converge.
- The value gained by the comparison provides important insight not readily obtained by solution of analytic problems.

Any method that provides additional examination of the application is valuable.

Some benchmark problems exist in the literature (see for example *MacNeal*[34]). Some benchmark problems will be solved using **Sierra/SD** and using MSC/NASTRAN, an industry standard for elastodynamics. Comparisons of the mesh-refined solutions will be made. Other codes may be used for other phenomena.

The list of such code to code comparisons will necessarily grow over time. An example includes a mock-AF&F which was analyzed for eigen response. This is a 500,000 degree of freedom model designed for optimization studies. It is a real design with the level of detail anticipated in practical models of this structure. It contains mostly Tet10 elements with shells constructed of Tria6. Much of the model was constructed using automatic mesh generation methods. Comparisons of the first 4 modes of this model are shown in Table 8-22.

Table 8-22. – AF&F code to code comparison.

#	Description	NASTRAN	Sierra/SD	Difference
1	Aft plate drum mode	434.3 Hz	437.0 Hz	.6%
2	First bending, X	627.4 Hz	629.1 Hz	.3%
3	First bending, Y	657.2 Hz	659.2 Hz	.3%
4	torsion	793.6 Hz	793.2 Hz	.05%

8.2.7. Direct Frequency Response

The **Sierra/SD** driver is verified using two mass spring systems problems. The first is a mass spring system with stiffness proportional damping, and the second is a mass spring system with mass proportional damping.

A diagram of the problem is available.[15] The exact solution to this problem[15] is

$$D_s = \frac{U}{U_0} = \frac{1}{\sqrt{\left((1 - r^2) + (2\zeta r)^2\right)}} \tag{8.2.16}$$

Here U is the displacement of the mass, U_0 is the magnitude of the forcing function, $r = \frac{\omega}{\omega_0}$ is the ratio of the circular frequency to the fundamental resonant frequency, and $\zeta = \frac{c}{2\sqrt{km}}$ is the level of damping. Damping is normalized with respect to the stiffness and mass of the spring mass system.

For proportional damping, we have $c = \alpha m + \beta k$. The exact solutions corresponding to equation 8.2.16 were computed and compared with simulations in **Sierra/SD** for two cases. In case 1, $\alpha = 0.0$ and $\beta = 1.0$. In case 2, $\alpha = 1.0$ and $\beta = 0.0$. Also, for convenience we set $k = m = U_0 = 1$ for this problem. In this way, the exact solutions for both mass and stiffness proportional damping were exactly the same.

Figure 8-64 shows the comparison of the computed and exact solutions for the case of stiffness proportional damping. The mass proportional damping case was exactly the same, and thus is not shown. We see that proportional damping decreases the peak of the resonant frequency, and shifts the frequency to the left. Excellent agreement is seen between **Sierra/SD** and the exact solutions.

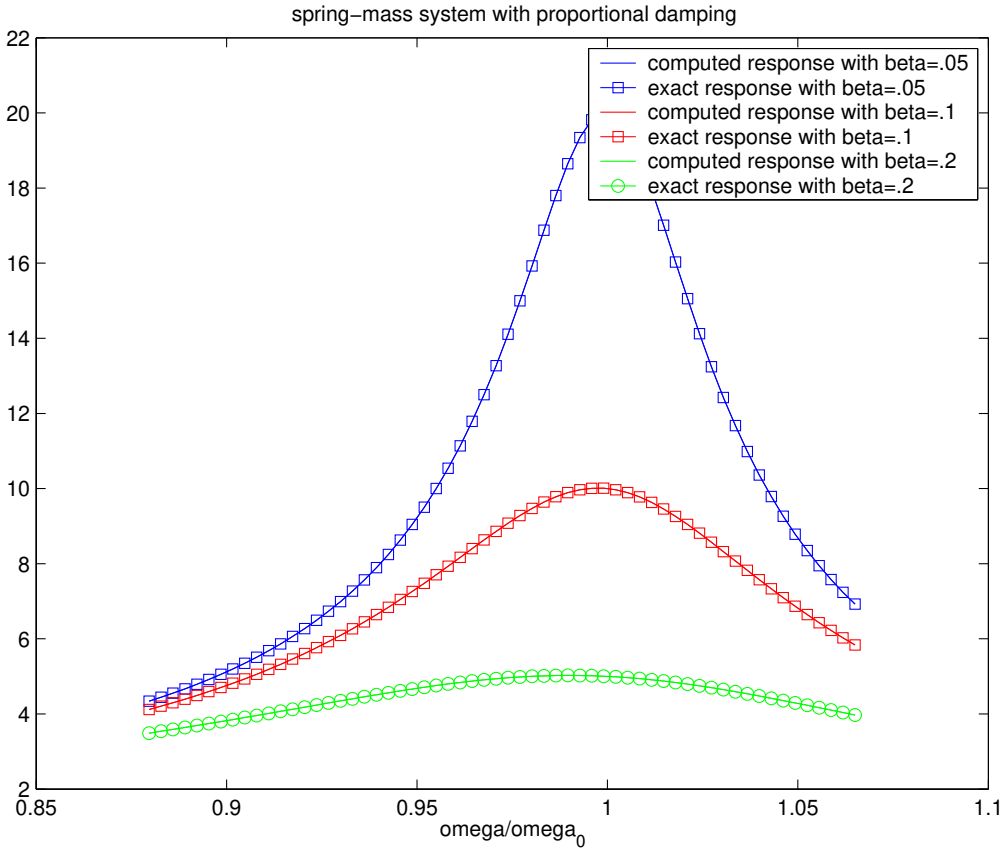


Figure 8-64. – Comparison of exact and computed responses from direct frequency response of a damped spring mass system.

8.2.8. Modal Frequency Response

Modal frequency response is verified against the corresponding direct frequency response analysis by assessing the convergence of the modal expansion. Test one concerns a free-free beam composed of 2 by 2 by 20 brick elements with a uniform pressure load is applied on both ends. The comparison of the results at a point in the center of the beam, versus the results from direct frequency response is given in Table 8-23.

Test two involves the same geometry as the previous test, and instead has one end fixed and the other subjected to a traction load of 111. Also, in this test, the modal acceleration method is used instead of modal frequency response. The results, compared with a direct solution, are given in Table 8-24. The modal frequency response results converge to the direct frequency response results as the number of modes in the modal expansion increases. We note that both of these tests are located in the **Sierra/SD** test suite under

`Salinas_test/verification/frf`

Table 8-23. – Convergence of Modal Frequency Response Method.

quantity	direct frf	modal			
		14 modes	30 modes	50 modes	100 modes
accx	12.7659	14.28	13.5	13.9	12.79
accy	-12.7659	-14.28	-13.5	-13.9	-12.79
accz	117.309	139.0	111.0	118.0	117.353

Table 8-24. – Convergence of Modal Acceleration Method.

quantity	direct frf	modal accel, 14 modes	modal accel, 30 modes
accx	-2350.82	-2349.75	-2350.81
accy	-2415.098	-2414.12	-2415.097
accz	-718.587	-718.321	-711.578

8.3. Mass Properties Verification Tests

Correction An old convention reversing the sign of the off-diagonal terms in the inertia tensor is used here.

Notation. A vector r is a column vector. Transpose is defined here for vectors over the real numbers, not the complex numbers. The transpose of r, denoted r^T is a row vector.

The inertia matrix is denoted by I. We need a new symbol for the identity matrix. Let's call it E.

A particle with position r and mass m is rotating about an axis passing through the origin has an inertia. Suppose that the axis is parallel to the unit vector n. The projection of a point r along the axis is $nn^T r$. In terms of $p = r - nn^T r$, The inertia is $mp^T p = mn^T(Er^T r - rr^T)n$. The new convention is to honor the minus sign in front of rr^T. The inertia of a system of particles r_i each of mass m_i is

$$I = \sum_i m_i (E r_i^T r_i - r_i r_i^T)$$

As the 3 vector a has a 3 by 3 spin matrix $[a]$, for any vector r, the cross product $a \times r = [a]r$. In terms of the spin matrix,

$$I = \sum_i m_i [r_i]^2.$$

The remainder of this note has not been updated to use the new sign convention.

The following problems were used to verify the mass properties calculations in **Sierra/SD**. These problems cover most element types, however superelements are not addressed here. The tests and results described here were generated with release 2.9.

8.3.1. 0D Verification Test

The following test was used to verify mass properties for conmass elements. The test consists of an assembly of three conmass elements as shown in Figure 8-65. In the finite element model, the masses were connected with RBar elements which do not add mass to the system.

The total mass of the assembly is $m_{total} = 3m$. The center-of-gravity is

$$x_{cg} = (mb + 0 - mb)/m_{total} = 0 \tag{8.3.1}$$
$$y_{cg} = (0 + mb + 0)/m_{total} = b/3 \tag{8.3.2}$$
$$z_{cg} = (0 + mb + 2mb)/m_{total} = 1 \tag{8.3.3}$$

The components of the inertia tensor are

$$I_{xx} = \bar{I}_{xx} + mr_x^2 \tag{8.3.4}$$
$$= \bar{I}_{xx} + m\left[(2b)^2 + (b^2 + b^2) + 0\right] = \bar{I}_{xx} + 6mb^2 \tag{8.3.5}$$
$$I_{yy} = \bar{I}_{yy} + mr_y^2 \tag{8.3.6}$$
$$= \bar{I}_{yy} + m\left[\left(b^2 + (2b)^2\right) + b^2 + b^2\right] = \bar{I}_{yy} + 7mb^2 \tag{8.3.7}$$
$$I_{zz} = \bar{I}_{zz} + mr_z^2 \tag{8.3.8}$$
$$= \bar{I}_{zz} + m\left[b^2 + b^2 + b^2\right] = \bar{I}_{zz} + 3mb^2 \tag{8.3.9}$$
$$I_{xy} = \bar{I}_{xy} + md_x d_y \tag{8.3.10}$$
$$= \bar{I}_{xy} + m\left[0 + 0 + 0\right] = \bar{I}_{xy} \tag{8.3.11}$$
$$I_{xz} = \bar{I}_{xz} + md_x d_z \tag{8.3.12}$$
$$= \bar{I}_{xz} + m\left[0 + 0 - 2b^2\right] = \bar{I}_{xz} - 2b^2 \tag{8.3.13}$$
$$I_{yz} = \bar{I}_{yz} + md_y d_z \tag{8.3.14}$$
$$= \bar{I}_{yz} + m\left[0 + b^2 + 0\right] = \bar{I}_{yz} + b^2 \tag{8.3.15}$$

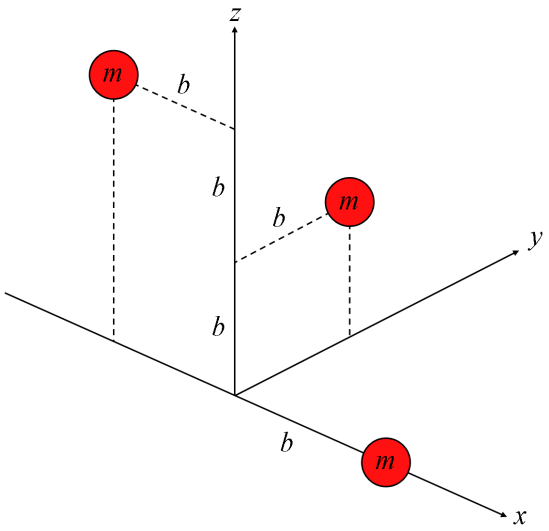


Figure 8-65. – Verification problem for conmass elements.

A comparison between these answers and the **Sierra/SD** predictions is shown in Table 8-25. Parameters used for this problem were $m = 1$, $b = 1$, and $\bar{I}_{xx} = \bar{I}_{yy} = \bar{I}_{zz} = \bar{I}_{xy} = \bar{I}_{xz} = \bar{I}_{yz} = 0$.

Table 8-25. – Comparison of **Sierra/SD** with exact solutions for the 0D verification problem.

Property	Exact	**Sierra/SD**
m_{total}	3.0	3.0
x_{cg}	0.0	0.0
y_{cg}	0.3333	0.3333
z_{cg}	1.0	1.0
I_{xx}	6.0	6.0
I_{yy}	7.0	7.0
I_{zz}	3.0	3.0
I_{xy}	0.0	0.0
I_{xz}	-2.0	-2.0
I_{yz}	1.0	1.0

8.3.2. 1D Verification Test

The following test was used to verify mass properties for the 1D elements which include the Beam2, TiBeam, Nbeam, and truss. This test case consists of a beam offset in all three dimensions from the coordinate frame as shown in Figure 8-66.

The total mass of the beam is

$$m_{total} = \rho V = \rho \pi r^2 l = 0.60kg \tag{8.3.16}$$

where V is the volume of the beam, r is the radius of the beam taken to be $5mm$, l is the length of the beam, and ρ is the beam material density taken as $2.8294 \times 10^{-5} kg/mm^3$ to give a total mass of $0.6kg$. The center-of-gravity is

$$x_{cg} = 180mm - \left(\frac{180mm + 90mm}{2} \right) = 45mm \tag{8.3.17}$$

$$y_{cg} = 150mm \tag{8.3.18}$$

$$z_{cg} = 90mm \tag{8.3.19}$$

Table 8-26. – Comparison of **Sierra/SD** with exact solutions for the 1D verification problem.

Property	Exact	Beam2	Nbeam	TiBeam	Truss
m_{total}	0.60	0.60	0.06	0.60	0.60
x_{cg}	45	45	45	44.875	45
y_{cg}	150	150	150	150	150
z_{cg}	90	90	90	90	90
I_{xx}	18367.5	18367.0	18367.0	18368.0	18360.0
I_{yy}	9723.75	9732.2	9733.7	9723.8	9720.0
I_{zz}	18363.75	18372.0	18374.0	18358	18360.0
I_{xy}	4050.0	4050.0	4050.0	4050.0	4050.0
I_{xz}	2430.0	2430.0	2430.0	2423.3	2430.0
I_{yz}	8100.0	8100.0	8100.0	8100.0	8100.0

The components of the inertia tensor are

$$I_{xx} = \bar{I}_{xx} + mr_x^2 \tag{8.3.20}$$

$$= \frac{1}{2}mr^2 + m(d_y^2 + d_z^2) = 18367.5kg \cdot mm^2 \tag{8.3.21}$$

$$I_{yy} = \bar{I}_{yy} + mr_y^2 \tag{8.3.22}$$

$$= \left[\frac{1}{4}mr^2 + \frac{1}{12}ml^2\right] + m(d_x^2 + d_z^2) = 9723.75kg \cdot mm^2 \tag{8.3.23}$$

$$I_{zz} = \bar{I}_{zz} + mr_z^2 \tag{8.3.24}$$

$$= \left[\frac{1}{4}mr^2 + \frac{1}{12}ml^2\right] + m(d_x^2 + d_y^2) = 18363.75kg \cdot mm^2 \tag{8.3.25}$$

$$I_{xy} = \bar{I}_{xy} + md_xd_y \tag{8.3.26}$$

$$= 0 + md_xd_y = 4050.0kg \cdot mm^2 \tag{8.3.27}$$

$$I_{xz} = \bar{I}_{xz} + md_xd_z \tag{8.3.28}$$

$$= 0 + md_xd_z = 2430.0kg \cdot mm^2 \tag{8.3.29}$$

$$I_{yz} = \bar{I}_{yz} + md_yd_z \tag{8.3.30}$$

$$= 0 + md_yd_z = 8100.0kg \cdot mm^2 \tag{8.3.31}$$

A comparison between these answers and the **Sierra/SD** predictions for the 1D elements is shown in Table 8-26. The finite element model used to generate these results contained 27 elements.

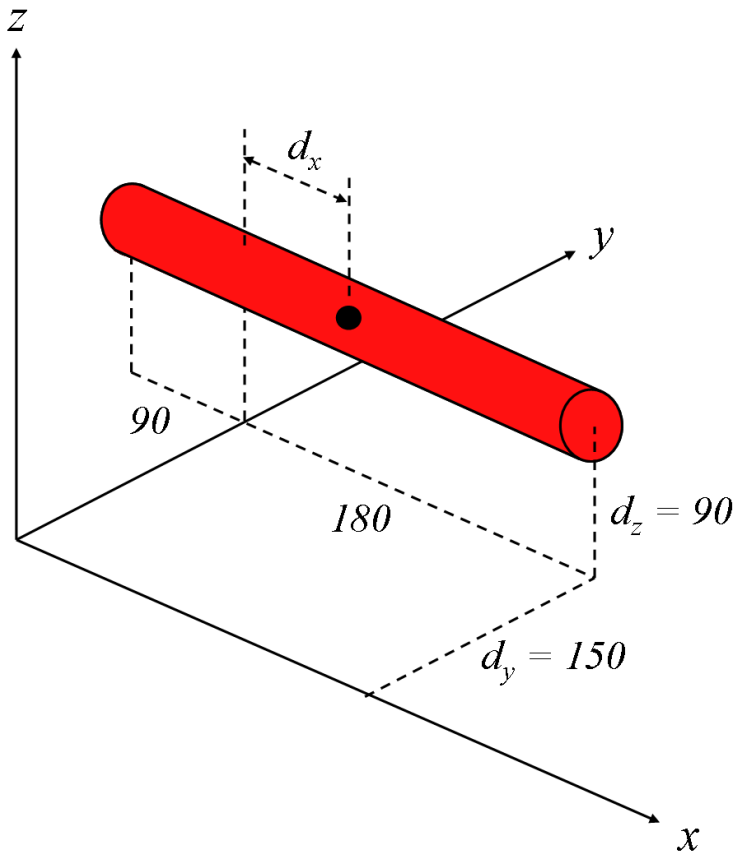


Figure 8-66. – Verification problem for 1D elements.

8.3.3. 2D Verification Test

The following test was used to verify mass properties for the 2D elements which include all the triangular and quadrilateral elements. This test case consists of an L-shaped plate as shown in Figure 8-67.

The total mass of the plate is

$$m_{total} = m_1 + m_2 = \rho \left(abt + \frac{1}{2} bct \right) \tag{8.3.32}$$

where m_1 and m_2 are the masses of the rectangular section and triangular section respectively. Both sections have the same material density, ρ, and the same thickness, t. The center-of-gravity is

$$x_{cg} = -\frac{1}{m_{total}} \left[m_1 a + m_2 \left(a + \frac{t}{2} \right) \right] \tag{8.3.33}$$

$$y_{cg} = \frac{1}{m_{total}} \left[m_1 \left(\frac{b}{2} \right) + m_2 \left(\frac{2}{3} b \right) \right] \tag{8.3.34}$$

$$z_{cg} = \frac{1}{m_{total}} \left[0 + m_2 \left(\frac{c}{3} \right) \right] \tag{8.3.35}$$

The components of the inertia tensor are

$$I_{xx} = \left(\bar{I}_{xx} + m_1 r_x^2\right) + \left(\bar{I}_{xx} + m_2 r_x^2\right) \tag{8.3.36}$$

$$= \left(\rho_1 t_1 \bar{I}_x + m_1 d_y^2\right) + \left[\left(\bar{I}_{yy} + \bar{I}_{zz}\right) + m_2\left(d_x^2 + d_y^2\right)\right] \tag{8.3.37}$$

$$= \left(\frac{m_1 b^2}{12} + \frac{m_1 b^2}{4}\right) + \left(\frac{m_2 c^2}{6} + \frac{m_2 b^2}{2}\right) \tag{8.3.38}$$

$$= \frac{m_1 b^2}{3} + \frac{m_2 c^2}{6} + \frac{m_2 b^2}{2} \tag{8.3.39}$$

$$I_{yy} = \left(\bar{I}_{yy} + m_1 r_y^2\right) + \left(\bar{I}_{yy} + m_2 r_y^2\right) \tag{8.3.40}$$

$$= \left(\rho_1 t_1 \bar{I}_x + m_1 d_x^2\right) + \left[\rho_2 t_2 \bar{I}_x + m_2\left(d_x^2 + d_z^2\right)\right] \tag{8.3.41}$$

$$= \left(\frac{m_1 a^2}{12} + \frac{m_1 a^2}{4}\right) + \left(\frac{m_2 c^2}{18} + \frac{m_2 c^2}{9} + m_2 a^2\right) \tag{8.3.42}$$

$$= \frac{m_1 a^2}{3} + \frac{m_2 c^2}{6} + m_2 a^2 \tag{8.3.43}$$

$$I_{zz} = \left(\bar{I}_{zz} + m_1 r_z^2\right) + \left(\bar{I}_{zz} + m_2 r_z^2\right) \tag{8.3.44}$$

$$= \left(\rho_1 t_1 \bar{I}_z + m_1 d_x^2\right) + \left[\left(\bar{I}_{xx} + \bar{I}_{yy}\right) + m_2\left(d_x^2 + d_y^2\right)\right] \tag{8.3.45}$$

$$= \left(\frac{m_1 a^2}{3} + \frac{m_1 b^2}{3}\right) + \left(\frac{m_2 b^2}{18} + \frac{8 m_2 b^2}{18} + m_2 a^2\right) \tag{8.3.46}$$

$$= \frac{m_1 a^2}{3} + \frac{m_1 b^2}{3} + \frac{m_2 b^2}{2} + m_2 a^2 \tag{8.3.47}$$

$$I_{xy} = \left(\bar{I}_{xy} + m_1 d_x d_y\right) + \left(\bar{I}_{xy} + m_2 d_x d_y\right) \tag{8.3.48}$$

$$= \left[0 + m_1\left(-\frac{a}{2}\right)\left(\frac{b}{2}\right)\right] + \left[0 + m_2\left(-a\right)\left(\frac{2b}{3}\right)\right] \tag{8.3.49}$$

$$= -\frac{m_1 a b}{4} - \frac{2 m_2 a b}{3} \tag{8.3.50}$$

$$I_{xz} = \left(\bar{I}_{xz} + m_1 d_x d_z\right) + \left(\bar{I}_{xz} + m_2 d_x d_z\right) \tag{8.3.51}$$

$$= (0 + 0) + \left[0 + m_2\left(-a\right)\left(\frac{c}{3}\right)\right] \tag{8.3.52}$$

$$= -\frac{m_2 a c}{3} \tag{8.3.53}$$

$$I_{yz} = \left(\bar{I}_{yz} + m_1 d_y d_z\right) + \rho_2 t_2 \int_0^b \int_0^{\frac{c}{b}y} yz\,dz\,dy \tag{8.3.54}$$

$$= (0 + 0) + \frac{\rho_2 t_2 c^2}{2b^2} \int_0^b y^3\,dy \tag{8.3.55}$$

$$= \frac{m_2 b c}{4} \tag{8.3.56}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 8-27. The

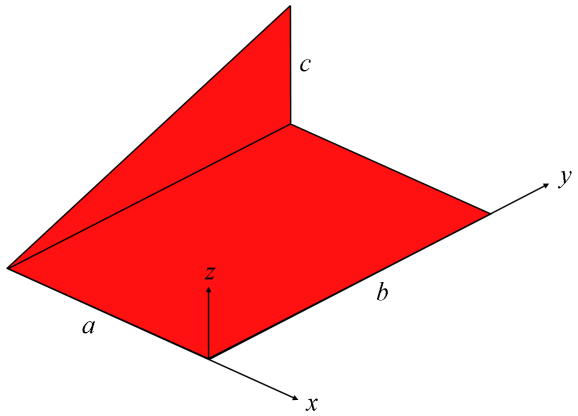


Figure 8-67. – Verification problem for shell elements.

Table 8-27. – Verification of 2D Mass Properties.

Property	Exact	Tri	Triashell	QuadTM
m_{total}	27.5	27.5	27.5	27.5
x_{cg}	-25.4682	-25.455	-25.455	-25.455
y_{cg}	27.2727	27.273	27.273	27.273
z_{cg}	2.7273	2.7273	2.7273	2.7273
I_{xx}	27167	27178	27167	27167
I_{yy}	23792	23801	23792	23792
I_{zz}	48708	48726	48708	48708
I_{xy}	-20000	-20000	-20000	-20000
I_{xz}	-3000	-3000	-3000	-3000
I_{yz}	2813	2812.4	2812.5	2812.5

finite element model of the plate contained 1679 elements. Parameters used for this problem were $a = 40in$, $b = 50in$, $c = 30in$, $t = 0.1in$, and $\rho = 0.1lb/in^3$.

8.3.4. 3D Verification Tests

The following tests were used to verify mass properties for the 3D elements which include the hexahedron, tetrahedron, and wedge elements. Solutions for these problems were mostly taken from the dynamics text by Meriam and Kraige.[35]

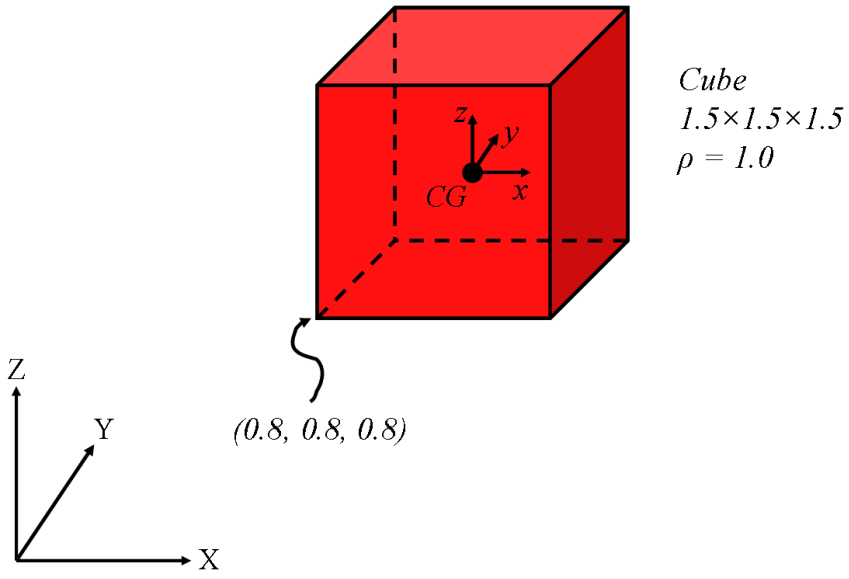


Figure 8-68. – Verification problem for solid elements.

8.3.4.1. Offset Block

The first 3D test consists of an offset cube as shown in Figure 8-68. The total mass of the block is given by

$$m_{total} = \rho l^3 = 3.375. \tag{8.3.57}$$

where ρ is the density of the block and l is the length of each side of the block. The center-of-gravity is

$$x_{cg} = y_{cg} = z_{cg} = 0.8 + \frac{1}{2}(1.5) = 1.55. \tag{8.3.58}$$

The components of the inertia tensor are

$$I_{xx} = \bar{I}_{xx} + mr_x^2 \tag{8.3.59}$$

$$= \frac{1}{12} m \left(2l^2\right) + m \left(d_y^2 + d_z^2\right) = 17.4825 \tag{8.3.60}$$

$$= I_{yy} = I_{zz} \tag{8.3.61}$$

$$I_{xy} = \bar{I}_{xy} + m d_x d_y = 8.1084375 \tag{8.3.62}$$

$$= I_{xz} = I_{yz} \tag{8.3.63}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 8-28. The tet model contained 26,430 elements, and the hex model contained 343 elements. Parameters used for this problem were $\rho = 1.0$ and $l = 1.5$

Table 8-28. – Comparison of **Sierra/SD** with exact solutions for the 3D block.

Property	Exact	Tet4	Hex8
m_{total}	3.375	3.375	3.375
x_{cg}	1.55	1.55	1.55
y_{cg}	1.55	1.55	1.55
z_{cg}	1.55	1.55	1.55
I_{xx}	17.4825	17.48	17.482
I_{yy}	17.4825	17.48	17.482
I_{zz}	17.4825	17.48	17.482
I_{xy}	8.1084	8.1084	8.1084
I_{xz}	8.1084	8.1084	8.1084
I_{yz}	8.1084	8.1084	8.1084

8.3.4.2. Half-torus

This test consists of a half-torus as shown in Figure 8-69. The total mass is

$$m_{total} = \rho V = \rho \pi r^2 \, (\pi R) = 0.61685. \tag{8.3.64}$$

where V is the volume of the body, and r and R are the radii as shown in the problem figure. The density, ρ, was taken as 1.0 in this non-dimensional problem. The center-of-gravity is

$$x_{cg} = y_{cg} = 0 \tag{8.3.65}$$

$$z_{cg} = \frac{r^2 + 4R^2}{2\pi R} = -0.64657. \tag{8.3.66}$$

The components of the inertia tensor are

$$I_{xx} = I_{zz} = \frac{1}{2} m R^2 + \frac{5}{8} m r^2 = 0.3474875 \tag{8.3.67}$$

$$I_{yy} = m R^2 + \frac{3}{4} m r^2 = 0.645765 \tag{8.3.68}$$

$$I_{xy} = I_{xz} = I_{yz} = 0. \tag{8.3.69}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 8-29. The tet model contained 175,592 elements. The hex model contained 62,300 elements.

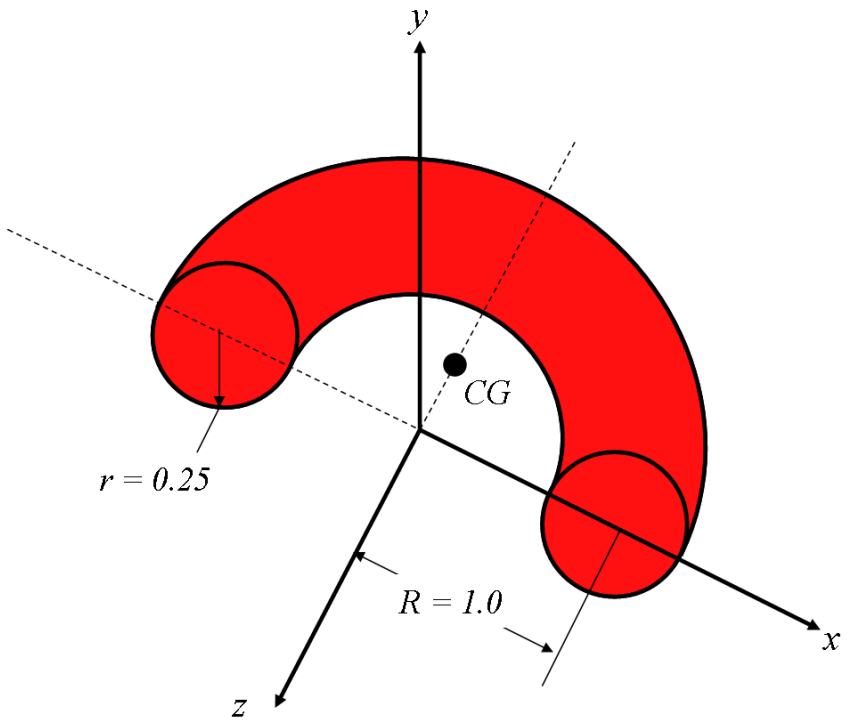


Figure 8-69. – Verification problem for solid elements.

Table 8-29. – Comparison of **Sierra/SD** with exact solutions for the 3D half-torus.

Property	Exact	Tet4	Hex8
m_{total}	0.61685	0.6153	0.61634
x_{cg}	0.0	0.0	0.0
y_{cg}	0.0	0.0	0.0
z_{cg}	-0.6466	-0.6465	-0.6465
I_{xx}	0.3475	0.3315	0.3321
I_{yy}	0.6458	0.6440	0.6451
I_{zz}	0.3475	0.3315	0.3321
I_{xy}	0.0	0.0	0.0
I_{xz}	0.0	0.0	0.0
I_{yz}	0.0	0.0	0.0

8.3.4.3. Hemispherical Shell

This test consists of a hemispherical shell as shown in Figure 8-70. The total mass is

$$m_{total} = \rho V = \frac{1}{2}\left[\frac{4}{3}\pi\left(r_o^2 - r_i^2\right)\right] = 0.318348. \tag{8.3.70}$$

where V is the volume of the body, and r_o and r_i are the outer and inner radii as shown in the problem figure. The density, ρ, was taken as 1.0 in this non-dimensional problem. The center-of-gravity is

$$x_{cg} = \frac{r}{2} = 0.25 \tag{8.3.71}$$

$$y_{cg} = z_{cg} = 0. \tag{8.3.72}$$

The components of the inertia tensor are

$$I_{xx} = I_{yy} = I_{zz} = \frac{2}{3}mr^2 = 0.053058 \tag{8.3.73}$$

$$I_{xy} = I_{xz} = I_{yz} = 0. \tag{8.3.74}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 8-30. The finite element model used to generate these results contained 108,000 hex elements.

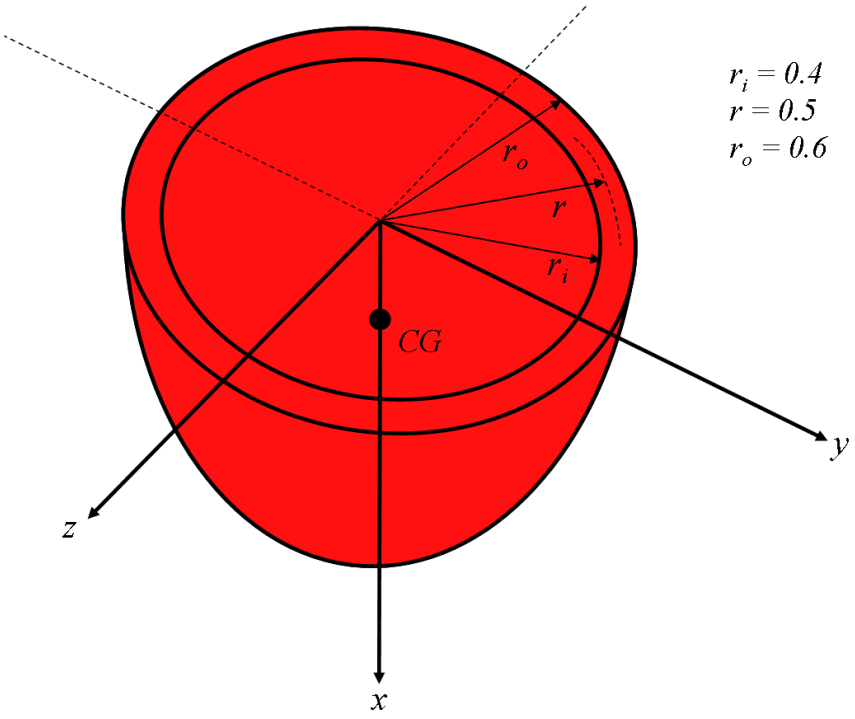


Figure 8-70. – Verification problem for solid elements.

Table 8-30. – Comparison of **Sierra/SD** with exact solutions for the 3D hemispherical shell.

Property	Exact	Hex8
m_{total}	0.3183	0.3182
x_{cg}	0.25	0.2566
y_{cg}	0.0	0.0
z_{cg}	0.0	0.0
I_{xx}	0.05306	0.05653
I_{yy}	0.05306	0.05653
I_{zz}	0.05306	0.05653
I_{xy}	0.0	0.0
I_{xz}	0.0	0.0
I_{yz}	0.0	0.0

8.3.4.4. Tetrahedron

This test consists of a tetrahedron with side lengths of a, b, and c as shown in Figure 8-71. The total mass is

$$m_{total} = \rho V = \rho \frac{1}{6} abc \tag{8.3.75}$$

where V is the volume of the tetrahedron. The density, ρ, was taken as 1.0 for this non-dimensional problem. The center-of-gravity is

$$x_{cg} = \frac{a}{4} \tag{8.3.76}$$

$$y_{cg} = \frac{b}{4} \tag{8.3.77}$$

$$z_{cg} = \frac{c}{4} \tag{8.3.78}$$

The components of the inertia tensor are

$$I_{xx} = \frac{1}{10}\left(b^2 + c^2\right) \tag{8.3.79}$$

$$I_{yy} = \frac{1}{10}\left(a^2 + c^2\right) \tag{8.3.80}$$

$$I_{zz} = \frac{1}{10}\left(a^2 + b^2\right) \tag{8.3.81}$$

$$I_{xy} = \int_m xy dm = \rho \int_V xy dV \tag{8.3.82}$$

$$= \int_0^a \int_0^{1-\frac{x}{a}} \int_0^{1-\frac{x}{a}-\frac{z}{c}} xy dy dz dx = \frac{1}{20} mab \tag{8.3.83}$$

$$I_{xz} = \int_m xz dm = \rho \int_V xz dV \tag{8.3.84}$$

$$= \int_0^a \int_0^{1-\frac{x}{a}} \int_0^{1-\frac{x}{a}-\frac{y}{b}} xz dz dy dx = \frac{1}{20} mac \tag{8.3.85}$$

$$I_{yz} = \int_m yz dm = \rho \int_V yz dV \tag{8.3.86}$$

$$= \int_0^b \int_0^{1-\frac{y}{b}} \int_0^{1-\frac{x}{a}-\frac{y}{b}} yz dz dx dy = \frac{1}{20} mbc \tag{8.3.87}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 8-31. The finite element model used for this problem used tet elements. Two different mesh densities were used and results for both are presented. The models contained 3933 elements and 26,650 elements respectively.

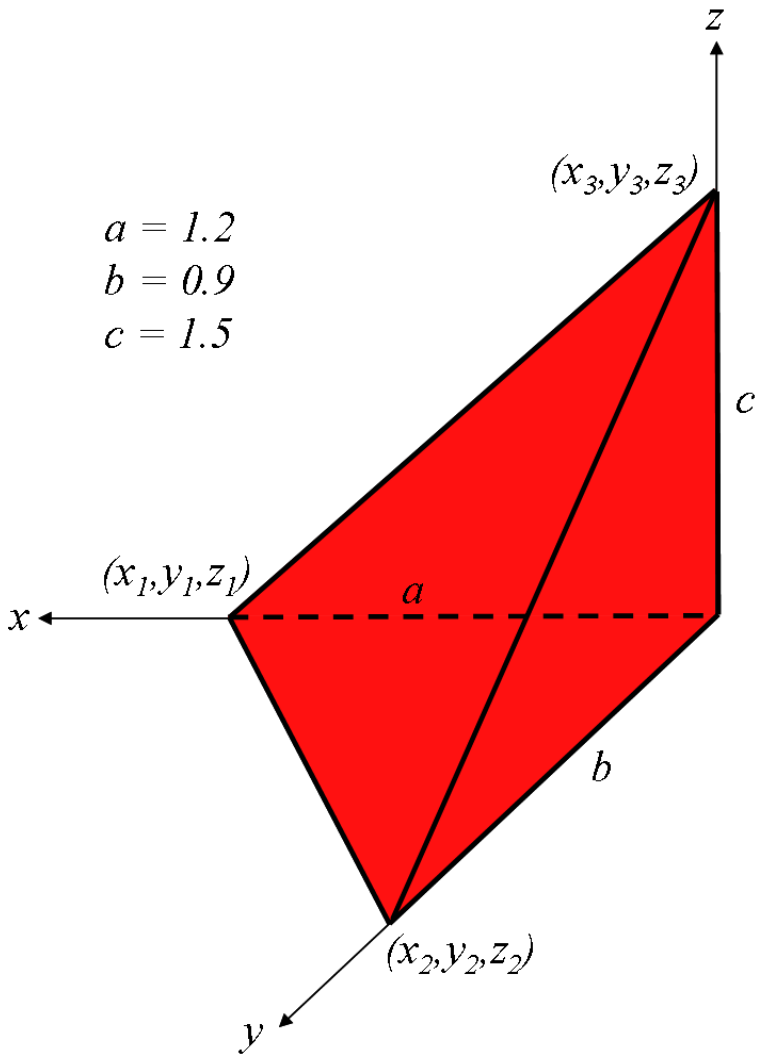


Figure 8-71. – Verification problem for solid elements.

Table 8-31. – Comparison of **Sierra/SD** with exact solutions for the 3D tetrahedron.

Property	Exact	Tet4 Coarse	Tet4 Fine
m_{total}	0.27	0.27	0.27
x_{cg}	0.3	0.3	0.3
y_{cg}	0.225	0.225	0.225
z_{cg}	0.375	0.375	0.375
I_{xx}	0.08262	0.08249	0.08262
I_{yy}	0.09963	0.09950	0.09963
I_{zz}	0.06075	0.06062	0.06075
I_{xy}	0.01458	0.01458	0.01458
I_{xz}	0.0243	0.02430	0.02430
I_{yz}	0.01823	0.01823	0.01823

8.4. Verification With Respect to Semi-Analytical Static Tests

Analytic and semi-analytic solutions for static deformation problems have been determined for many geometries and reported in *Roark*.[53] Note that these solutions are for idealized models. Thus, the beam models are appropriate to Euler Beams, but are exact for beams made of solid elements only in the limits where shear terms can be neglected.

Tables 8-32 and 8-33 will be used for Beam Elements: For Shell elements Tables 8-34 through 8-36 are used. The reference table is from *Roark*.[53]

Table 8-32. – Straight Beam Element Analytic Solutions.

Roark Table	**Description**	**Case**	**Max Disp**	**Max Rot.**
3 1a	cantilever free. Applied point force	*Roark*	-13.33	20.0
		Beam2	-13.33	20.0
		Tria3	-12.13	18.2
		Tria3⊥	-13.33	20.0
		Hex8	-13.44	N/A
3 1e	simply supported simply supported. Applied point force	*Roark*	.6356	2.311
		Beam2	.6356	2.312
		Tria3	.5783	2.104
		Tria3⊥	.5785	2.104
3 3b	cantilever guided. Applied point moment	*Roark*	4.032	-8.064
		Beam2	4.032	-8.064

The "Tria6⊥" model is rotated so a pure membrane deformation occurs. A finer mesh is required.

Table 8-33. – Curved Beam Element Analytic Solutions.

Roark Table	Description	Case	Dv
17 1	opposed radial loading on circular ring	*Roark* Beam2	-5.9513 -5.950
17 2	opposed in-line loading on circular ring (measured at $\theta = 30^o$)	*Roark* Beam2	.8263 .8259
17 3	opposed moments on circular ring (measured at $\theta = 30^o$)	*Roark* Beam2	7.9743 7.967

For solids, we employ Table 4-40. In addition, examples from the beams and shells may be computed using solid elements and a suitable discretization.

Table 8-34. – Annular Plate with Uniform Annular Line Load.
The test of having the outer edge simply supported and the inner edge free cannot be done at this time because the loading would require a non-cartesian coordinate system.

Roark Table	Description	Case	Max Disp
24 1a	Outer edge simply supported. Inner edge free	*Roark* Tria3	0.01701 0.01696
24 1b	Outer edge simply supported. Inner edge guided	*Roark* Tria3	.0068853 .006885
24 1e	Outer edge fixed. Inner edge free	*Roark* Tria3	.0034952 .0034946
24 5a	Outer edge simply supported. Inner free		

Table 8-35. – Square Plate.

Roark Table	Description	Case	Max Disp	Center Stress
26 1a	Simply supported.	*Roark*	5.3280	1.0346e7
	Uniform load over plate	Tria3	5.3225	1.03327e7
		QuadT	5.3225	1.03327e7
26 8a	Fixed edges.	*Roark*	1.6560	4.9896e6
	Uniform load over entire plate	Tria3	1.6590	4.9407e6
		QuadT	1.6590	4.9406e6

Table 8-36. – Thin Walled Pressure Vessels.
The second half of this table cannot be computed at this time because the pressure load would require using a non-cartesian coordinate system.

Roark Table	Description	Case	Max Disp	Max Stress	Comment
28 1a	uniform axial load on cylinder	*Roark*	-4.074e-6	407.4	ΔZ =2.037E-5
		Tria3	-4.626e-6	408.4	ΔZ =2.039E-5
		Hex8	-3.67e-6	408.0	ΔZ =2.057E-5
28 1b	uniform radial pressure on cylinder	*Roark*	3.333e-7	10.0	R=1, h=1.5, t=.1
		Tria3	3.333e-7	10.035	
		Hex8	3.445e-7	10.231	
28 3a	uniform pressure on sphere	*Roark*			
		Tria3			
28 5	uniform pressure on toroid	*Roark*			
		Tria3			
30 1a	uniform radial force on edge of partial sphere	*Roark*			
		Tria3			
30 1b	uniform edge moment on partial sphere	*Roark*			
		Tria3			

Table 8-37. – Solid Spheres.

Roark Table	Description	Exact Disp	FE Disp	Exact Stress	FE Stress
33 1A	Sphere on a flat plate				
33 1B	Sphere on a sphere				

8.5. Membranes and Transfer from Sierra/SM

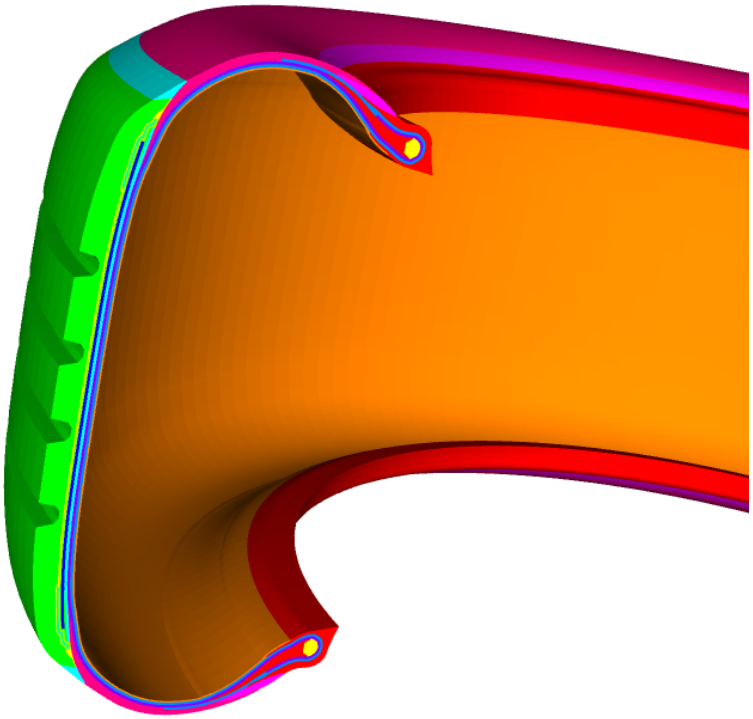


Figure 8-72. – Tire Analysis Model.

In this case, analyses of a preloaded (inflated) tire from Sierra are compared to ABAQUS^{TM}. The tire model (Figure 8-72) consists of a rim, and multiple layers of rubber and membranes. The tire is preloaded using **Sierra/SM**. The **Sierra/SD** analysis in this test case involves reading the results from that SM analysis, transferring material parameters, and computation of the eigenvalues of the system.

Eigenvalue results are shown in Table 8-38. As seen in the table, there is excellent agreement between ABAQUS and **Sierra/SD** for this problem.

#	ABAQUS	Sierra	% difference
1	39.912	40.3718	1.1
2	53.586	51.3133	4.3
3	55.650	53.5655	3.8
5	75.071	73.3562	2.3
7	97.202	96.6323	0.6
9	98.984	98.6028	0.4
11	119.35	119.045	0.3
13	142.54	142.219	0.2
15	142.56	142.287	0.2
17	167.07	166.891	0.1
19	171.37	171.045	0.2
21	193.59	193.372	0.1
23	193.75	193.540	0.1
25	214.47	214.001	0.2
27	221.77	221.814	0.0
29	235.20	234.640	0.2

Table 8-38. – Comparison of Eigen Frequencies of the Mooney-Rivlin Inflated Tire. For double modes one frequency is listed.

9. INPUT DECKS FOR VERIFICATION TESTS

The files required to run most verification problems are available from the test repositories. On Sandia systems these can be accessed at

```
#The majority of verification tests
/projects/sierra/tests/Salinas_rtest/verification_manual

#Tests involving thermal/fluid coupling with Aria or Fuego
/projects/sierra/tests/arpeggio_rtest/salinas_verification

#Tests involving Sierra SD/SM transfer
/projects/sierra/tests/sd_sm_coupled_rtest/verification_manual
```

9.1. Craig Bampton Reduction

Refer to Section 2.1

```
SOLUTION
    solver=gdsw
    eigen nmodes=all
    shift = -1e6
END

FILE
    geometry_file  'system_plus_se.exo'
    omit block 2
END

BOUNDARY
   sideset 1
       fixed
END

LOADS
END

ECHO
END

OUTPUTS
    disp
END

MATERIAL 'steel'
    E 30e6
    nu 0.3
    density 0.288
END
```

```
BLOCK 1,2
    material 'steel'
END

BLOCK 3
    superelement
    sensitivity_param 1 0.28800
    // this will use taylor series expansion to get the matrices
    file = 'onehex_super.ncf'
    map locations
END
```

9.2. Beam CBR

Refer to Section 2.2 for details of the test.

```
SOLUTION
cbr
        nmodes=90
title 'single beam model. 100 elements. xy only'
END

FILE
geometry_file 'beam100b.exo'
END

cbmodel
   file=beamcbr.ncf
   format=netcdf
   nodeset 1
end

BOUNDARY
    nodeset 3
      y = 0
      z = 0
      rotx = 0
      roty = 0
      rotz = 0
END

LOADS
END

OUTPUTS
     deform
END

ECHO
END



BLOCK 1
material 'Aluminum'
Beam2
Area 0.1
orientation 0 .1 0
I1 .2
I2 .3
J .5
END

Material 'Aluminum'
E 10.0E6
nu 0.33
density 253.82e-6
END
```

9.3. Waterline of a ship

Refer to Section 2.5

```
SOLUTION
    solver=gdsw
  title=' uhwm_20150113'
  waterline
  max_iterations 100
  vizoption = ensight
  tolerance_force   1e-10
  delta 1e-8
  point_a 2479.9 0. 100
  point_b 3479.9 0. 100
  point_c 2479.9 1000. 100
  load 1
END

FILE
   geometry_file 'uhwm_20150113.exo'
end

PARAMETERS
   eigen_norm=visualization
end

BOUNDARY
end

LOAD 1
  sideset 50000001
    pressure = 1
    function = 1
  body
    gravity = 0 0 -980.0
end
LOADS
end
FUNCTION 1
  name 'pressure versus depth'
  type LINEAR
  data 0.0 0.0
  data 1000.0 980e3
end
OUTPUTS
      force // applied forces
      npressure
end
HISTORY
    nodeset '500000011,'
    disp
    velocity
    acceleration
    nskip 1
end
{include(uhwmBlocks)}#
{include(uhwmMaterials)}#
```

9.4. Transient Convergence

Refer to Section 2.6

```
SOLUTION
    solver=gdsw
  transient
  nsteps  4500
  time_step  1.0e-3
END

FILE
  geometry_file 'beam.exo'
END

OUTPUTS
END

BOUNDARY
  nodeset 1
  fixed
END

BLOCK 1
  Beam2
  material=1
  Area=0.03
  I1=0.09
  I2=0.01
  J=0.1
  orientation = 1 1 0
END

MATERIAL 1
  Isotropic
  E = 1e+07
  NU = 0.3
  density = 1
END

LOADS
  nodeset 2
  force 0 1 0
  function 1
END

ECHO
END

HISTORY
  nodeset '2'
  disp
END

FUNCTION 1
  type LINEAR
  name "test_func1"
  data 0 0
  data 1 1
  data 2 0
END
```

9.5. Modal Transient Temporal Convergence

Refer to Section 2.7

```
SOLUTION
    solver=gdsw
  case dummyEigen
    eigen
      shift=-1
      nmodes 3
  case out
    modaltransient
      nsteps  4500
      time_step  1.0e-3
      nmodes 3
END

FILE
  geometry_file 'beam.exo'
end

OUTPUTS
  disp
end

BOUNDARY
  nodeset 1
  fixed
end

BLOCK 1
  Beam2
  material=1
  Area=0.03
  I1=0.09
  I2=0.01
  J=0.1
  orientation = 1 1 0
end

MATERIAL 1
  Isotropic
  E = 1e+07
  NU = 0.3
  density = 1
end

LOADS
  nodeset 2
  force 0 1 0
  function 1
end

HISTORY
  nodeset '2'
  disp
end

FUNCTION 1
  type LINEAR
  name "test_func1"
  data 0 0
  data 1 1
  data 2 0
end
```

9.6. Modal Transient

Refer to Section 2.11

```
SOLUTION
    solver=gdsw
  Case rigid
    geometric_rigid_body_modes
  Case flexible
   restart=none
   eigen nmodes=10
   shift=-1.e8
  Case filter1
   modalfilterCase
   modalfilter norot
  Case one
   modaltransient
   time_step=0.001
   nsteps=400
   load=1
  Case four
   modaltransient
   time_step=0.001
   nsteps=2000
   load=4
END
Modalfilter norot
   add all
   remove 4,5,6,9
End

Modalfilter norbm
   add 7:10
End
History
    block 101
    disp
    acceleration
End
Parameters
 wtmass=0.00259
 num_rigid_mode 6
End
File
   geometry_file 'temp1/modaltransver.exo'
End
Load=1
  nodeset 12
    force = 0 0 1
    scale 1.0e5
    function 8
End
Outputs
  disp
End
{include(blocks_and_materials.inp)}#
{include(functions.inp)}
Load=4
  nodeset 12
    force = 0 0 1
    scale 1.0e5
    function 2
End
```

9.7. Transient Restart Examples

9.7.1. Linear Transient in Step 1

Refer to Section 2.8 for output of the tests.

```
SOLUTION
    solver=gdsw
  case eig
    eigen
    nmodes 40
    shift = -1e6
    restart=auto
  case out
    transient
    time_step 1.0e-4
    nsteps    30
    nskip=1
    restart=WRITE
    load=1
END
FILE
  geometry_file  ninjabot.exo
end
BOUNDARY
end
LOAD 1
  sideset 28 pressure 100 function 1
  sideset 30 pressure 100 function 1
end
Function 1
  type Linear
  name "test_func1"
    data 0 0
    data 1 1
end
Echo
mass
end

History
  nskip 1
  sideset 28
  disp
end
Outputs
  disp
  velocity
end

Tied Joint
  Normal Definition = slip
  surface 13 14
  search tolerance = 1e-6
  connect to block 50
end
  Tied Joint
  Normal Definition = slip
  surface 16 17
  search tolerance = 1e-6
  connect to Block 51
end
RigidSet set1
  sideset 30
  sideset 31
  sideset 32
```

```
end
Tied Data
  surface 20, 21
  search tolerance =1e-3
end
Tied Data
  surface 22, 23
  search tolerance =1e-3
end
Tied Data
  surface 24, 26
  search tolerance =1e-3
end
Tied Data
  surface 25, 27
  search tolerance =1e-3
end
Tied Data
  surface 28, 29
  search tolerance =1e-3
end
Tied Data
  surface 40, 41
  search tolerance =1e-3
end
Tied Data
  surface 42, 43
  search tolerance =1e-3
end
Tied Data
  surface 42, 44
  search tolerance =1e-3
end
MATERIAL "steel"
  E 30e6
  nu 0.3
  density 0.288
end
MATERIAL 'dead'
  isotropic
  E = 10
  nu = 0.29
  density = 0
end

MATERIAL 'aluminum'
  isotropic
  E = 10e6
  nu = 0.45
  density = 0.27
end

Block 1 2 3 4 5 6 7 8 9 10
  material 'aluminum'
end

Block 11 12 15
  material "steel"
end

Block 16
  RBAR
end

Block 17
  RBAR
end
```

```
Block 18
  conmass
  MASS = 100
end

Block 19
  conmass
  MASS = 100
end

Block 20
  superelement
  file = 'super_sword.ncf'
  map locations
end

Block 21
  superelement
  file = 'super_shield.ncf'
  map locations
end

Block 50
  Joint2g
  kx = iwan 1
  ky = iwan 1
  krz = elastic 1.0e9
end

Block 51
  Joint2g
  kx = iwan 1
  ky = iwan 1
  krz = elastic 1.0e9
end

Property 1
    chi -.82
    phi_max = 1.75e-4
    R = 5.5050e+6
    S = 2.1097e+6
end
```

9.7.2. Restarted Modal Transient in Step 2

```
SOLUTION
    solver=gdsw
  case eig
    eigen
    nmodes 40
    shift = -1e6
    restart=auto
  case out
    modaltransient
    time_step 1.0e-4
    nsteps    40
    nskip=1
    restart=READ
    load=1
END
FILE
  geometry_file  ninjabot.exo
end
BOUNDARY
end
LOAD 1
```

```
  sideset 28 pressure 100 function 1
  sideset 30 pressure 100 function 1
end
Function 1
  type Linear
  name "test_func1"
    data 0 0
    data 1 1
end
Echo
mass
end

History
  nskip 1
  sideset 28
  disp
end
Outputs
  disp
  velocity
end

Tied Joint
  Normal Definition = slip
  surface 13 14
  search tolerance = 1e-6
  connect to block 50
end
  Tied Joint
  Normal Definition = slip
  surface 16 17
  search tolerance = 1e-6
  connect to Block 51
end
RigidSet set1
  sideset 30
  sideset 31
  sideset 32
end
Tied Data
  surface 20, 21
  search tolerance =1e-3
end
Tied Data
  surface 22, 23
  search tolerance =1e-3
end
Tied Data
  surface 24, 26
  search tolerance =1e-3
end
Tied Data
  surface 25, 27
  search tolerance =1e-3
end
Tied Data
  surface 28, 29
  search tolerance =1e-3
end
Tied Data
  surface 40, 41
  search tolerance =1e-3
end
Tied Data
  surface 42, 43
  search tolerance =1e-3
end
```

```
Tied Data
  surface 42, 44
  search tolerance =1e-3
end
MATERIAL "steel"
  E 30e6
  nu 0.3
  density 0.288
end
MATERIAL 'dead'
  isotropic
  E = 10
  nu = 0.29
  density = 0
end

MATERIAL 'aluminum'
  isotropic
  E = 10e6
  nu = 0.45
  density = 0.27
end

Block 1 2 3 4 5 6 7 8 9 10
  material 'aluminum'
end

Block 11 12 15
  material "steel"
end

Block 16
  RBAR
end

Block 17
  RBAR
end

Block 18
  conmass
  MASS = 100
end

Block 19
  conmass
  MASS = 100
end

Block 20
  superelement
  file = 'super_sword.ncf'
  map locations
end

Block 21
  superelement
  file = 'super_shield.ncf'
  map locations
end

Block 50
  Joint2g
  kx = iwan 1
  ky = iwan 1
  krz = elastic 1.0e9
end
```

```
Block 51
  Joint2g
  kx = iwan 1
  ky = iwan 1
  krz = elastic 1.0e9
end

Property 1
    chi -.82
    phi_max = 1.75e-4
    R = 5.5050e+6
    S = 2.1097e+6
end
```

9.8. Fluid Structure Interaction Added Mass

Refer to Section 2.12

```
SOLUTION
    solver=gdsw
  case 'qevp'
    qevp
       method=sa_eigen
       nmodes = 400
       nmodes_acoustic=100
       nmodes_structure=100
       shift = -1.e+5
       sort method= magnitude
       reorthogonalize=yes
      check_diag=yes
END

FILE
 //  geometry_file addedmass_shell_0.01_sphere.exo
   geometry_file temp1/addedmass_shell_0_01_sphere.par
END

BOUNDARY
   sideset 1
   fixed
nodeset 1
fixed
   nodeset 10
      y=0 x=0 rotx=0 roty=0 rotz=0
END

OUTPUTS
        disp
END

ECHO
disp
mass = block
END

BLOCK 1
material "steel"
quadT
thickness = 0.1
        membrane_factor 0.0005
END

BLOCK 3
        material "water"
END

BLOCK 4
material "water"
END

BLOCK 5
material "water"
END

BLOCK 2
// name "spring"
// Coordinate 1
         spring
 kx=10000
         ky=10000
```

```
         kz=10000
END

//TIED DATA
//  surface 3,2
//  search tolerance = 1e-1
//END

MATERIAL "steel"
E 19.5e9
nu .3
density 7700.0
END

MATERIAL "water"
density 1000
        acoustic
        c0 1500
END

GDSW
  solver_tol 1e-11
  overlap = 3
  prt_memory yes
  prt_timing yes
END
```

9.9. Fluid Structure Cavitation

Refer to Section 2.13

```
SOLUTION
    solver=gdsw
     transient
     time_step = 1.313e-5
     nsteps 1200
     rho = 0.8
     scattering
END

FILE
  geometry_file 'Plate_fluid_shell.exo'
END

damping
  beta = 1.5e-5
end

Frequency
  freq_min = 1.0
  freq_step = 2.0
  freq_max = 80.0
  apressure
  block 1
End

OUTPUTS
apressure
velocity
END


HISTORY
    block all
    apressure
    velocity
    nskip  10
END


ECHO
END

BOUNDARY
  sideset 10
    infinite_element
    use block 111
sideset 5
        x=0 z=0 rotx=0 rotz=0
sideset 4
        x=0 z=0 rotx=0 rotz=0
sideset 2
        x=0 z=0 rotx=0 rotz=0
sideset 3
        x=0 z=0 rotx=0 rotz=0
END

TIED DATA
   Surface 1, 6
    search tolerance =1e-3
END

FUNCTION 3
```

```
  type planar_step_wave
  direction 0 1 0
  material "water"
  origin 0 149 0
  beta = 1.0042e3
END

LOADS
    sideset 1
       acoustic_vel = 103
       function = 3
    sideset 6
       pressure = 103
       function = 3
END

BLOCK 1
     quadT
        material "Steel"
        thickness = 1.0
END

BLOCK 2
        material "water"
END

Block 111
       infinite_element
       radial_poly legendre
       order 3
       ellipsoid_dimensions 20000 20000  20000
       source_origin = 0 19850 0
END

MATERIAL "Steel"
        E 30e6
        nu 0.3
density 5.32986e-4
//        density 0.288
END

MATERIAL "water"
        density 9.3455e-5
        c0 57120
        acoustic
END

GDSW
        solver_tol = 1e-8
END
```

9.10. Buckling of Constant Pressure Ring Input

Refer to Section 2.14 for details of the test.

```
SOLUTION
    solver=gdsw
 buckling
  bucklingSolver = {ARPACK_MODE}
  nmodes 1
  shift=-100
END

FILE
  geometry_file 'temp1/ring20.par'
END

BOUNDARY
nodeset 1
  y=0
nodeset 2
  x=0
nodeset 3
  z=0
END

LOADS
  sideset 1
    pressure = 1.0
END

OUTPUTS
  deform
END

ECHO
END

BLOCK 1
  material 1
END

Material 1
  E 10e6
  nu 0.0
  density 0.098 // not used in statics
END
```

9.11. Buckling of Cantilever Beam Input

Refer to Section 2.15 for details of the test.

```
SOLUTION
    solver=gdsw
    buckling
    nmodes 4
    shift=-1.e5
END

FILE
  geometry_file 'bar.exo'
// geometry_file 'bar.exo'
END

OUTPUTS
    deform
END

ECHO
END

BOUNDARY
  sideset 1
   fixed
END


BLOCK 1
        material "steel"
END

MATERIAL "steel"
density 1.293
        E 3.0e7
        nu 0.0
END

LOADS
  sideset 2
  pressure=1.0
END
```

9.12. Eigenvalue Restart with Virtual Nodes and Elements

Refer to Section 2.16

```
SOLUTION
  case grbm
  geometric_rigid_body_modes
  case eig
  eigen
  shift = -1e5
  restart=write
END
FILE
  geometry_file ninja_SE_IE_TJ.exo
  omit block 51,52,53,54,60,61,62,100
END

LOADS
  sideset 101
  pressure 1.0
  function 1
END

PARAMETERS
  num_rigid_mode 6
  eig_tol = 1.0e-16
END

FUNCTION 1
  type LINEAR
  name "test_func1"
  data 0 0
  data 1 1
END

OUTPUTS
  disp
END

RIGIDSET set1
  sideset 30
  sideset 31
  sideset 32
END

PROPERTY 1
  chi -.82
  phi_max = 1.75e-4
  R = 5.5050e+6
  S = 2.1097e+6
END

GDSW
  solver_tol 1.0e-12
END

{include(blocks_and_materials.inp)}
{include(tied_data_and_joints.inp)}
```

9.13. Rotating rigid bodies: Statics

Refer to Section 2.17.1

```
SOLUTION
    solver=gdsw
     statics
END
loads
     body
       angular_velocity 0 0 1.1
end
file
    geometry_file 'dumbbell.exo'
end
boundary
    nodeset 1 fixed
end
outputs
  force
end
echo
    mass
end
block 2
    conmass
    mass=2
end
block 1
    beam2
    material=light
    area=1e-2
    i1=1e-2
    i2=1e-2
    j=2e-2
    orientation 0 0 1
end
material light
    isotropic
    density = 0
    nu = .3
    E = 1e7
end
```

9.14. Rotating rigid bodies: Transient

Refer to Section 2.17.3

```
solution
     title 'pt hex starting at rest in rotating frame'
     transient
        time_step = 0.001
        nsteps = 1000
end

loads
     body
       angular_velocity 0 0 1.1
       function 1
end

file
    geometry_file 'phex_at_rest.exo'
end

function 1
   type=linear
   data 0 1
   data 1 1
end

boundary
end


outputs
  disp
  force
end

echo
    mass
end

block 1
    material heavy
    rotational_type lagrangian
end

material heavy
    isotropic
    E = 30.0e6
    nu = 0.3
    density = 10
end
```

9.15. Angular Velocity, Beam, Statics

Refer to Section 2.17.7

```
SOLUTION
    solver=gdsw
   statics
END

FILE
geometry_file 'beam_beam.exo'
END

LOADS
    body
      angular_velocity 0 0 5
      function = 1
END

FUNCTION 1
       type LINEAR
       name "test_func1"
       data 0.0 1.0
       data 1.0 1.0
       data 2.0e4 1.0
END

//PARAMETERS
// wtmass=0.00259
//END

OUTPUTS
disp
force
END

ECHO
mass
mass=block
END

BOUNDARY
   nodeset 1
       fixed
END

BLOCK 1
    beam2
    material="steel"
    area=1e-2
    i1=1e-2
    i2=1e-2
    j=2e-2
    orientation 0 0 1
END

MATERIAL "steel"
        E 19.5e10
        nu 0
        density 7700.0
END
```

9.16. Angular Acceleration, Statics, BEAM2

Refer to Section 2.17.8

```
$ Algebraic Preprocessor (Aprepro) version 6.25 (2023/10/12)
SOLUTION
statics
END

parameters
end

FILE
geometry_file beam40x.g
END

BOUNDARY
nodeset baseNode
fixed
END

LOADS
body
angular_acceleration = 0 0 100000
coordinate rotz
nodeset loadNode
force = 10000 0 0

END

OUTPUTS
mfile
disp
END

HISTORY
  disp
  node dispy at node 2 as atEndpointy
  node dispy at node 22 as atMidpointy
  node dispx at node 2 as atEndpointx
  node dispx at node 22 as atMidpointx
END

ECHO
END

BLOCK myTestBeam
material stainless_steel
Beam2
Area 1
orientation 1 1 1
I1 .08333333
I2 .08333333
J 0.16666666
END

Material stainless_steel
E 30.0E6
nu 0.33
density 0.00074
END

begin rectangular coordinate system rotz
origin = 0 10 0
z point = 0 10 5
xz point = 1 10 0
end

begin rectangular coordinate system rotx
origin = 0 0 10
z point = 5 0 10
xz point = 0 1 10
```

```
end

begin rectangular coordinate system roty
origin = 10 0 0
z point = 10 5 0
xz point = 10 0 1
end

GDSW
prt_overlap yes
prt_coarse yes
END
```

9.17. Angular Acceleration, Statics, HEX20

Refer to Section 2.17.8

```
$ Algebraic Preprocessor (Aprepro) version 6.25 (2023/10/12)
SOLUTION
statics
END

parameters
end

FILE
geometry_file hex20Beam40x.g
END

BOUNDARY
sideset baseSurf
fixed
END

LOADS

body
angular_acceleration = 0 0 100000
coordinate rotz
sideset loadSurf
pressure = -10000
END

OUTPUTS
END

HISTORY
  disp
  node dispy at node 291 as atEndpointy
  node dispy at node 311 as atMidpointy
  node dispx at node 291 as atEndpointx
  node dispx at node 311 as atMidpointx
END

ECHO
END

BLOCK myTestBeam
material stainless_steel
END

Material stainless_steel
E 30.0E6
nu 0.33
density 0.00074
```

```
END

begin rectangular coordinate system rotz
origin = 0 10 0
z point = 0 10 5
xz point = 1 10 0
end

begin rectangular coordinate system rotx
origin = 0 0 10
z point = 5 0 10
xz point = 0 1 10
end

begin rectangular coordinate system roty
origin = 10 0 0
z point = 10 5 0
xz point = 10 0 1
end

GDSW
prt_overlap yes
prt_coarse yes
END
```

9.18. Angular Acceleration, Statics, TRIA3

Refer to Section 2.17.8

```
$ Algebraic Preprocessor (Aprepro) version 6.25 (2023/10/12)
SOLUTION
statics
END

parameters
end

FILE
geometry_file tri3Beam40x.g
END

BOUNDARY
nodeset baseNodes
fixed
END

LOADS
body
angular_acceleration = 0 0 10000
coordinate rotz
nodeset cornerLoadNodes
force = 250 0 0
nodeset centerLoadNode
force = 500 0 0
END

OUTPUTS
END

HISTORY
  disp
  node dispy at node 84 as atEndpointy
  node dispy at node 104 as atMidpointy
  node dispx at node 84 as atEndpointx
  node dispx at node 104 as atMidpointx
```

```
END

ECHO
END

BLOCK myTestBeam
thickness 0.1
material stainless_steel
END

Material stainless_steel
E 30.0E6
nu 0.00 //shells are sensitive to nu. Need nu close to 0 to equate to beam theory.
density 0.00074
END

begin rectangular coordinate system rotz
origin = 0 10 0
z point = 0 10 5
xz point = 1 10 0
end

begin rectangular coordinate system rotx
origin = 0 0 10
z point = 5 0 10
xz point = 0 1 10
end

begin rectangular coordinate system roty
origin = 10 0 0
z point = 10 5 0
xz point = 10 0 1
end

GDSW
prt_overlap yes
prt_coarse yes
END
```

9.19. Rotating Shell Statics

Refer to Section 2.17.9

```
SOLUTION
    solver=gdsw
 case one
   statics
   load = 1
 case two
    tangent
 case out
     qevp
     method=anasazi
     nmodes=10
     subspace_size 100
END
FILE
geometry_file 'beam_shell.exo'
end

parameters
  eig_tol=1.0e-12
end

LOAD 1
```

```
    body
      angular_velocity 0 0 50.0
      function = 1
end
FUNCTION 1
       type LINEAR
       name "test_func1"
       data 0.0 1.0
       data 1.0 1.0
       data 2.0e4 1.0
end
OUTPUTS
disp
force
end
ECHO
mass
mass=block
end
BOUNDARY
   nodeset 1
       fixed
end
BLOCK 1 //        hex8u
        material "steel"
        quadt
        thickness 1.0
        rotational_type lagrangian
end
MATERIAL "steel"
        E 19.5e10
        nu 0
        density 7700.0
end
```

9.20. Rotating Ring Statics

Refer to Section 2.17.10

```
SOLUTION
    solver=gdsw
    statics
END
loads
     body
       angular_velocity 0 0 1.1
       coordinate 10
end
Begin rectangular coordinate system 10
    origin   = 3 1 4
    z point  = 3 1 5
    xz point = 4 1 4
end

file
    geometry_file 'ring.exo'
end
boundary
    nodeset 1 fixed
end
outputs
  force
end
echo
    mass
end
block 1
    quadt
    material "Al6061-2"
end
block 2
    beam2
    material=light
    area=1e-2
    i1=1e-2
    i2=1e-2
    j=2e-2
    orientation 0 0 1
end
material "Al6061-2"
    isotropic
    density = 4
    nu = .3
    E = 1e7
end
material light
    isotropic
    density = 0
    nu = .3
    E = 1e7
end
```

9.21. Rotating Ring Acceleration

Refer to Section 2.17.11

```
SOLUTION
    solver=gdsw
    statics
END
loads
     body
       angular_acceleration 0 0 1.1
       coordinate myCoord
end
Begin rectangular coordinate system myCoord
    origin   = 3 1 4
    z point  = 3 1 5
    xz point = 4 1 4
end

file
    geometry_file 'ring.exo'
end
boundary
    nodeset 1 fixed
end
outputs
  force
end
echo
    mass
end
block 1
    quadt
    material "Al6061-2"
end
block 2
    beam2
    material=light
    area=1e-2
    i1=1e-2
    i2=1e-2
    j=2e-2
    orientation 0 0 1
end
material "Al6061-2"
    isotropic
    density = 4
    nu = .3
    E = 1e7
end
material light
    isotropic
    density = 0
    nu = .3
    E = 1e7
end
```

9.22. Rotating Superelement Statics

Refer to Section 2.17.12

```
// solution should be identical to the single hex solution, but
// this model uses a superelement.
SOLUTION
    solver=gdsw
     statics
END

loads
     body
       angular_velocity 3 0 0
end

file
    geometry_file 'rotating_hex_se.exo'
end

boundary
    nodeset 1 fixed
    nodeset 2 fixed
end

outputs
  force
end

echo
    mass
    force
end

block 1
    superelement
    format=netcdf
    file=rotating_hex_gold.ncf
    map locations
end
```

9.23. Rotating Superelement Beam Statics

Refer to Section 2.17.15

```
SOLUTION
    solver=gdsw
 case out
 statics
     load=1
END

HISTORY
   nodeset 2
   disp
end
FILE
geometry_file 'beam_se.exo'
  omit block 1
end
LOAD 1
    body
      angular_velocity 0 0 5.0
    function = 1
end
FUNCTION 1
       type LINEAR
       name "test_func1"
       data 0.0 1.0
       data 1.0 1.0
       data 2.0e4 1.0
end
outputs
disp
force
end
echo
mass
mass=block
        input
end

MATERIAL 'dead'
 isotropic
        E = 10
        nu = 0.29
        density = 0
END

block 1
        material dead
end
block 2
    superelement
    file=cbr_hex.netcdf
    map = location
end
```

9.24. Fatigue Output of Single DOF in Random Vibration

Refer to Section 2.18

9.24.1. Modal Random Vibration

```
Solution
   title 'test of a single dof example for fatigue'
   case eig
        eigen nmodes=all
   case frf
        modalfrf
        load=1
   case ran
        modalranvib
        noSVD
end

FILE
   geometry_file 'onehexran.exo'
end

Ranloads
   matrix 97
   load=1
      sideset 2
      pressure 7
end

Matrix-function 97
   dimension 1x1
   symmetry=symmetric
   data 1,1
   real function 99
end

Function 99
   type linear
   data 0 9
   data 5000 9
end

Frequency
  freq_min 10
  freq_max  100
  freq_step  10
  sideset 2
  disp
  acceleration
end
BOUNDARY
  sideset 1
   fixed
  sideset 2
    y=0 z=0
end
OUTPUTS
   maa
   disp
   vrms
end
Echo
 input
 disp
 force
```

```
 rhs
 mass
end
LOAD 1
   sideset 2
       pressure 3
       function 1
end
Function 1
   type linear
   data 0 3.0
   data 1e5 3.0
end
Block 1
    material 3
end
Material 3
        E = 1e7
        nu = .3
        density 0.000259
end
```

9.24.2. Fatigue Solution

```
SOLUTION
   title 'test of a single dof example for fatigue'
   case eig
        eigen nmodes=all
   case frf
        modalfrf
        load=1
   case ran
        modalranvib
        noSVD
   case out
        fatigue
           method=wirsching
           duration=0.001
end

FILE
   geometry_file 'onehexran.exo'
end

Ranloads
   matrix 97
   load=1
      sideset 2
      pressure 7
end

Matrix-function 97
   dimension 1x1
   symmetry=symmetric
   data 1,1
   real function 99
end

Function 99
   type linear
   data 0 9
   data 5000 9
end

Frequency
  freq_min 10
```

```
  freq_max  100
  freq_step  10
  sideset 2
  disp
  acceleration
end

BOUNDARY
  sideset 1
   fixed
  sideset 2
    y=0 z=0
end

OUTPUTS
   disp
   vrms
end

ECHO
 input
 disp
 force
   rhs
 mass
 fatigue
end

LOAD 1
   sideset 2
       pressure 3
       function 1
end

Function 1
   type linear
   data 0 3.0
   data 1e5 3.0
end

BLOCK 1
    material 3
end

MATERIAL 3
        E = 1e7
        nu = .3
        density 0.000259
        Fatigue_A1 12.1689
        Fatigue_A2 -3
        Stress_Ratio -1.0
        Fatigue_Stress_Scale 1.0 // 0.001
        std_err 0.01
        t_dist 123.4
end
```

9.25. Fatigue Output of Dogbone

Refer to Section 2.18.6

```
SOLUTION
case eig
eigen nmodes=150 // a 5 Hz wave is effectively static loading for this thing. We need a TON of modes for this to work.
shift=-1e9
        restart = auto
    case rand
      modalranvib
      lfcutoff = 10 // DON"T USE -1 FOR STRESS.
      nosvd
    case fatigue
        fatigue
           method=wirsching
end

FILE
   geometry_file 'dogbone_eng.exo'
end


PARAMETERS
   wtmass=0.002589
end

BOUNDARY
nodeset 1
 rotx=0
 roty=0
 rotz=0
end

RANLOADS
   matrix=11
   load=1
      nodeset 1
      force=1 0 0
      scale = 1.0
      nodeset 2
      force=1 0 0
      scale = -1.0
end

Matrix-Function 11
   symmetry = symmetric
   dimension = 1x1
   data 1,1
   real FUNCTION 1
end


FUNCTION 1
   Name="psd"
   type  loglog
   data 4.0 1e-13
   data 4.49 1e-13
   data 4.5 4931280.0 // 2219.96  // for sine wave of 3.141165e3, want rms of sqrt(4933460)=2221.139.
   data 5.5 4931280.0 // 2219.96
   data 5.51 1e-13
   data 6.0 1e-13
end

// PSD magnitude based on integral function from wolfram alpha. Execute the following line:
// 4931280.000000*1.000000+integral 10^-13307.372658 * x^20382.441146 dx from x=4.490000 to x=4.500000 + integral 10^18487.818124
```

```
Frequency
   nodeset 1
   freq_min=4.0
   freq_max=6.0
   freq_step=0.01
end

LOADS
end

OUTPUTS
    vrms
end

ECHO
  mass
end

BLOCK 77
   rbar
end

BLOCK 99
  material AISI4140
end

BLOCK 100
   material AISI4140
end

block 1
 conmass
 mass =  0.0 // used to be dead
end

MATERIAL AISI4140
   E=29.0e6 // psi
   nu=0.32
   density=0.283 // lb/in^3
   // these values are not appropriate for this material
        Fatigue_A1 31.6
        Fatigue_A2 -14.0845
   fatigue_stress_scale 1.0e-3
end
```

9.26. Fatigue Output of Pinned Shell

Refer to Section 2.18.7

```
SOLUTION
   title 'test of a simple pinned plate for fatigue'
   case statics
       statics
       load=11
   case eig
        eigen
        nmodes=1
   case ran
        modalranvib
        noSVD
   case nb
        fatigue
           method=wirsching
END

FILE
   geometry_file 'pinned_plate_fatigue.exo'
END

Ranloads
   matrix 97
   load=1
      sideset 1
      pressure 1
END

matrix-function 97
   dimension 1x1
   symmetry=symmetric
   data 1,1
   real function 99
end

function 99
   type linear
   data 1 1e-20
   data 4 1e-20
   data 4.01 1
   data 4.99 1
   data 5.00 1e-20
   data 500 1e-20
end


frequency
  freq_min 4
  freq_max  5
  freq_step  0.001
  block 1 displacement
end


damping
end

BOUNDARY
  nodeset 1
   x=0 y=0 z=0
  nodeset 2
   x=0 y=0 z=0
END
```

```
OUTPUTS
   disp
   vrms
   stress
END

ECHO
 input
 mass
END

LOAD 11
   sideset 1
       pressure 1
END

LOAD 14
   sideset 1
       pressure 1
       function=14
END

function 14
   // white noise
   type linear
   data 0 1.0
   data 1e5 1.0
end

BLOCK 1
    material 3
END

MATERIAL 3
        E = 1e7
        nu = .3
        density 0.000259
        Fatigue_A1 12.1689
        Fatigue_A2 -3
        Stress_Ratio -1.0
        Fatigue_Stress_Scale=1e-4
END
```

9.27. Nodal Loading vs Sideset Loading for Modal Random Vibration

Refer to Section 2.18.10

```
// Flat Plate Problem Solution in Roark

solution
case 'eig'
eigen
        nmodes=50
case 'randomvib'
modalranvib
end

RANLOADS
matrix = 1
load = 1
nodeset 10
force = 0 0 1
scale = 9.3234e-4
END

MATRIX-FUNCTION 1
name 'pressure spectral density'
symmetry = hermitian
dimension = 1x1
data 1,1
real function 1
END

FUNCTION 1
name = 'psd'
type = 'loglog'
data 10.0   690.0
data 20.0   690.0
  data 30.0   6900.0
data 100.0  6900.0
data 500.0  690.0
data 1000.0 690.0
END

parameters
end

damping
gamma 0.02
end

FREQUENCY
method=log
freq_min 1.0
freq_max 1000
NF 1000
nodeset 5
disp
accel
END

file
    geometry_file 'flat_input.exo'
end

boundary
    nodeset 1
z = 0.0
nodeset 2
```

```
z = 0.0
nodeset 3
fixed
nodeset 4
fixed
end

history
nodeset '5'
disp
end

outputs
  disp
end

echo
end

block 1
    HEXSHELL
    sideset 1
    material "Example-2"
end

material "Example-2"
    isotropic
    density = 8.56e3
    nu = 0.34
    E = 9.02e10
end
```

9.28. Sierra/SM to Sierra/SD Coupling

Refer to Section 2.20

9.28.1. Files for Preloaded Static Beam

Sierra/SM input file

```
begin sierra chatter_contact

  begin function ramp1
    type is piecewise linear
    begin values
      0.0     0.0
      0.5     1.0
      1.0     1.0
    end
  end

  begin function ramp2
    type is piecewise linear
    begin values
      0.0     0.0
      0.5     0.0
      1.0     1.0
    end
  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 material aluminum
    density         = 2.59e-2
    begin parameters for model neo_hookean
      youngs modulus = 7.8e+7
      poissons ratio = 0.0
    end
  end

  begin material stiff
    density         = 2.59e-2
    begin parameters for model neo_hookean
      youngs modulus = 7.8e+11
      poissons ratio = 0.0
    end
  end

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

    begin parameters for block block_1
      material = aluminum
      model = neo_hookean
    end
    begin parameters for block block_2
      material = stiff
      model = neo_hookean
    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 = 0.05
    end
  end
  termination time = 1.0
end time control

begin adagio region adagio
  use finite element model mesh1


  begin user output
    nodeset = nodelist_2
    compute global extension as average of nodal displacement
  end

  begin user output
    surface = surface_1000
    compute global deflection as average of nodal displacement
  end


  ### output description ###
  begin Results Output output_adagio
    Database Name = bar_preload_{extensionPressure}_{deflectionPressure}.e
    Database Type = exodusII
    At Step 0, Increment = 1
    nodal Variables = displacement as displ
    nodal variables = coordinates
     nodal variables = reaction
    element variables = stress
    element variables = log_strain
    component separator character = none
  end results output output_adagio

  begin history output
database name = bar_preload_{extensionPressure}_{deflectionPressure}.h
     at time 1.0 interval = 1.0
     global extension
     global deflection
  end


  ### definition of BCs ###
  begin fixed displacement
    node set = nodelist_1
    component = xy
  end
  begin fixed displacement
    node set = nodelist_2
    components = Y
  end
  begin fixed displacement
    block = block_1 block_2
    components = z
  end

  begin traction
    surface = surface_2
    direction = x
    function = ramp2
    scale factor = {extensionPressure}
  end


  begin traction
```

```
         surface = surface_1000
         direction = y
         function = ramp1
         scale factor = {deflectionPressure}
      end

      begin solver
        begin cg
          target relative residual = 1.0e-6
          begin full tangent preconditioner
          end
        end
      end

    end
  end
end
```

Sierra/SD input file for

```
solution
{ifdef(preload)}
case preload
  receive_sierra_data
  load = 0
  {ifdef(no_geom_stiff)}
  no_geom_stiff
  {endif}
  {ifdef(equilibrium)}
  include_internal_force = off
  {endif}
{endif}
case static
  statics
  solver = gdsw
  load = 10
end

GDSW
END

file
  geometry_file '{geomFile}'
end

history
  database name = '{historyFile}'
  sideset = surface_1000
  displacement
end

outputs
  disp
  stress
end

boundary
  nodeset 1 x=0 y=0
  nodeset 2 y=0
  block 1 2 z=0
end

BLOCK 1
  material aluminum
END
```

```
BLOCK 2
  material stiff
END

MATERIAL aluminum
  E 7.8e7
  nu = 0.0
  density 2.59e-2
END

MATERIAL stiff
  E 7.8e11
  nu = 0.0
  density 2.59e-2
END


LOAD 0
END

function ramp
  type = linear
  data 0 1
  data 1 1
end


load 10
{ifdef(extensionPressure)}
  sideset surface_2    traction 1 0 0 scale {extensionPressure}
{endif}
{ifdef(deflectionPressure)}
  sideset surface_1000 traction 0 1 0 scale {deflectionPressure}
{endif}
end

ECHO
  input
END
```

9.28.2. Files for Preloaded Eigen Comparison to Abaqus

Sierra/SM input file

```
Begin sierra  cylinder only
  title  Membrane

  define direction y with vector 0.0 1.0 0.0
  define direction x with vector 1.0 0.0 0.0
  define direction z with vector 0.0 0.0 1.0
  define point origin with coordinates 0.0 0.0 0.0

  Begin definition for function zero
    type is constant
    Begin values
      0.0
    end values
  end
  Begin definition for function one
    type is constant
    Begin values
      1.0
```

```
  end values
end
Begin definition for function function_100
  type is piecewise linear
  Begin values
    0.0  0.0
    0.8  1.0
    1.0  1.0
  end values
end definition for function function_100

Begin property specification for material mat_100
   density = 0.1E-08                      # 10+3 kgm/mm3
   Begin parameters for model elastic
      youngs modulus = 6
      poissons ratio =0.3
   end parameters for model elastic
end property specification for material mat_100

Begin solid section solid_100
   strain incrementation = strongly_objective
   hourglass formulation = total
end solid section solid_100

Begin finite element model plate
  Database name = bar.exo
  Database type = exodusII
  component separator character = ""
  Begin parameters for block block_100
    material mat_100
    solid mechanics use model elastic
    section = solid_100
  end parameters for block block_100
end

Begin adagio procedure procedure_1
  Begin time control
    Begin time stepping block p0
      start time = 0.0
      Begin parameters for adagio region region_1
        number of time steps = 100
      end parameters for adagio region region_1
    end time stepping block p0
    termination time = 1.0
  end time control

  Begin adagio region region_1
    jas mode solver
    jas mode output
    jas mode reactions
    failure debug output
    logfile detail = -1

    use finite element model plate

    Begin fixed displacement
      node set = nodelist_1
      components = x,y,z
    end fixed displacement
    Begin fixed displacement
      node set = nodelist_2
      components = y,z
    end fixed displacement
    Begin prescribed displacement
      node set = nodelist_2
      component = x
      function = function_100
      scale factor = 1.0
```

```
      end prescribed displacement

      Begin results output output_1
        database name = c_adagio_preload.e
        database type = exodusII
        component separator character = ""
        at step 0 increment = 1
        nodal variables = displacement as displ
      end results output output_1
    Begin results output output_2
        database name = salinas_preload.e
        database type = exodusII
        component separator character = ""
        additional times = 1.0
        nodal variables = displacement as displ
        nodal variables = node_filter as filter
        element variables = stress as stress
        element variables = density as fiberdensity
        element Variables = element_thickness as fiberthickness
        element variables = ends_per_length as epl
        element variables = cord_modulus as fibmod
        element variables = memb_stress as memstr
        element variables = cord_ax as ax
        element variables = cord_ay as ay
      end results output output_2
      Begin solver
        Begin loadstep predictor
          type = scale_factor
          scale factor = 1.0 0.0
        end loadstep predictor
        Begin control contact
          level = 1
          target relative residual = 0.01
          acceptable relative residual = 0.1
          target relative contact residual = 0.001
          acceptable relative contact residual = 0.01
          maximum iterations = 500
          minimum iterations = 10
          lagrange initialize = none
          lagrange adaptive penalty = off
        end control contact
        Begin cg
          target relative residual = 0.005
          acceptable relative residual = 0.05
          minimum residual improvement = 0.5
          maximum iterations = 500
          minimum iterations = 10
          reset limits 70 30 10.0 0.5
          iteration print = 1
          line search actual
          preconditioner = block
        end cg
      end solver
    end adagio region region_1
  end adagio procedure procedure_1
end sierra  cylinder only
```

Sierra/SD input file

```
SOLUTION
    solver=gdsw
  case one
    receive_sierra_data
    lumped
  case two
    eigen
```

```
    nmodes all
    shift = -1000
END

FILE
geometry_file 'salinas_preload.e'
end

BOUNDARY
  nodeset 1 x=0 y=0 z=0
  nodeset 2 y=0 z=0
end

OUTPUTS
deform
end

ECHO
  mass=block
end

BLOCK 100
  material "steel"
  hex8u
  sd_factor 1.0
end
//E=6C10 @ 0.001 strain
//bulk modulus = E/[3(1-2*nu0]=16.66E=100
MATERIAL "steel"
E  6.0
nu 0.3
density 0.1E-08
end
```

9.28.3. Files for SD to SM Element Comparison

Sierra/SM input file

```
BEGIN SIERRA spot_weld

  BEGIN DEFINITION FOR FUNCTION linearFunc
    type = piecewise linear
    begin values
     -1.0 -1.0e+0
      1.0  1.0e+0
    end
  end

  BEGIN DEFINITION FOR FUNCTION loadRamp
    type = analytic
    evaluate expression = "cos_ramp(t, 0.0, 0.8)"
  END

  BEGIN PROPERTY SPECIFICATION FOR MATERIAL a
    density = 7800
    BEGIN PARAMETERS FOR MODEL elastic
      youngs modulus = 2E11
      poissons ratio = 0.33
    END
  END
  BEGIN PROPERTY SPECIFICATION FOR MATERIAL b
    density = 7800
    BEGIN PARAMETERS FOR MODEL elastic
      youngs modulus = 3E11
```

```
      poissons ratio = 0.33
    END
  END

  BEGIN PROPERTY SPECIFICATION FOR MATERIAL c
    density = 7800
    BEGIN PARAMETERS FOR MODEL elastic
      youngs modulus = 4E11
      poissons ratio = 0.33
    END
  END
  BEGIN PROPERTY SPECIFICATION FOR MATERIAL d
    density = 7800
    BEGIN PARAMETERS FOR MODEL elastic
      youngs modulus = 5E11
      poissons ratio = 0.33
    END
  END

  BEGIN SOLID SECTION s1
{if(form != "composite")}
    formulation = {form}
    {endif}
  END

  BEGIN FINITE ELEMENT MODEL plate
  database name = {mesh}
    database type = exodusII

    BEGIN PARAMETERS FOR BLOCK block_1
      material = a
      model = elastic
      section = s1
    END
    BEGIN PARAMETERS FOR BLOCK block_2
      material = b
      model = elastic
      section = s1
    END
    BEGIN PARAMETERS FOR BLOCK block_3
      material = c
      model = elastic
      section = s1
    END
    BEGIN PARAMETERS FOR BLOCK block_4
      material = d
      model = elastic
      section = s1
    END
  END

    define direction tracDir with vector 1 2 3

    BEGIN PRESTO PROCEDURE procedure

      BEGIN TIME CONTROL
        BEGIN TIME STEPPING BLOCK p1
  START TIME = 0.0
  BEGIN PARAMETERS FOR adagio REGION region
            number of time steps = 1
  END
END
termination time = 1.0
      END TIME CONTROL

      BEGIN PRESTO REGION region

USE FINITE ELEMENT MODEL plate
```

```
BEGIN FIXED DISPLACEMENT
  surface = surface_100
  component = x y z
END
        begin traction
           surface = surface_1000
           function = loadRamp
           direction = sierra_direction_x
           scale factor = 1.0e+5
        end
        begin traction
           surface = surface_1000
           function = loadRamp
           direction = sierra_direction_y
           scale factor = 2.0e+5
        end
        begin traction
           surface = surface_1000
           function = loadRamp
           direction = sierra_direction_z
           scale factor = 3.0e+5
        end

BEGIN RESULTS OUTPUT output
      database name = {out}
  database type = exodusII
  start time = 1.0
  at time 1.0, increment = 1e-2
          nodal variables = displacement(x) as dispx
          nodal variables = displacement(y) as dispy
          nodal variables = displacement(z) as dispz
          #nodal variables = reaction_force
          nodal variables = force_external
END

        BEGIN SOLVER
          begin cg
            target relative residual = 1.0e-8
            begin full tangent preconditioner
            end
          end
        END

      END
    END
  END
```

Sierra/SD input file

```
SOLUTION
  statics
END

  BEGIN FUNCTION linearFunc
    type = piecewise linear
    begin values
     -1.0 -1.0e+0
      1.0  1.0e+0
    end
  end

  GDSW
    solver_tol 1.0e-8
  END
```

```
  MATERIAL a
    density = 7800
    e = 2E11
    nu = 0.33
  END
  MATERIAL b
    density = 7800
    e = 3E11
    nu = 0.33
  END
  MATERIAL c
    density = 7800
    e = 4E11
    nu = 0.33
  END
  MATERIAL d
    density = 7800
    e = 5E11
    nu = 0.33
  END

  FILE
    geometry_file = {mesh}
  END

  BLOCK block_1
    material = a
    {form}
  END
  BLOCK block_2
    material = b
    {form}
  END
  BLOCK block_3
    material = c
    {form}
  END
  BLOCK block_4
    material = d
    {form}
  END

  BOUNDARY
    sideset 100 y=0 z=0 x=0
  END

  LOADS
    sideset 1000 traction 1 2 3 scale 1.0e+5
  END


   OUTPUTS
     database name = {out}
     disp
#     reaction_force
#     force
   END
```

Sierra/SM input file

```
begin sierra bender
  begin function prescribed_force
    type is piecewise linear
    ordinate is force
    abscissa is time
    begin values
      0.0           0.0
      1.0           0.5
      2.0           1.0
      3.0           1.0
    end
  end

  begin function force_strain
    type is piecewise linear
    ordinate is force
    abscissa is engineering_strain
    begin values
      -1.0     -1.0e7
       1.0      1.0e7
    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 spring section HydrogenScattering
    force strain function = force_strain
    default stiffness =1.0e7
    mass per unit length = 0.0
  end

  begin finite element model one_beam
    Database Name = unit_length_beam.g
    Database Type = exodusII
    begin parameters for block UnitLengthBeam
      section = HydrogenScattering
    end
  end finite element model one_beam

  begin adagio procedure adagio
    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for adagio region SouthWest
          time increment = 1
        end
      end
      termination time = 3
    end
    begin adagio region SouthWest
      use finite element model one_beam
      begin Results Output output
        Database Name = spring.exo
        Database Type = exodus
        additional times = 3.0
        nodal variables = displacement(x) as dispx
        nodal variables = displacement(y) as dispy
        nodal variables = displacement(z) as dispz
        nodal Variables = rotational_displacement(x) as rotx
        nodal Variables = rotational_displacement(y) as roty
        nodal Variables = rotational_displacement(z) as rotz
        element variables = spring_engineering_strain as spring_engineering_strain
```

```
      end
      begin prescribed force
         node set = right
         direction = x
         function = sierra_constant_function_one
         scale factor = {force}
         active periods = p1
      end

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

      begin fixed displacement
        node set = right
        components = Y Z
      end
      begin fixed rotation
        node set = left, right
        components = X Y Z
      end
      begin solver
        begin cg
          target relative residual = 1.0e-5
          maximum iterations = 65
          minimum iterations = 0
          preconditioner = probe
        end
      end
    end
  end
end sierra bender
```

Sierra/SD input file

```
Solution
  case d
  statics
  load 1
end

File
geometry_file unit_length_beam.g
end

Boundary
  nodeset left
    fixed
  nodeset right
    y 0
    z 0
    rotx 0
    roty 0
    rotz 0
end

Load 1
  nodeset right
  force 1 0. 0.
  scale {force}
  function 1
end

Function 1
  type linear
```

```
  data 0.0    1.0
  data 1.0e8  1.0
end


Outputs
deform
  database name s.exo
end

Echo
  input on
end

Begin rectangular coordinate system Euclidean
  Origin = 0 0 0
  Z Point = 0 0 1
  XZ Point = 1 0 0
end

Block UnitLengthBeam
spring
  coordinate Euclidean
  Kx 10.0e6
  Ky 10.0e6
  Kz 10.0e6
end
```

Sierra/SM input file

```
begin sierra truss
  begin function prescribed_force
    type is piecewise linear
    ordinate is force
    abscissa is time
    begin values
      0.0           0.0
      1.0           0.5
      2.0           1.0
      3.0           1.0
    end
  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 material aluminium
    density = 2.5880e-4
    begin parameters for model elastic
      youngs modulus = 10.0e6
      poissons ratio = 0.3
    end
  end

  begin material linear_elastic
    density = 7.3240e-4
    begin parameters for model elastic
      youngs modulus = 30.0e6
      poissons ratio = 0.3
    end
  end

  begin truss section HydrogenScattering
    area = 0.25
  end

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

    begin parameters for block block_1
      material = linear_elastic
      model = elastic
      section = HydrogenScattering
    end

    begin parameters for block block_2
      material = aluminium
      model    = elastic
      section  = HydrogenScattering
    end
  end

  begin adagio procedure adagio
    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for adagio region SouthWest
          time increment = 1
        end
      end
      termination time = 3
    end
```

```
    begin adagio region SouthWest
      use finite element model mesh1
      begin Results Output output
        Database Name = truss.exo
        Database Type = exodus
        additional times = 3.0
        nodal variables = displacement(x) as dispx
        nodal variables = displacement(y) as dispy
        nodal variables = displacement(z) as dispz
        nodal Variables = rotational_displacement(x) as rotx
        nodal Variables = rotational_displacement(y) as roty
        nodal Variables = rotational_displacement(z) as rotz
      end

      begin fixed displacement
        node set = nodelist_100
        components = x y z
      end
      begin fixed displacement # x axis, x deformation
        node set = nodelist_500
        components = y z
      end
      begin prescribed force
        node set = nodelist_200
        direction = x
        function = prescribed_force
        scale factor = 1.0e+3
      end
      begin fixed displacement # y axis, y deformation
        node set = nodelist_600
        components = x y
      end
      begin prescribed force
        node set = nodelist_400
        direction = y
        function = prescribed_force
        scale factor = 1.0e+3
      end
      begin fixed displacement # z axis, z deformation
        node set = nodelist_700
        components = x z
      end
      begin prescribed force
        node set = nodelist_300
        direction = z
        function = prescribed_force
        scale factor = 1.0e+3
      end
      begin solver
        begin cg
          target relative residual = 1.0e-10
          maximum iterations = 65
          minimum iterations = 0
          preconditioner = probe
        end
      end
    end
  end
end sierra truss
```

Sierra/SD input file

```
SOLUTION
  case d
  statics
  load 1
```

```
// lumped
end

FILE
geometry_file truss.g
end

BOUNDARY
    nodeset nodelist_100
    x 0
    y 0
    z 0
    nodeset nodelist_500
    y 0
    z 0
    nodeset nodelist_600
    x 0
    y 0
    nodeset nodelist_700
    x 0
    z 0
end

load 1
nodeset nodelist_200
  force 1 0. 0.
  scale 1000
  nodeset nodelist_400
  force 0. 1 0.
  scale 1000
  nodeset nodelist_300
  force 0. 0. 1
  scale 1000
end

OUTPUTS
deform
  database name s.exo
end

ECHO
input off
end

BLOCK 1
material linear_elastic
truss
Area 0.25
end

BLOCK 2
  material aluminium
truss
Area 0.25
end

material linear_elastic
  density 7.3240e-4
  E 30.0e6
  nu 0.3
end

material aluminium
  density 2.5880e-4
  E 10.e6
  nu 0.3
end
```

Sierra/SM input file

```
begin sierra axial_force
  begin function prescribed_force
    type is piecewise linear
    ordinate is force
    abscissa is time
    begin values
      0.0           0.0
      1.0           0.5
      2.0           1.0
      3.0           1.0
    end
  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 material aluminium
    density = 2.5880e-4
    begin parameters for model elastic
      youngs modulus = 10.0e6
      poissons ratio = 0.3
    end
  end

  begin beam section beam_1
    section = bar
    D1 = 1.0
    D2 = 1.0
    t axis = 0.0 1.0 0.0
  end

  begin beam section beam_2
    section = box
    D1 = 1.005
    D2 = 1.005
    D3 = 4.5243753901374861e-01
    t axis = 0.0 1.0 0.0
  end

  begin beam section beam_3
    section = rod
    D1 = 1.1283791670955126
    D2 = 1.1283791670955126
    t axis = 0.0 1.0 0.0
  end

  begin beam section beam_4
    section = tube
    D1 = 1.1340210629309899
    D2 = 1.1340210629309899
    D3 = 5.1052109343507690e-01
    t axis = 0.0 1.0 0.0
  end

  begin finite element model mesh1

    Database Name = beam.g
    Database Type = exodusII

    begin parameters for block block_1
      material = aluminium
      model = elastic
      section = beam_1
    end
```

```
  begin parameters for block block_2
    material = aluminium
    model = elastic
    section = beam_2
  end

  begin parameters for block block_3
    material = aluminium
    model = elastic
    section = beam_3
  end

  begin parameters for block block_4
    material = aluminium
    model = elastic
    section = beam_4
  end
end

begin adagio procedure adagio
  begin time control
    begin time stepping block p1
      start time = 0.0
      begin parameters for adagio region SouthWest
        time increment = 1
      end
    end
    termination time = 3
  end

  begin adagio region SouthWest
    use finite element model mesh1
    begin Results Output output
      Database Name = axial_force.exo
      Database Type = exodus
      additional times = 3.0
      nodal variables = displacement(x) as dispx
      nodal variables = displacement(y) as dispy
      nodal variables = displacement(z) as dispz
      nodal Variables = rotational_displacement(x) as rotx
      nodal Variables = rotational_displacement(y) as roty
      nodal Variables = rotational_displacement(z) as rotz
    end

    begin prescribed force
      node set = nodelist_1
      direction = X
      function = prescribed_force
      scale factor = 1.0e+3
    end

    begin fixed displacement
      node set = nodelist_1
      components = Y Z
    end

    begin fixed displacement
      node set = nodelist_2 nodelist_3 nodelist_4 nodelist_5
      components = X Y Z
    end

    begin fixed rotation
      include all blocks
      components = X Y Z
    end

    begin solver
```

```
        begin cg
          target relative residual = 1.0e-5
          maximum iterations = 65
          minimum iterations = 0
          preconditioner = probe
        end
      end
    end
  end
end sierra axial_force
```

Sierra/SD input file

```
SOLUTION
  case d
  statics
  load 1
// lumped
end

FILE
geometry_file beam.g
end

BOUNDARY
    nodeset 2 3 4 5
      fixed
end

load 1
nodeset 1
  force 1 0. 0.
  scale 1000
end

OUTPUTS
deform
  database name s.exo
end

ECHO
input off
end

block 1
material aluminium
beam2
  area 1.0
  I1 = 8.3333333333333329e-02
  I2 = 8.3333333333333329e-02
  J = 1.6666666666666666e-01
end

block 2
  material aluminium
  beam2
  area 1.0
  I1 = 8.1520710209194588e-02
  I2 = 8.1520710209194588e-02
  J = 1.6304142041838918e-01
end
block 3
material aluminium
beam2
  area 1.0
  I1 = 1.2732395447351625
```

```
  I2 = 1.2732395447351625
  J = 2.5464790894703251
end
block 4
  material aluminium
  beam2
  area 1.0
  I1 = 1.2455447034389040
  I2 = 1.2455447034389040
  J = 2.4910894068778080
end

material aluminium
  density 2.5880e-4
  E 10.e6
  nu 0.3
end
```

Sierra/SM input file

```
begin sierra bender
  begin function prescribed_force
    type is piecewise linear
    ordinate is force
    abscissa is time
    begin values
      0.0           0.0
      1.0           0.5
      2.0           1.0
      3.0           1.0
    end
  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 material aluminium
    density = 2.5880e-4
    begin parameters for model elastic
      youngs modulus = 10.0e6
      poissons ratio = 0.3
    end
  end

  begin beam section HydrogenScattering
    section = bar
    D1 = 1.0
    D2 = 1.0
    t axis = 0.0 1.0 0.0
  end

  begin finite element model along_x_axis
    Database Name = {mesh}.g
    Database Type = exodusII
    begin parameters for block rodlength20
      material = aluminium
      model = elastic
      section = HydrogenScattering
    end
  end

  begin adagio procedure adagio
    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for adagio region SouthWest
          time increment = 1
        end
      end
      termination time = 3
    end
    begin adagio region SouthWest
      use finite element model along_x_axis
      begin Results Output output
        Database Name = bendy.exo
        Database Type = exodus
        additional times = 3.0
        nodal variables = displacement(x) as dispx
        nodal variables = displacement(y) as dispy
        nodal variables = displacement(z) as dispz
        nodal Variables = rotational_displacement(x) as rotx
        nodal Variables = rotational_displacement(y) as roty
        nodal Variables = rotational_displacement(z) as rotz
      end
```

```
      begin prescribed force
         node set = right
         direction = y
         function = sierra_constant_function_one
         scale factor = {force}
      end

      begin fixed displacement
        include all blocks
        components = X Z
      end

      begin fixed displacement
         node set = left
         components = X Y Z
      end
      begin fixed rotation
         node set = left
        components = X Y Z
      end
      begin fixed rotation
        block = rodlength20
        components = X Y
      end

      begin solver
        begin cg
          target relative residual = 1.0e-2
          maximum iterations = 120
          minimum iterations = 0
          preconditioner = probe
        end
      end
    end
  end
end sierra bender
```

Sierra/SD input file

```
SOLUTION
  case d
  statics
  load 1
// lumped
end

GDSW
  solver_tol = 1.e-6
end

FILE
geometry_file {mesh}.g
end

BOUNDARY
nodeset left
fixed

end

load 1
  nodeset right
  force 0. 1 0.
  scale {force}
end
```

```
OUTPUTS
deform
  database name s.exo
end

ECHO
input off
end

block 1
  material aluminium
beam2
  area 1.0
  I1 = 8.3333333333333329e-02
  I2 = 8.3333333333333329e-02
  J = 1.6666666666666666e-01
end

material aluminium
  density 2.5880e-4
  E 10.e6
  nu 0.3
end
```

Sierra/SM input file

```
begin sierra torque

  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 aluminium
    density = 2.5880e-4
    begin parameters for model elastic
      youngs modulus  = 1.0e7
      poissons ratio  = 0.3
    end
  end

  begin beam section HydrogenScattering
    section = bar
    D1 = 1.0
    D2 = 1.0
    t axis = 0.0 1.0 0.0
  end

  begin finite element model one_beam
    Database Name = twist.g
    Database Type = exodusII
    begin parameters for block bxexaxm
      material = aluminium
      model = elastic
      section = HydrogenScattering
    end
  end


  begin adagio procedure adagio
    begin time control
      begin time stepping block p1
        start time = 0.0
        begin parameters for adagio region SouthWest
          time increment = 1
        end
      end
      termination time = 3
    end
    begin adagio region SouthWest
      use finite element model one_beam
      begin Results Output output
        Database Name = twistx.exo
        Database Type = exodus
        additional times = 3.0
        nodal variables = displacement(x) as dispx
        nodal variables = displacement(y) as dispy
        nodal variables = displacement(z) as dispz
        nodal Variables = rotational_displacement(x) as rotx
        nodal Variables = rotational_displacement(y) as roty
        nodal Variables = rotational_displacement(z) as rotz
      end

      begin prescribed moment
         node set = right
         direction = x
         function = sierra_constant_function_one
         scale factor = {moment}
         active periods = p1
      end
      begin fixed displacement
         node set = left, right
```

```
        include all blocks
        components = X Y Z
      end
      begin fixed rotation
        node set = left
        components = X Y Z
      end
      begin fixed rotation
        node set = right
        components = Y Z
      end
      begin solver
        begin cg
          target relative residual = 1.0e-5
          maximum iterations = 65
          minimum iterations = 0
          preconditioner = probe
        end
      end
    end
  end
end sierra torque
```

Sierra/SD input file

```
Solution
  case d
  statics
  load 1
end

File
geometry_file twist.g
end

Boundary
  nodeset left
    fixed
  nodeset right
    x 0
    y 0
    z 0
    roty 0
    rotz 0
end

Load 1
  nodeset right
  moment 1 0. 0.
  scale {moment}
  function 1
end

Function 1
  type linear
  data 0.0    1.0
  data 1.0e8  1.0
end


Outputs
deform
  database name s.exo
end

Echo
```

```
input off
end

Block bxexaxm
  material aluminium
beam2
  area 1.0
  I1 = 8.3333333333333329e-02
  I2 = 8.3333333333333329e-02
  J = 1.6666666666666666e-01
  orientation 0 1 0
  offset 0 0 0
end


Material aluminium
  density 2.5880e-4
  E 10.e6
  nu 0.3
end
```

9.29. Sensitivity to Parameters

Refer to Section 2.21

```
//salinas input created using nasgen from nastran file 'springrbar.bdf'
SOLUTION
    solver=gdsw
  case sens
  title 'two hexes, connected by tied joint'
   eigen nmodes=1

END

FILE
//   geometry_file 'twoHex.exo'
   geometry_file 'twoHex.exo'
END

sensitivity
   values = all
end

gdsw
  solver_tol=1e-12
  prt_debug=2
  orthog_option 0
end


PARAMETERS
// wtmass=0.00259
//   eigen_norm=visualization
END

BOUNDARY
    sideset 2  // nastran SID=2
        fixed
    sideset 4  // nastran SID=2
        fixed
    sideset 1
        y=0 x=0
END

LOADS
END

OUTPUTS
        disp
END

ECHO
END

BLOCK 1
   material 1
END

BLOCK 2
   material 2
END

material 1
   E=10e6
   nu=0.0
   density = 0.000256
end
```

```
material 2
   E=20e6
   nu=0.0
   density = 0.000256
end

tied joint
  normal definition = none
    surface 1,3
    search tolerance = 0.02
    connect to block 13
    side = average
end

block 13
   joint2g
   Kx = elastic 1e8
   Ky = elastic 1e8
   Kz = elastic 1e7  +/- 10 %
   Krx = elastic 1e8
   Kry = elastic 1e8
   Krz = elastic 1e8
End
```

9.30. Shock Tube SI

Refer to Section 2.22

```
SOLUTION
    solver=gdsw
    nltransient
    tolerance 1.0e-8
    time_step 4e-05
    nsteps 400
END

FILE
    geometry_file 'temp1/shocktube_SI.par'
END

LOADS
    sideset 4
    acoustic_vel = -5
    function = 1
END

BOUNDARY
    sideset 6
    absorbing
END

HISTORY
    node_list_file 'nodeshock'
    apressure
END

FUNCTION 1
   type analytic
   name "sine 1000"
   evaluate expression = "omega = 2 * pi * 1000; sin(omega*t)"
END

OUTPUTS
END

ECHO
NLresiduals
END

BLOCK 1
material "air"
END

MATERIAL "air"
density 1.1934
        acoustic
        nonlinear
        c0 343.2048
        B_over_A 0.4
END

GDSW
   solver_tol 1.0e-8
   prt_summary 0
END
```

9.31. Thermally Induced Elastic Waves: Hollow Sphere

Refer to Section 3.2 for details of the test.

9.31.1. Sierra SD Input Deck

```
// Based on "Thermal Stress-Wave Propagation in Hollow Elastic Spheres" - Tsui and Kraus (1965)
// Note that this choice of gamma implies an unphysically small G and/or an unphysically large kappa

// shear modulus: G = mu = {mu=G=shear_mod=25/4}
// Poisson ratio: nu = {nu=poisson_ratio=1/3}
// density = Beta = {Beta=density=1}
// pressure wave speed: c = sqrt( 2*(1-nu)*G/(density*(1-2*nu)) ) = {c = sqrt( 2*(1-nu)*G/(density*(1-2*nu)) )}
// inner sphere radius: a = {a=1}
// outer sphere radius: b = {b=1}
// thermal diffusivity: kappa = {kappa=1}
// dimensionless inertia parameter: gamma = kappa/(c*a) = {gamma = kappa/(c*a)} <-- should be 1/5 to match figure 1
// coefficient of thermal expansion: alpha = {alpha=coeff_thermal_expansion=1e-2}
// Ta = {Ta=1} <-- from aria input deck

SOLUTION
    solver=gdsw
case t2
  transient
  nsteps 200
  time_step 1e-3
  nUpdateTemperature 1
END

FILE
  geometry_file 'hollow_sphere.e'
END

FUNCTION xStar
  type analytic
  expression variable dx = nodal DispX
  evaluate expression "dx*{(1-nu)/(a*alpha*Ta*(1+nu))}"
END

USER OUTPUT
  compute nodal xStar as function xStar
END

FUNCTION rho
  type analytic
  expression variable c = coord
  evaluate expression = "c[1]/{a}"
END

USER OUTPUT
  compute nodal rho as function rho
END

#FUNCTION tau
#  type analytic
#  expression variable time = time
#  evaluate expression = "{kappa}*time/({a}*{a})"
#END
#
#USER OUTPUT
#  compute nodal tau as function tau
#END

OUTPUTS
force
```

```
disp
elmat
temperature
thermal_strain

nodal variables xStar
nodal variables rho
#nodal variables tau
END

ECHO
END

BOUNDARY
  sideset 1
    x=0
  sideset 2
    y=0
  sideset 3
    z=0
END

PARAMETERS
  thermal_exo_var = "TND"
END

DAMPING
  alpha 1.0e-3
  beta  1.0e-3
END


LOADS
  body
    thermal
    function 8245
END

function 8245
    type linear
    data 0 1
    data 1 1
end

BLOCK 1
        material "foo"
END

MATERIAL "foo"
        G = {shear_mod}
        density = {density}
        nu = {poisson_ratio}
        alphat = {coeff_thermal_expansion}
        tref 0.0
END
```

9.31.2. Aria Input Deck

```
begin sierra Calore

  title heat conduction through concentric sphere, test of dash contact


  Begin Aria Material mat1
```

```
  # kappa = k/(cp*rho)
  density  = constant rho = 1
  specific heat = constant    cp = 1
  thermal conductivity = constant k = 1
  heat conduction = basic
End

BEGIN FINITE ELEMENT MODEL test
  Database Name = hollow_sphere.exo
  use material  mat1 for block_1
END FINITE ELEMENT MODEL test

BEGIN TPETRA EQUATION SOLVER solve_temperature
  BEGIN BICGSTAB SOLVER
    BEGIN JACOBI PRECONDITIONER
    END
    CONVERGENCE TOLERANCE = 1.0e-8
    MAXIMUM ITERATIONS = 1000
    RESIDUAL SCALING = NONE
  END
END

begin procedure myProcedure

    Begin Solution Control Description
       Use System Main
       Begin System Main
          Simulation Start Time = 0.0
          simulation Termination Time = 0.2
          begin transient timeblock
             advance myRegion1
          end
       End

       Begin Parameters For Transient TimeBlock
          Start Time = 0.0
          Begin Parameters For Aria Region myRegion1
             Time Step Variation     = Adaptive
             Time Integration Method = First_Order
             Initial time step size  = 1e-3
             Maximum Time Step Size Ratio  = 1.5
             minimum resolved time step size = 1.e-4
             minimum time step size = 1.e-4
             maximum time step size = 1e-2
             Predictor Order = 1
             Predictor-corrector tolerance = 1e-3
             Predictor-Corrector Begin After Step = 4
          End
       End

    End

  begin Aria region myRegion1

    Begin Results Output Label diffusion output1
      database Name = hollow_sphere.e
      At Step 0, Increment = 1
      Title Aria Dash Tied Contact Test
      Nodal Variables = solution->temperature as TND
    End

    ############################
    ### boundary conditions ###
    ############################

    Begin Temperature Boundary Condition  t1
      temperature = 1.0
      add surface surface_1000
```

```
      End
      Begin Temperature Boundary Condition  t2
        temperature = 0.0
        add surface surface_2000
      End

      IC for Temperature on all_blocks = constant value = 0

      use finite element model test
      $ model coordinates are model_coordinates
      use linear solver solve_temperature

      nonlinear solution strategy = newton

      maximum nonlinear iterations = 10
      nonlinear residual tolerance = 1.0e-8
      nonlinear relaxation factor  = 1.0
      use dof averaged nonlinear residual
      accept solution after maximum nonlinear iterations = true

       EQ Energy for Temperature On all_blocks Using Q1 With Lumped_Mass Diff

    end

  end procedure myProcedure

end sierra Calore
```

9.32. Euler Beam Bending

Refer to Section 4.1

```
SOLUTION
    solver=gdsw
statics
title 'single beam model. 100 elements. xy only'
lumped
END

FILE
geometry_file 100.exo
END

BOUNDARY
    nodeset 1
    fixed
    nodeset 3
      x = 0
      z = 0
      rotx = 0
      roty = 0
END

LOADS
nodeset 2
force = 0. .25 0.
END

OUTPUTS
     deform
END

ECHO
END



BLOCK 1
material 'Aluminum'
Beam2
Area 0.1
orientation 0 .1 0
I1 .2
I2 .3
J .5
END

Material 'Aluminum'
E 10.0E6
nu 0.33
density 253.82e-6
END
```

9.33. Euler Beam Properties

Refer to Section 4.1.1

```
//salinas input for 100 element beam
SOLUTION
    solver=gdsw
  title='multi-element beam modal'
  eigen
  nmodes=10
  shift=-1
END

FILE
   geometry_file 'beam2.exo'
END

PARAMETERS
END

BOUNDARY
  nodeset 1
  fixed
END

LOADS
END

OUTPUTS
        disp
END

HISTORY
    disp
    block '1'
END

ECHO
      mass
END

BLOCK 1
   Beam2
   material=1
   Area=0.03
   I1=0.09
   I2=0.01
   J=0.1
   orientation = 1 1 0
END

MATERIAL  1
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 1
END
```

9.34. A Navy Beam

Refer to Section 4.1.2

```
//salinas input for 100 element beam
SOLUTION
    solver=gdsw
  title='multi-element beam modal'
  eigen
  nmodes=10
  shift=-1
END

FILE
   geometry_file 'nbeam.exo'
END

PARAMETERS
END

BOUNDARY
  nodeset 1
  fixed
END

LOADS
END

OUTPUTS
        disp
END

HISTORY
    disp
    block '1'
END

ECHO
      mass
END

BLOCK 1
   Nbeam
   material=1
   Area=0.03
   I1=0.09
   I2=0.01
   J=0.1
   orientation = 1 1 0
END

MATERIAL  1
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 1
END
```

9.35. Preloaded Beam

Refer to Section 4.1.3

```
SOLUTION
 case statics
   statics
   load = 1
 case update
    tangent
 case modal
    eigen
    nmodes = 10
end
FILE
    geometry_file  'kgperm3.exo'
end
LOAD 1
    nodeset 2
    force 44482 0 0
end
OUTPUTS
    disp
    force
end
ECHO
    mass
    mass=block
end
BOUNDARY
   nodeset 1
       fixed




end
BLOCK 1
    beam2
    material="steel"
    area=0.0000202683
    i1 = 3.2690739721e-11
    i2 = 3.2690739721e-11
    j = 6.5381479442e-11

end
MATERIAL "steel"
    E 187e9
    nu .3
    density 8015.19
end
```

9.35.1. Prescribed Displacement

Refer to Subsection 4.1.4

```
SOLUTION
    solver=gdsw
 case statics
   statics
   load = 1
 case update
    tangent
 case modal
    eigen
    nmodes = 30
END
FILE
    geometry_file  'Beam.exo'
end
LOAD 1
    nodeset 2
        force 1e10 0 0
end
OUTPUTS
    disp
    force
end
ECHO
    mass
    mass=block
end
BOUNDARY
   nodeset 1
      x=0 y=0
   nodeset 3
      x=1.3368983957E-01 y=0
   nodeset 4
    z=0 rotx=0 roty=0
end
BLOCK 1
    beam2
    material="steel"
    area=4
    i1=1.33333
    i2=1.33333
    j=2.6666
    orientation 0 0 1
end
MATERIAL "steel"
    E 187e9
    nu .3
    density 8015.19
end
```

9.36. Membrane Quad

Refer to Section 4.2.1

```
SOLUTION
    solver=gdsw
  case two
  eigen
  lumped
  nmodes 14
  shift = -1.e8
END
File
  geometry_file 'temp1/Membrane_quad.par'
end
Boundary
  nodeset 1 x=0 y=0
  nodeset 2 x=0 y=0
end

Loads
end
Outputs
  deform
end
Block 100
  QuadTM
  material "steel"
  thickness 0.1
end
Material "steel"
  E 10.0
  nu 0.49
  density 1.0e-9
end
```

9.37. QuadM membrane Patch

Refer to Section 4.2.2

```
SOLUTION
    solver=gdsw
   statics
END
FILE
geometry_file 'model.exo'
end
BOUNDARY
  nodeset 2 fixed
  nodeset 3 x=0
  nodeset 1 x=0.1
  nodeset 4 z=0
  nodeset 5 y=0
end
LOADS
end
OUTPUTS
deform
stress
end
BLOCK 1
  QuadM
material "steel"
thickness 1.0
end
BLOCK 2
  QuadM
  material "steel"
  thickness 1.0
end
BLOCK 3
  QuadM
  material "steel"
  thickness 1.0
end
BLOCK 4
  QuadM
  material "steel"
  thickness 1.0
end
BLOCK 5
  QuadM
  material "steel"
  thickness 1.0
end
BLOCK 6
  QuadM
  material "steel"
  thickness 1.0
end
BLOCK 7
  QuadM
material "steel"
  thickness 1.0
end
MATERIAL "steel"
E 30e6
nu 0.3
density 0.288
end
```

9.38. Membrane Geometrical Stiffness

Refer to Section 4.2.7

```
SOLUTION
    solver=gdsw
   case transfer
     receive_sierra_data
   case eig
     eigen nmodes=all
     lumped
END

file
   geometry_file membrane_geometric.exo
//   geometry_file plate101.exo
end

$$
boundary
  nodeset 1
    y=0
  nodeset 4
    x=0
  nodeset 2
    y=0
  nodeset 5
    z=0
end

boundary
end

loads
end

block 100
   QuadM
//   thickness=0.1
//   thickness = 0.095435875007294
   thickness = from_transfer
   material=1
end

material 1
  e=10.
  nu=0.49
//  density=1e-9
  density = 1.047823996923137e-9
end

outputs
   disp
   mfile
end
parameters
    mfile_format 3column
end
echo
  mass
end
```

9.39. QuadS_GY Shear Membrane Shell

Refer to Section 4.2.8

```
SOLUTION
    solver=gdsw
   eigen
   nmodes = 20
   shift = -1.e5
   lumped
END

GDSW
  solver_tol = 1e-10
end

Parameters
   wtmass 0.00259
end

FILE
    geometry_file   'mesh_quadt.g'
end

BOUNDARY
  nodeset 1 rotx=0 roty=0 rotz=0 x=0 y=0 z=0
  nodeset 2 rotx=0 roty=0 rotz=0 x=0 y=0 z=0
  nodeset 3 rotx=0 roty=0 rotz=0 x=0 y=0 z=0
  nodeset 4 rotx=0 roty=0 rotz=0 x=0 y=0 z=0
end

OUTPUTS
   globals
end

ECHO
end

BLOCK 1
  QuadS_GY
  material "steel"
  thickness 0.001
  fiber orientation = 0
end

MATERIAL "steel"
  orthotropic_layer
  E1 = 30e6
  E2 = 0.5e6
  nu12 = 0.3
  G12 = 0.5e6
  density 0.288
end
```

9.40. QuadS_GY Shear Membrane Shell - Geometric Stiffness and Preload

Refer to Section 4.2.9

```
SOLUTION
    solver=gdsw
 case one
     receive_sierra_data
     lumped
 case out
    eigen
    nmodes = 10
END


FILE
//     geometry_file  'shell_beam.exo'
    geometry_file 'shell_adagio_salinas.e'
END

LOAD 1
    sideset 1
    pressure -2245852908.28
    // pressure 0
END

OUTPUTS

    disp
    force
END

ECHO
    mass=block
END

BOUNDARY
   nodeset 1
       fixed
   nodeset 3
    z=0
//    y=0
END

BLOCK 1
    material "steel"
    QuadS_GY
    thickness = 0.004450425122033
END

MATERIAL "steel"
        E 187e9
        nu 0.3
        density 8015.19
END
```

9.41. Partial Cylinder Patch

Refer to Section 4.3.1

```
$ Algebraic Preprocessor (Aprepro) version 6.25 (2023/10/12)
SOLUTION
       statics
END

FILE
  geometry_file 'cyl_q4.g'
END

BOUNDARY
  nodeset 100
     x=0
  nodeset 200
     y=0
  nodeset 300
     z=0
  nodeset 301
     z=0.01
  nodeset 1000
     rotx=0
     roty=0
     rotz=0
END

LOADS
END

OUTPUTS
  eorient
  strain
  stress
  disp
  energy
  genergies
END

ECHO
  genergies
END

BLOCK 1000
  MATERIAL "STEEL"
$ loop
  THICKNESS 0.01
$
$
END

MATERIAL "STEEL"
  E 1.0E+6
  NU 0.3
  DENSITY 1.0E-6
END
```

9.42. Thin Plate Bending

Refer to Section 4.3.2 for details of the test.

```
//salinas input created using nasgen from nastran file 'bending.nas'
SOLUTION
    solver=gdsw
  title=' NEi Nastran Static Analysis Set'
   statics
END

GDSW
    solver_tol=1.0e-10
END

FILE
   geometry_file 'bending.exo'
END


PARAMETERS
// wtmass=0.00259
   eigen_norm=visualization
END

BOUNDARY
    nodeset 11  // nastran SID=1
        fixed
END

LOADS
  sideset 1
      pressure 1.0
END

OUTPUTS
        disp
        stress
        genergies
END

ECHO
      input
      mass
END

BLOCK 13
   material=1
   thickness=0.5
   { QUAD }
END

MATERIAL        1
   Isotropic
   E = 3e+07
   NU = 0.3
   density = 0.0007324
END
```

9.43. Two Layered Hexshell

Refer to Section 4.3.3

```
SOLUTION
    solver=gdsw
  title 'Two-ply_rectangular_plate_pressure_A'
  statics
END
FILE
  geometry_file 'Two-ply_rectangular_plate.g'
end
PARAMETERS
  wtmass = 0.00259
end
BOUNDARY
  nodeset 1
    y = 0.0
  node_list_file=node1.txt
    x = 0.0
    z = 0.0
  node_list_file=node2.txt
    x = 0.0
end
LOADS
  sideset 1
  pressure 2.0
end
OUTPUTS
  disp
  eorient
  force
end
ECHO
  mass block
  disp
end
BLOCK 1
  HexShell
   tcoord 0 2
   layer 1
   material 1
   thickness .5
   layer 2
   material 2
   thickness .5
end
MATERIAL 1
  density 0.1
  E 1.0e5
  nu 0.25
end
MATERIAL 2
  density 0.05
  E 1.0e3
  nu 0.2
end
```

9.44. Hex Membrane Sandwich

Refer to Section 4.4

```
SOLUTION
    solver=gdsw
   statics
END
File
geometry_file 'Model_hex.exo'
end
Boundary
  nodeset 2 fixed
  nodeset 3 x=0
  nodeset 1 x=0.1
  nodeset 4 z=0
  nodeset 5 y=0
end
Loads
end
Outputs
deform
stress
end

Block 1
  Hex8u
material "steel"
end
Block 2
  Hex8u
  material "steel"
end
Block 3
  Hex8u
  material "steel"
end
Block 4
  Hex8u
  material "steel"
end
Block 5
  Hex8u
  material "steel"
end
Block 6
  Hex8u
  material "steel"
end
Block 7
  Hex8u
material "steel"
end
MATERIAL "steel"
E 30e6
nu 0.3
density 0.288
end
```

9.45. Higher Order Hex Acoustic Element Convergence

Refer to Section 4.5

```
SOLUTION
    solver=gdsw
   eigen
   nmodes 10
   shift = -1.e4
END

FILE
geometry_file 'temp1/wg_hex2.par'
// geometry_file '1/wg_hex5.par'
END

BOUNDARY
END

LOADS
END

OUTPUTS
  globals
END

ECHO
END


BLOCK 1
pelement
order=3
material "steel"
END

MATERIAL "steel"
acoustic
c0 332.0
density 1.3
END

PARAMETERS
  usepelements
END

GDSW
  solver_tol 1.0e-12
  orthog 0
  sc_option no
END
```

9.46. Higher Order Tet Acoustic Element Convergence

Refer to Section 4.6

```
SOLUTION
    solver=gdsw
   eigen
   nmodes 10
   shift = -1.e3
END

FILE
geometry_file 'temp1/wg_tet2.par'
END

BOUNDARY
END

LOADS
END

OUTPUTS
   globals
END

ECHO
END


BLOCK 1
pelement
order=3
material "steel"
END

MATERIAL "steel"
acoustic
c0 332.0
density 1.3
END

PARAMETERS
  usepelements
END

GDSW
  solver_tol 1.0e-12
  orthog 0
  sc_option no
END
```

9.47. Superelement Damping

Refer to Section 4.7.1

```
SOLUTION
    solver=gdsw
  transient
    time_step 2.0e-5
    nsteps 250
END

FILE
  geometry_file 'full_system.exo'
END

BOUNDARY
    sideset 3
  fixed
END

LOADS
  sideset 4
  pressure = 10.0
  function = 1
END

FUNCTION 1
  type linear
  data 0 0
  data 1e-4 1
  data 3e-4 -1
  data 4e-4 0
  data 10 0
END

HISTORY
  nodeset 3
    disp
    stress
END

OUTPUTS
  deform
    elemeigchecks
END

// the following element block is hex.
// exodus tells us it is an 8-node hex.
// The default integration mode is "UNDER"
// The only required argument is the material card
BLOCK 1
  material "steel"
        hex8u
END

MATERIAL "steel"
  E 30e6
  nu .3
  density 0.288
END

BLOCK 3 // Formerly block 2
   superelement
   file=SE_DampTwoBlock.ncf
   map locations
END
```

9.48. Sensitivity Analysis with a Superelement

Refer to Section 4.7.2

```
SOLUTION
    solver=gdsw
  title 'sensitivity of system with CBR model'
  eigen nmodes=20
END
FILE
   geometry_file blade1_residual_se.exo
   omit block 13
end
PARAMETERS
   eigen_norm=visualization
end
BOUNDARY
    nodeset 11
        fixed
end


OUTPUTS
        disp
end
ECHO
      mass
      input
end
BLOCK 13
   // 1 element of type SHELL. 4 nodes/element
   // property card 'PSHELL   1      '
   material=3001
   thickness=0.111
end
BLOCK 17
   // 2 elements of type SPHERE-MASS. 1 node/element
   // no property card
   ConMassA
   // 'CONM2    4800719'
   mass=11268.5
   Ixx=0
   Iyy=0
   Izz=0
   Ixy=0
   Ixz=0
   Iyz=0
   offset=0 0 0
end
BLOCK 480000
   // 6 elements of type BEAM. 2 nodes/element
   // property card 'PBAR     48000  '
   material=48001
   Area=0.05693
   I1=0.00374
   I2=0.00374
   J=0.00749
end
BLOCK 480020
   // 14 elements of type TRIANGLE. 3 nodes/element
   // property card 'PSHELL   48002  '
   material=48000
   thickness=0.0254
end
BLOCK 480023
   // 209 elements of type SHELL. 4 nodes/element
```

```
   // property card 'PSHELL   48002  '
   material=48000
   thickness=0.0254
end
BLOCK 480024
    superelement
    file='blade1_se.ncf'
    diagnostic=0
    sensitivity_param 1 = 2.04e11  // E
    sensitivity_param 2 = 8017.2
    map locations
end
MATERIAL  3001
   // from 'MAT1     3001   '
   Isotropic
   E=2e+11
   NU=0.3
   density=7861.06
end
MATERIAL  48000
   // from 'MAT1     48000  '
   Isotropic
   E=2e+11
   NU=0.29
   density=7860
end
MATERIAL  48001
   // from 'MAT1     48001  '
   Isotropic
   E=2e+11
   NU=0.29
   density=7860
end
```

9.49. Superelement Superposition

Refer to Section 4.7.3 for details of the test.

9.49.1. Full Model

```
//salinas input created using nasgen from nastran file 'trusses-4.bdf'
SOLUTION
  title=' MSC.Nastran job created on 02-Apr-12 at 16:56:43'
  case full
  transient
     time_step=1e-5
     nsteps=1000
     load=1
END

FILE
   geometry_file 'truss_full.exo'
END


BOUNDARY
    nodeset 11  // nastran SID=1
        x=0
    nodeset 32  // nastran SID=3
        y=0
    nodeset 33  // nastran SID=3
        z=0
END

LOAD 1
   node_list_file 'endtruss_node_list'
      force 1 0 0
      function=1
END

function 1
    type=linear
    data 0 0
    data 1e-3 1
    data 4e-3 -1
    data 5e-3 0
end

OUTPUTS
        disp
END

ECHO
      mass
END


BLOCK 12
   material=1
   Area=0.01
   Truss
END

MATERIAL  1
   // from 'MAT1     1      '
   Isotropic
   E=1e+07
   NU=0
   density=0.1
```

```
END
```

9.49.2. CB Reduction

```
//salinas input created using nasgen from nastran file 'trusses-4.bdf'
SOLUTION
  title=' MSC.Nastran job created on 02-Apr-12 at 16:56:43'
  case basis
   cbr nmodes=1
   lumped
END

FILE
   geometry_file 'endtruss.exo'
END

CBMODEL
  nodeset 11
  format=netcdf
  file=endtruss.ncf
  inertia_matrix=yes
END

BOUNDARY
    nodeset 32  // nastran SID=3
        y=0
    nodeset 33  // nastran SID=3
        z=0
END

LOADS
END

OUTPUTS
        disp
        genergies
END

ECHO
      mass
END


BLOCK 12
   // 4 elements of type TRUSS. 2 nodes/element
   // property card 'PROD     1      '
   material=1
   Area=0.01
   Truss
END

MATERIAL  1
   // from 'MAT1     1      '
   Isotropic
   E=1e+07
   NU=0
   density=0.1
END
```

9.49.3. System Analysis with Superelement

```
//salinas input created using nasgen from nastran file 'trusses-4.bdf'
SOLUTION
  title='2 residual trusses, and a superelement'
```

```
  transient
     time_step=1e-5
     nsteps=1000
END

FILE
   geometry_file 'truss_se.exo'
END


BOUNDARY
    nodeset 11  // nastran SID=1
        x=0
    nodeset 32  // nastran SID=3
        y=0
    nodeset 33  // nastran SID=3
        z=0
END

LOADS
   node_list_file 'endtruss_node_list'
      force 1 0 0
      function=1
END

function 1
    type=linear
    data 0 0
    data 1e-3 1
    data 4e-3 -1
    data 5e-3 0
end

OUTPUTS
        disp
END

ECHO
      mass
END


BLOCK 12
   // 2 elements of type TRUSS. 2 nodes/element
   // property card 'PROD     1       '
   material=1
   Area=0.01
   Truss
END

block 13
    superelement
    map = locations
    file = endtruss.ncf
END

MATERIAL  1
   // from 'MAT1     1       '
   Isotropic
   E=1e+07
   NU=0
   density=0.1
END
```

9.50. Superelement Inertia Tensor Input

Refer to Section 4.7.4 for details of the test.

```
{include(beam_model.inp)}#

outputs
 genergies
 eorient
   disp
end


cbmodel
  nodeset 1
  format=dmig
  file=cbr.dmig
  inertia_matrix=yes
end
```

9.50.1. Beam Model

```
//#########################################################################################
//#
//# This salinas input file was generated by lsdyna2sierra
//#
//#########################################################################################

SOLUTION
    solver=gdsw
 title 'beam_med'
 cbr
 nmodes = 10
 correction=vectors
 rbmdof=123456
END

file
 geometry_file 'temp1/beam_med.exo'
end

echo
 mass
end

block 1
 material 'boxsolid'
end

material 'boxsolid'
 isotropic
 e = 207
 nu = 0.300000
 density = 0.0000071
end
```

9.51. Nastran/Sierra/SD Interoperability with Superelements

Refer to Section 4.7.6 for details of the test.

9.51.1. Sierra/SD *Full Model*

```
//salinas input created using nasgen from nastran file 'tuningforkz.bdf'
SOLUTION
  title=' MSC.Nastran job created on 28-Nov-17 at 08:50:40'
  case eig
    eigen
        nmodes=10
        shift=-1e6
   case frf
      modalfrf
      load=1
   case trn
      modaltransient
      time_step=2e-5
      nsteps=200
      load=30
END

FILE
   geometry_file 'tuningforkz.exo'
END


PARAMETERS
// wtmass=0.00259
   eigen_norm=visualization
END

BOUNDARY
    nodeset 11  // nastran SID=1
        x=0 y=0 z=0
    nodeset 53  // nastran SID=4
        z=0
END

LOAD 1
   sideset 1
      pressure 1
      function 1
END

function 1
   type linear
   data 0 1
   data 1e4 1
end

LOAD 30
   sideset 1
      pressure 1
      function 30
END

function 30
   type linear
   data 0 0
   data 0.5e-3 1
   data 1e-3 0
end
```

```
damping
   gamma=0.03
end

frequency
   freq_min   1
   freq_step  1
   freq_max   1000
   nodeset 43
   displacement
end

OUTPUTS
        disp
END

ECHO
      mass
END

BLOCK 11
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   1      '
   material=1
END

BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2      '
   material=1
END

BLOCK 31
   // 4 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   3      '
   material=1
END

MATERIAL  1
   // from 'MAT1     1      '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.51.2. Nastran Full Model

```
$ NASTRAN input file created by the Patran 2010.2.3 64-Bit input file
$ translator on November  28, 2017 at 08:52:49.
$ Direct Text Input for Nastran System Cell Section
$ Direct Text Input for File Management Section
$ Direct Text Input for Executive Control
$ Linear Static Analysis, Database
SOL 101
CEND
$ Direct Text Input for Global Case Control Data
TITLE = MSC.Nastran job created on 28-Nov-17 at 08:50:40
ECHO = NONE
SUBCASE 1
   SUBTITLE=no-bc-on-interface
   SPC = 2
   LOAD = 2
   DISPLACEMENT(SORT1,REAL)=ALL
   SPCFORCES(SORT1,REAL)=ALL
   STRESS(SORT1,REAL,VONMISES,BILIN)=ALL
```

```
$ Direct Text Input for this Subcase
BEGIN BULK
$ Direct Text Input for Bulk Data
PARAM    POST    0
PARAM   PRTMAXIM YES
$ Elements and Element Properties for region : load-tine
PSOLID   1       1       0
$ Pset: "load-tine" will be imported as: "psolid.1"
CHEXA    10      1       36      12      8       30      72      52
         48      71      157     103     158     159     160     104
         96      161     162     108     163     164
CHEXA    11      1       37      36      30      31      74      72
         71      73      165     159     166     167     168     160
         161     169     170     164     171     172
CHEXA    12      1       38      37      31      32      76      74
         73      75      173     167     174     175     176     168
         169     177     178     172     179     180
CHEXA    13      1       39      38      32      33      78      76
         75      77      181     175     182     183     184     176
         177     185     186     180     187     188
CHEXA    14      1       40      39      33      34      80      78
         77      79      189     183     190     191     192     184
         185     193     194     188     195     196
$ Elements and Element Properties for region : rom-tine
PSOLID   2       1       0
$ Pset: "rom-tine" will be imported as: "psolid.2"
CHEXA    5       2       24      6       14      18      60      47
         55      59      117     111     118     119     120     97
         113     121     122     116     123     124
CHEXA    6       2       25      24      18      19      62      60
         59      61      125     119     126     127     128     120
         121     129     130     124     131     132
CHEXA    7       2       26      25      19      20      64      62
         61      63      133     127     134     135     136     128
         129     137     138     132     139     140
CHEXA    8       2       27      26      20      21      66      64
         63      65      141     135     142     143     144     136
         137     145     146     140     147     148
CHEXA    9       2       28      27      21      22      68      66
         65      67      149     143     150     151     152     144
         145     153     154     148     155     156
$ Elements and Element Properties for region : fork
PSOLID   3       1       0
$ Pset: "fork" will be imported as: "psolid.3"
CHEXA    1       3       4       3       1       2       44      41
         42      43      81      82      83      84      85      86
         87      88      89      90      91      92
CHEXA    2       3       8       4       2       6       48      44
         43      47      93      84      94      95      96      85
         88      97      98      92      99      100
CHEXA    3       3       12      11      4       8       52      49
         44      48      101     102     93      103     104     105
         85      96      106     107     98      108
CHEXA    4       3       6       2       13      14      47      43
         54      55      94      109     110     111     97      88
         112     113     99      114     115     116
$ Referenced Material Records
$ Material Record : aluminum
$ Description of Material : Date: 27-Nov-17           Time: 08:57:23
MAT1     1      1.+7            .3      2.59-4
$ Nodes of the Entire Model
GRID     1               0.      0.      0.
GRID     2              1.       0.      0.
GRID     3               0.     1.       0.
GRID     4              1.      1.       0.
GRID     6              2.       0.      0.
GRID     8              2.      1.       0.
GRID     11             1.      2.       0.
```

```
GRID     12             2.      2.       0.
GRID     13             1.      -1.      0.
GRID     14             2.      -1.      0.
GRID     18             3.      -1.      0.
GRID     19             4.      -1.      0.
GRID     20             5.      -1.      0.
GRID     21             6.      -1.      0.
GRID     22             7.      -1.      0.
GRID     24             3.       0.      0.
GRID     25             4.       0.      0.
GRID     26             5.       0.      0.
GRID     27             6.       0.      0.
GRID     28             7.       0.      0.
GRID     30             3.      1.       0.
GRID     31             4.      1.       0.
GRID     32             5.      1.       0.
GRID     33             6.      1.       0.
GRID     34             7.      1.       0.
GRID     36             3.      2.       0.
GRID     37             4.      2.       0.
GRID     38             5.      2.       0.
GRID     39             6.      2.       0.
GRID     40             7.      2.       0.
GRID     41              0.     1.      .5
GRID     42              0.      0.     .5
GRID     43             1.       0.     .5
GRID     44             1.      1.      .5
GRID     47             2.       0.     .5
GRID     48             2.      1.      .5
GRID     49             1.      2.      .5
GRID     52             2.      2.      .5
GRID     54             1.      -1.     .5
GRID     55             2.      -1.     .5
GRID     59             3.      -1.     .5
GRID     60             3.       0.     .5
GRID     61             4.      -1.     .5
GRID     62             4.       0.     .5
GRID     63             5.      -1.     .5
GRID     64             5.       0.     .5
GRID     65             6.      -1.     .5
GRID     66             6.       0.     .5
GRID     67             7.      -1.     .5
GRID     68             7.       0.     .5
GRID     71             3.      1.      .5
GRID     72             3.      2.      .5
GRID     73             4.      1.      .5
GRID     74             4.      2.      .5
GRID     75             5.      1.      .5
GRID     76             5.      2.      .5
GRID     77             6.      1.      .5
GRID     78             6.      2.      .5
GRID     79             7.      1.      .5
GRID     80             7.      2.      .5
GRID*    81                             .5              1.
*       -1.46144-8
GRID*    82                             -5.9644-10      .5
*       -6.88986-9
GRID*    83                             .5              -2.08654-9
*       -4.99122-9
GRID*    84                             1.              .5
*       -5.11059-9
GRID     85             1.      1.      .25
GRID*    86                             4.25148-10      1.
*       .25
GRID*    87                             2.65251-9       7.2653-9
*       .25
GRID*    88                             1.              3.57292-9
*       .25
```

```
GRID     89             .5      1.      .5
GRID*    90                             4.44603-9       .5
*       .5
GRID*    91                             .5              6.6322-9
*       .5
GRID     92             1.      .5      .5
GRID*    93                             1.5             1.
*       -1.46144-8
GRID*    94                             1.5             -2.08654-9
*       -4.99122-9
GRID*    95                             2.              .5
*       -5.11059-9
GRID     96             2.      1.      .25
GRID*    97                             2.              3.57292-9
*       .25
GRID     98             1.5     1.      .5
GRID*    99                             1.5             6.6322-9
*       .5
GRID     100            2.      .5      .5
GRID*    101                            1.5             2.
*       -1.46144-8
GRID*    102                            1.              1.5
*       -6.88986-9
GRID*    103                            2.              1.5
*       -5.11059-9
GRID     104            2.      2.      .25
GRID     105            1.      2.      .25
GRID     106            1.5     2.      .5
GRID     107            1.      1.5     .5
GRID     108            2.      1.5     .5
GRID*    109                            1.              -.5
*       -6.88986-9
GRID*    110                            1.5             -1.
*       -4.99122-9
GRID*    111                            2.              -.5
*       -5.11059-9
GRID     112            1.      -1.     .25
GRID     113            2.      -1.     .25
GRID     114            1.      -.5     .5
GRID     115            1.5     -1.     .5
GRID     116            2.      -.5     .5
GRID*    117                            2.5             1.5939-8
*       -1.46144-8
GRID*    118                            2.5             -1.
*       -4.99122-9
GRID*    119                            3.              -.5
*       -5.11059-9
GRID*    120                            3.              1.94968-8
*       .25
GRID     121            3.      -1.     .25
GRID*    122                            2.5             1.6149-9
*       .5
GRID     123            2.5     -1.     .5
GRID     124            3.      -.5     .5
GRID*    125                            3.5             1.5939-8
*       -1.46144-8
GRID*    126                            3.5             -1.
*       -4.99122-9
GRID*    127                            4.              -.5
*       -5.11059-9
GRID*    128                            4.              1.94968-8
*       .25
GRID     129            4.      -1.     .25
GRID*    130                            3.5             1.6149-9
*       .5
GRID     131            3.5     -1.     .5
GRID     132            4.      -.5     .5
GRID*    133                            4.5             1.5939-8
```

```
*       -1.46144-8
GRID*    134                        4.5             -1.
*       -4.99122-9
GRID*    135                        5.              -.5
*       -5.11059-9
GRID*    136                        5.              1.94968-8
*       .25
GRID     137            5.      -1.     .25
GRID*    138                        4.5             1.6149-9
*       .5
GRID     139            4.5     -1.     .5
GRID     140            5.      -.5     .5
GRID*    141                        5.5             1.5939-8
*       -1.46144-8
GRID*    142                        5.5             -1.
*       -4.99122-9
GRID*    143                        6.              -.5
*       -5.11059-9
GRID*    144                        6.              1.94968-8
*       .25
GRID     145            6.      -1.     .25
GRID*    146                        5.5             1.6149-9
*       .5
GRID     147            5.5     -1.     .5
GRID     148            6.      -.5     .5
GRID*    149                        6.5             1.5939-8
*       -1.46144-8
GRID*    150                        6.5             -1.
*       -4.99122-9
GRID*    151                        7.              -.5
*       -5.11059-9
GRID*    152                        7.              1.94968-8
*       .25
GRID     153            7.      -1.     .25
GRID*    154                        6.5             1.6149-9
*       .5
GRID     155            6.5     -1.     .5
GRID     156            7.      -.5     .5
GRID*    157                        2.5             2.
*       -1.46144-8
GRID*    158                        2.5             1.
*       -4.99122-9
GRID*    159                        3.              1.5
*       -5.11059-9
GRID     160            3.      2.      .25
GRID     161            3.      1.      .25
GRID     162            2.5     2.      .5
GRID     163            2.5     1.      .5
GRID     164            3.      1.5     .5
GRID*    165                        3.5             2.
*       -1.46144-8
GRID*    166                        3.5             1.
*       -4.99122-9
GRID*    167                        4.              1.5
*       -5.11059-9
GRID     168            4.      2.      .25
GRID     169            4.      1.      .25
GRID     170            3.5     2.      .5
GRID     171            3.5     1.      .5
GRID     172            4.      1.5     .5
GRID*    173                        4.5             2.
*       -1.46144-8
GRID*    174                        4.5             1.
*       -4.99122-9
GRID*    175                        5.              1.5
*       -5.11059-9
GRID     176            5.      2.      .25
GRID     177            5.      1.      .25
```

```
GRID     178            4.5     2.      .5
GRID     179            4.5     1.      .5
GRID     180            5.      1.5     .5
GRID*    181                            5.5             2.
*       -1.46144-8
GRID*    182                            5.5             1.
*       -4.99122-9
GRID*    183                            6.              1.5
*       -5.11059-9
GRID     184            6.      2.      .25
GRID     185            6.      1.      .25
GRID     186            5.5     2.      .5
GRID     187            5.5     1.      .5
GRID     188            6.      1.5     .5
GRID*    189                            6.5             2.
*       -1.46144-8
GRID*    190                            6.5             1.
*       -4.99122-9
GRID*    191                            7.              1.5
*       -5.11059-9
GRID     192            7.      2.      .25
GRID     193            7.      1.      .25
GRID     194            6.5     2.      .5
GRID     195            6.5     1.      .5
GRID     196            7.      1.5     .5
$ Loads for Load Case : no-bc-on-interface
SPCADD   2       1       3       4       5
LOAD     2      1.      1.       1
$ Displacement Constraints of Load Set : base
SPC1     1       123     1       3       41      42      82      86
         87      90
$ Displacement Constraints of Load Set : interface
SPC1     3       1       6       14      47      55      97      111
         113     116
$ Displacement Constraints of Load Set : otm
SPC1     4       3       67
$ Displacement Constraints of Load Set : z0
SPC1     5       3       1       2       3       4       6       8
         11      12      13      14      18      19      20      21
         22      24      25      26      27      28      30      31
         32      33      34      36      37      38      39      40
         81      82      83      84      93      94      95      101
         102     103     109     110     111     117     118     119
         125     126     127     133     134     135     141     142
         143     149     150     151     157     158     159     165
         166     167     173     174     175     181     182     183
         189     190     191
$ Pressure Loads of Load Set : pressure
PLOAD4   1       14      1.                              40      78
$ Referenced Coordinate Frames
ENDDATA
```

9.52. Sierra/SD Superelement File Formats

Refer to Section 4.7.10 for details of the test.

9.52.1. Sierra/SD *Full Model*

```
//salinas input created using nasgen from nastran file 'tuningfork.bdf'
SOLUTION
  title=' MSC.Nastran job created on 27-Nov-17 at 09:29:23'
   eigen
       nmodes=10
       shift=-1e6   // needed only for floating
END

FILE
   geometry_file 'tuningfork.exo'
END


PARAMETERS
// wtmass=0.00259
   eigen_norm=visualization
END

BOUNDARY
    nodeset 11  // nastran SID=1
        x=0
    nodeset 12  // nastran SID=1
        y=0
    nodeset 13  // nastran SID=1
        z=0
    nodeset 33  // nastran SID=3
        z=0
END

history
  nodeset 100
  disp
end

OUTPUTS
  disp
END

ECHO
      mass
END


BLOCK 11
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   1       '
   material=1
END

BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2       '
   material=1
END

BLOCK 31
   // 4 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   3       '
   material=1
```

```
END

MATERIAL  1
   // from 'MAT1     1      '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.52.2. Netcdf Output

```
SOLUTION
  title='ROM tine of tuning fork'
  cbr
    nmodes=10
END

cbmodel
   nodeset 41
   format=netcdf
   file=rom4.ncf
end

FILE
   geometry_file 'rom4.exo'
END


PARAMETERS
   eigen_norm=visualization
END

BOUNDARY
    nodeset 33  // nastran SID=3
        z=0
END

LOADS
END

OUTPUTS
        disp
END

ECHO
      mass
END


BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2      '
   material=1
   blkbeta=1e-6
END

MATERIAL  1
   // from 'MAT1     1      '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.52.3. DMIG Output

```
SOLUTION
  title='ROM tine of tuning fork'
  cbr
    nmodes=10
END

cbmodel
   nodeset 41
   format=dmig*
   file=rom4.dmig
   inertia_matrix=no
end

FILE
   geometry_file 'rom4.exo'
END


PARAMETERS
   eigen_norm=visualization
END

BOUNDARY
    nodeset 33  // nastran SID=3
        z=0
END

LOADS
END

OUTPUTS
        disp
END

ECHO
      mass
END


BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2       '
   material=1
   blkbeta=1e-6
END

MATERIAL  1
   // from 'MAT1     1       '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.52.4. Netcdf Input

```
SOLUTION
  title='Residual calculations using a CBR/ROM of right tine'
  case eigNCF
    eigen
        nmodes=10
        shift=-1e6
END
```

```
FILE
   geometry_file 'residual.exo'
END

gdsw
    solver_tol=1e-12
end

PARAMETERS
   eigen_norm=visualization
END

BOUNDARY
    nodeset 11  // nastran SID=1
        x=0 y=0 z=0
    nodeset 33  // nastran SID=3
        z=0
END

LOAD 30
   sideset 1
      pressure 1
      function 30
END

function 30
   type linear
   data 0 0
   data 0.5e-3 1
   data 1e-3 0
end

damping
   gamma=0.03
end

frequency
   freq_min   1
   freq_step  1
   freq_max   1000
   nodeset 33
   displacement
end

history
  nodeset 100
  disp
end

OUTPUTS
        disp
END

ECHO
      mass
END


BLOCK 11
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   1      '
   material=1
END

$$ BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2      '
   material=1
```

```
END

BLOCK 31
   // 4 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   3       '
   material=1
END

BLOCK 32
   superelement
   map=locations
   file=rom4.ncf
END

MATERIAL  1
   // from 'MAT1     1       '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.52.5. DMIG Input

```
SOLUTION
  title='Residual calculations using a CBR/ROM of right tine'
  case eigDMIG
    eigen
        nmodes=10
        shift=-1e6
END

FILE
   geometry_file 'residual.exo'
END

gdsw
    solver_tol=1e-12
end

PARAMETERS
   eigen_norm=visualization
END

BOUNDARY
    nodeset 11  // nastran SID=1
        x=0 y=0 z=0
    nodeset 33  // nastran SID=3
        z=0
END

LOAD 30
   sideset 1
      pressure 1
      function 30
END

function 30
   type linear
   data 0 0
   data 0.5e-3 1
   data 1e-3 0
end

damping
   gamma=0.03
```

```
end

frequency
   freq_min   1
   freq_step  1
   freq_max   1000
   nodeset 33
   displacement
end

OUTPUTS
        disp
END

history
  nodeset 100
  disp
end

ECHO
      mass
END


BLOCK 11
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   1       '
   material=1
END

$$ BLOCK 21
   // 5 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   2       '
   material=1
END

BLOCK 31
   // 4 elements of type HEX. 20 nodes/element
   // property card 'PSOLID   3       '
   material=1
END

BLOCK 32
   superelement
   format = dmig
   file=rom4.dmig
END

MATERIAL  1
   // from 'MAT1     1       '
   Isotropic
   E = 1e+07
   NU = 0.3
   density = 0.000259
END
```

9.53. Parallel Distribution of Load through Rbars

Refer to Section 5.1

```
solution
 case eig
   eigen nmodes=4
   shift -1e8
   enforce_modeshape_residual = false
 case out
   modalranvib
   keepmodes=3   // force modal truncation
   lfcutoff=-10
  title 'hex and spiders'
end
ranloads
   matrix=1
   load=1
     nodeset 1
force=0 1 0
        scale=1000
end
frequency
   freq_step=100
   freq_min=300
   freq_max=1e4
   BLOCK=all
   accel
end
matrix-function 1
   Name=input_psd
   symmetry=symmetric
   dimension=1x1
   data 1,1
     real function 1
end
function 1
   Name='psd'
   type=loglog
   data 1.0 1e-8
   data 299 1e-8
   data 300 0.01
   data 2000 1
   data 8000 1.
   data 10000 0.01
   data 10001 1e-8
end
damping
  gamma=0.01
end
parameters
 wtmass=0.00259
end
file
     geometry_file 'hex_spider.exo'
end
boundary
  nodeset 1
  roty=0 rotz=0 rotx=0 x=0 z=0
end
loads
end
outputs
        disp
vrms
end
```

```
echo
      mass=block
        mass
end

gdsw
  solver_tol 1e-9
end

block 1
   material 1
end
block 2
  ConMass
    Mass=0.7075
    Ixx =0
    Ixy =0
    Iyy =0
    Ixz =0
    Iyz =0
    Izz =0
    Offset= 0 0 0 // patran/exo type 'BEAM'/BEAM. Number nodes 2
end
block 10
    RBAR  // RBE type element
end
material 1
   density=0.283
   E=29e2
   nu=0.3
end
```

9.54. Perfectly Matched Layers: Offset Sphere

Refer to Section 5.2 for details of the test.

```
SOLUTION
  directfrf
end
FILE
   geometry_file 'OffsetSphere3.exo'
end
FREQUENCY
  freq_min = 100.0
  freq_step = 1
  freq_max = 101.0
  disp
  block 1
end
LOADS
    sideset 1
    acoustic_vel = 1.0
    function = 2
end
BOUNDARY
  sideset 2
   pml_element
   use block 57
end
FUNCTION 2
    type LINEAR
    data 0 1
    data 1e6 1
end
OUTPUTS
  apressure
end
BLOCK 1
    material "air"
end
BLOCK 57
   pml_element
   stack_depth 5
   ellipsoid_dimensions 5 5 5
   pml_thickness 1
   loss_function = polynomial
   loss_params = 0 960 960 0
end
MATERIAL "air"
  density 1.293
  acoustic
  c0 332.0
end
GDSW
   precondUpdateFreq 3
   prt_debug 1
   useBarrierTimers yes
end
```

9.55. Periodic Boundary Conditions

Refer to Section 5.3 for details of the test.

```
SOLUTION
    solver=gdsw
    statics
END

FILE
    geometry_file 'SingleVoidCenterPbc.exo'
END

BOUNDARY
    nodeset 1
       x=0 y=0 z=0
    nodeset 2
       x=0 z=0
    nodeset 2
       x=0
END

BEGIN-PERIODIC
    side a = 1
    side b = 2
    name = X_directional_PBC
    search tolerance = 1e-2
    geometric offset = -1.0 0.0 0.0
    Ux = -1.5
END

LOADS
END

HISTORY
    element stress nearest location  0.0 0.4 0.0 as ExpectedMaxStress1
    element stress nearest location  0.0 0.0 0.4 as ExpectedMaxStress2
    element stress nearest location -0.4 0.0 0.0 as ExpectedMinStress
END

OUTPUTS
    displacement
    stress
END

ECHO
  MPC
END

BLOCK 1
    material "simple_solid"
END

MATERIAL "simple_solid"
    E 100.0
    nu 0.3
    density 1.0
END
```

9.56. Multidirectional Periodic BC: Periodic Volume Elements

Refer to Section 5.4 for details of the test.

```
SOLUTION
    solver=gdsw
    statics
END

FILE
    geometry_file 'SingleVoidCenterPve.exo'
END

BOUNDARY
    nodeset 1
       x=0 y=0 z=0
END

BEGIN-PERIODIC
    side a = 1
    side b = 2
    name = X_directional_PBC
    search tolerance = 1e-4
//    geometric offset = 1.0 0.0 0.0
    Ux = 1.5
    Uy = 1.0
    Uz = 0.5
END

BEGIN-PERIODIC
    side a = 3
    side b = 4
    name = Y_directional_PBC
    search tolerance = 1e-4
//    geometric offset = 0.0 1.0 0.0
    Ux = 1.0
    Uy = -1.0
    Uz = 0.25
END

BEGIN-PERIODIC
    side a = 5
    side b = 6
    name = Z_directional_PBC
    search tolerance = 1e-4
//    geometric offset = 0.0 0.0 1.0
    Ux = 0.5
    Uy = 0.25
    Uz = -0.5
END

LOADS
END

HISTORY
    element stress nearest location  -0.1 -0.5 -0.5 as Stress1
    element stress nearest location  -0.5 -0.1 -0.5 as Stress2
    element stress nearest location  -0.5 -0.5 -0.1 as Stress3
END

OUTPUTS
    displacement
    stress
END

ECHO
```

```
  MPC
END

BLOCK 1
    material "simple_solid"
END

MATERIAL "simple_solid"
    E 100.0
    nu 0.3
    density 1.0
END
```

9.57. Filter Rigid Modes from Loads

Refer to Section 5.5

```
SOLUTION
    solver=gdsw
  statics
END


FILE
geometry_file 'temp1/beam_hex8.par'
// geometry_file 'beam_hex8.exo'
END

BOUNDARY
END

PARAMETERS
  FilterRbmLoad=allStructural
  rbmtolerance=1e-6
  num_rigid_mode 6
END

GDSW
  prt_summary = 3
END

LOADS
        sideset 1
              traction = 0 1000.0 0
END

OUTPUTS
disp
  force
  rhs
END

ECHO
 input off
END


BLOCK 1
material "steel"
         hex8u
END

MATERIAL "steel"
E 30.0e6
nu 0.0
density 0.288
END
```

9.58. Modal Force on a Biplane Model

Refer to Section 5.6 for details of the test.

```
SOLUTION
    solver=gdsw
   case eig2
      eigen
      shift=-1e5
      nmodes=30
   case two
     modaltransient
     nsteps 260
     time_step 1e-3
     load=10
END

FILE
   geometry_file 'biplane.exo' // 'biplane.exo'
END

LOAD 10
  body
    modalforce
    function 60
END

FUNCTION 60
  type table
    tablename 35
END

TABLE 35
   dimension 2
   size 260 30
   delta 1e-3 1
   origin 1e-3 0
   datafile=ModalForces.txt
END

TIED DATA
  surface 1,6 //tail stalk to main body
END
TIED DATA
  surface 2,7 //top fin to top of tailstalk
END

{include(blocks_and_materials.inp)}
{include(common.inp)}
```

9.59. Lighthill Analogy - Helmholtz Resonator

Refer to Section 5.7 for details of the test.

```
SOLUTION
    solver=gdsw
  transient
    time_step 0.5e-3
    nsteps 500
END
File
   geometry_file temp1/lighthill_helmholtz_resonator_ns.par
end
Loads
  nodeset 1
  lighthill = 1.0
  function = 1
end
Damping
alpha 50
end
Function 1
  type exodusread
  nodeset 1
  name "divT"
  exo_var vector divT
  interp = linear
end
Boundary
sideset 13 absorbing radius = 100
end
History
   node_list_file nodelist1873
   aforce
   apressure
end

Outputs
end

Block 1
material 1
end

Block 2
material 1
end

MATERIAL 1
acoustic
density 1.2256e-3
c0 34300 // cm/s
end

Tied Data
surface 1, 10
End

Tied Data
surface 2, 11
End

Tied Data
surface 3, 12
End
```

9.60. Lighthill Tensor Verification Input

Refer to Section 5.8 for details of the test.

```
SOLUTION
    solver=gdsw
    solver=gdsw
  transient
  time_step 0.5
  lumped_consistent
  nsteps 600
END

LINESAMPLE
  samples per line 1000
  endpoint -500. 0. 0. 500. 0. 0.
  format exodus
END

FILE
  geometry_file temp1/lighthill_waveguide_1000x1x1_pulse.par
END

LOADS
  nodeset 1
  lighthill = 1
  function = 1
END

FUNCTION 1
  type exodusread
  nodeset 1
  name "divT"
  exo_var vector divT
  interp = linear
END

BOUNDARY
  sideset 1
  absorbing
END

OUTPUTS
  database name = temp1/james.exo
  apressure
  aforce
END

BLOCK 1
  material 1
END

MATERIAL 1
  acoustic
  density 1
  c0 1
END
```

9.61. Acoustic Point Source in Frequency Domain

Refer to Section 5.9 for details of the test.

```
Solution
    directfrf
End

File
   geometry_file 'point_source.exo'
End

Frequency
  freq_min = 1.0
  freq_step = 10.0
  freq_max = 150.0
  sideset 1
  pressure
End

Loads
    nodeset 1
    point_volume_vel = 1.0
    function = 2
End

Damping
  beta 1.0e-5
End

Boundary
  sideset 1
    absorbing
    radius 2.0
End

Function 2
       type LINEAR
       name "test_func1"
       data 0.0 1.0
       data 5.0e9  1.0
End

Outputs
 apressure
End

Block 1
    material "air"
End

Material "air"
        density 1.293
        acoustic
        c0 343.0
End

GDSW
  solver_tol 1.0e-8
  max_previous_sols 10
  cull_method eigen
  orthog 40
  num_GS_steps 2
End
```

9.62. Acoustic Point Source in Time Domain

Refer to Section 5.10 for details of the test.

```
SOLUTION
     transient
     time_step 1.0e-4
     nsteps 1000
     rho 0.7
END

FILE
   geometry_file 'point_source.exo'
END

LOADS
    nodeset 1
    point_volume_vel = 1.0
    // point_volume_accel = 1.0
    function = 1
END

DAMPING
    beta 1.0e-5
END

BOUNDARY
  sideset 1
    absorbing
    radius 2.0
END

FUNCTION 1
   type analytic
   evaluate expression = "omega = pi * 50; sin(omega*time)"
END

OUTPUTS
 apressure
END

ECHO
END

BLOCK 1
material "air"
END

MATERIAL "air"
        density 1.293
        acoustic
        c0 343.0
END


GDSW
  solver_tol 1.0e-10
END
```

9.63. Acoustic Plane Wave Scattering in Frequency Domain

Refer to Section 5.11 for details of the test.

```
# rho0 = {rho0 = 1.21}
# c0 = {c0 = 343.0}
# vscale = {vscale = 1/(rho0*c0)}

SOLUTION
    directfrf
    scattering
    solver=gdsw
END

FILE
    geometry_file mie/cylinderScatterer.exo
END

Frequency
    freq_min = 1000.0
    freq_step =100.0
    freq_max = 1000.0
    block 1,2
    apressure
    disp
End

LOADS
    sideset 2
        acoustic_vel = 1.0
        scale = {vscale}
        function = 1
    sideset 2
        iacoustic_vel = 1.0
        scale = {vscale}
        function = 2
    sideset 3
        pressure = 1
        scale = {vscale}
        function = 1
    sideset 3
        ipressure = 1
        scale = {vscale}
        function = 2
END

BOUNDARY
    nodeset 1
        z = 0
    sideset 1
        pml_element
        use block 326
        hex
END

BLOCK 326
    pml_element
    stack_depth 20
    source_origin 0 0 0
    ellipsoid_dimensions 0.8 0.8 1000
    pml_thickness 0.00025
    loss_function = polynomial
    loss_params = 0 6000 6000 6000
END

Function 1
```

```
    type plane_wave_freq
    Material "air"
    Direction 1.0 0.0 0.0
    Origin 0 0 0
END
Function 2
    type iplane_wave_freq
    Material "air"
    Direction 1.0 0.0 0.0
    Origin 0 0 0
END

TIED DATA
    Surface 2,3
END

OUTPUTS
    deform
    apressure
END

ECHO
END

BLOCK 1
    material "air"
END

BLOCK 2
    material "steel"
END

MATERIAL "air"
    acoustic
    density {rho0}
    c0 {c0}
END

MATERIAL "steel"
    E 19.5e10
    nu 0.3
    density 7700.0
END

GDSW
    solver_tol 1.0e-11
    overlap 1
    SC_optionH yes
    max_iter 100
END
```

9.64. Transient Reaction Forces

Refer to Section 5.12 for details of the test.

9.64.1. Vibration from Initial Conditions

```
OUTPUTS
    database name = initCond.e
    disp
    velocity
    accel
    force
    reaction_force
END

BOUNDARY
    nodeset 2 3 4 5
        fixed
END

LOAD 1
END

INITIAL-CONDITIONS
  velocity = by_block
END
```

9.64.2. Prescribed Acceleration

```
OUTPUTS
    database name = accel.e
    disp
    velocity
    accel
    force
    reaction_force
END

BOUNDARY
    nodeset 2 3 4 5
        fixed
    nodeset 1
        accelx 1
        function xaccel
        disp0 = 0
        vel0 = 0
    nodeset 1
        accely 1
        function yaccel
        disp0 = 0
        vel0 = 0
END

FUNCTION xaccel
    type = linear
    data 0 4
    data 1 4
END

FUNCTION yaccel
    type = linear
    data 0 8
    data 1 8
```

```
END

DAMPING
  alpha = 0.1
  beta = 0.2
END

LOAD 1
    nodeset 1
       force 0 0 1.5
END
```

9.65. Relative Displacement PSD

Refer to Section 6.1 for details of the test.

9.65.1. In Phase Response

9.65.1.1. Nodal closest distance user output

```
Solution
  solver=gdsw
  case eig
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file nodeAndQuad.exo
end
Boundary
  nodeset 1,2 y = 0 z = 0 rotx = 0 roty = 0 rotz = 0
end

//Rigidset
//  sideset 1
//end

user output
  nodeset 2
  compute nodal bar as closest distance to block 1
end

Outputs
  disp
  vrms
  bar
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force 1 0 0
    scale 1e6
  load=2
    nodeset 1
    force 1 0 0
    scale 1e6
end

Matrix-function matFun
  symmetry hermitian
  dimension 2x2
  data 1,1
    real function squareBand scale 1
  data 1,2
    real function squareBand scale 1
  data 2,2
    real function squareBand scale 1
end
Function squareBand
  type linear
```

```
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
Frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 1 2 disp
  bar
end
Block 2
  conmass
  mass 1e2
end
Block 3
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
  ky elastic 1e5
  kz elastic 1e5
end
Block 22
  joint2g
  kx elastic 1e10
  ky elastic 1e10
  kz elastic 1e10
end
Block 1
  material aluminum
  thickness = 0.1
end

Material aluminum
  E = 1e7
  nu = 0.35
  density = 0.1
end
```

9.65.1.2. Element relative_disp output

```
Solution
  solver=gdsw
  case eig
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file oneD.exo
end
Boundary
  nodeset 1,2 y = 0 z = 0 rotx = 0 roty = 0 rotz = 0
```

```
end
Outputs
  disp
  relative_disp
  vrms
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force 1 0 0
    scale 1e6
  load=2
    nodeset 1
    force 1 0 0
    scale 1e6
end

Matrix-function matFun
  symmetry hermitian
  dimension 2x2
  data 1,1
    real function squareBand scale 1
  data 1,2
    real function squareBand scale 1
  data 2,2
    real function squareBand scale 1
end
Function squareBand
  type linear
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
Frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 1 2 disp
  block 12 relative_disp
end
Block 1 2
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
  ky elastic 1e5
  kz elastic 1e5
end
```

9.65.2. Opposite Phase Response

9.65.2.1. Nodal closest distance user output

```
Solution
  solver=gdsw
  case eig
```

```
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file nodeAndQuad45.exo
end
Boundary
  nodeset 1,2 z = 0 rotx = 0 roty = 0 rotz = 0
end

user output
  nodeset 2
  compute nodal bar as closest distance to block 1
end

Outputs
  disp
  vrms
  bar
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force {sqrt(2)/2} {sqrt(2)/2} 0
    scale 1e6
  load=2
    nodeset 1
    force {sqrt(2)/2} {sqrt(2)/2} 0
    scale 1e6
end

Matrix-function matFun
  symmetry hermitian
  dimension 2x2
  data 1,1
    real function squareBand scale 1
  data 1,2
    real function squareBand scale -1
  data 2,2
    real function squareBand scale 1
end
Function squareBand
  type linear
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
Frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 1 2 disp
  bar
end
Block 2
  conmass
```

```
  mass 1e2
end
Block 3
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
  ky elastic 1e5
  kz elastic 1e5
end
Block 22
  joint2g
  kx elastic 1e05
  ky elastic 1e05
  kz elastic 1e05
end
Block 1
  material aluminum
  thickness = 0.1
end

Material aluminum
  E = 1e7
  nu = 0.35
  density = 0.1
end
```

9.65.2.2. Nodal closest distance user output

```
Solution
  solver=gdsw
  case eig
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file oneD.exo
end
Boundary
  nodeset 1,2 y = 0 z = 0 rotx = 0 roty = 0 rotz = 0
end
Outputs
  disp
  relative_disp
  vrms
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force 1 0 0
    scale 1e6
  load=2
    nodeset 1
    force 1 0 0
    scale 1e6
end

Matrix-function matFun
```

```
  symmetry hermitian
  dimension 2x2
  data 1,1
    real function squareBand scale 1
  data 1,2
    real function squareBand scale -1
  data 2,2
    real function squareBand scale 1
end
Function squareBand
  type linear
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
Frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 1 2 disp
  block 12 relative_disp
end
Block 1 2
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
  ky elastic 1e5
  kz elastic 1e5
end
```

9.65.3. One Node Fixed Response

9.65.3.1. Nodal closest distance user output

```
Solution
  solver=gdsw
  case eig
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file nodeAndQuad.exo
end
Boundary
  nodeset 1 fixed
  nodeset 2 y = 0 z = 0 rotx = 0 roty = 0 rotz = 0
end

user output
  nodeset 2
  compute nodal bar as closest distance to block 1
```

```
end

Outputs
  disp
  vrms
  bar
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force 1 0 0
    scale 1e6
end
Matrix-function matFun
  symmetry hermitian
  dimension 1x1
  data 1,1
    real function squareBand scale 1
end
Function squareBand
  type linear
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 2 disp
  bar
end
Block 2
  conmass
  mass 1e2
end
Block 3
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
  ky elastic 1e5
  kz elastic 1e5
end
Block 22
  joint2g
  kx elastic 1e05
  ky elastic 1e05
  kz elastic 1e05
end
Block 1
  material aluminum
  thickness = 0.1
end

Material aluminum
  E = 1e7
  nu = 0.35
  density = 0.1
```

```
end
```

9.65.3.2. Element relative_disp output

```
Solution
  solver=gdsw
  case eig
    eigen nmodes=all
  case random
    modalranvib
    lfcutoff -1
end
Parameters
  wtmass 0.00259
end
File
  geometry_file oneD.exo
end
Boundary
  nodeset 1 fixed
  nodeset 2 y = 0 z = 0 rotx = 0 roty = 0 rotz = 0
end
Outputs
  disp
  relative_disp
  vrms
end
Ranloads
  matrix matFun
  load=1
    nodeset 2
    force 1 0 0
    scale 1e6
end
Matrix-function matFun
  symmetry hermitian
  dimension 1x1
  data 1,1
    real function squareBand scale 1
end
Function squareBand
  type linear
  data 1       1e-6
  data 9.9999  1e-6
  data 10      1e-1
  data 25      1e-1
  data 25.0001 1e-6
  data 30      1e-6
end
Damping
  gamma = 0.05
end
frequency
  freq_min 10
  freq_max 25
  freq_step 0.1
  nodeset 2 disp
  block 12 relative_disp
end
Block 1 2
  conmass
  mass 1e2
end
Block 12
  joint2g
  kx elastic 1e5
```

```
  ky elastic 1e5
  kz elastic 1e5
end
```

9.65.4. Tuning Fork Response

```
Solution
  solver gdsw
  case eig
    eigen
    nmodes 12
    shift -100
  case mRand
   modalranvib
   truncationMethod none
   lfcutoff -10
end
GDSW
  max_numterm_c1 1000
end
Parameters
  wtmass 0.00259
end
File
  geometry_file tuningFork.exo
end
Damping
  gamma 0.08
end
Boundary
  block 1
    z=0
  nodeset 1
    x=0 y=0
end
Matrix-function 1x1
  symmetry hermitian
  dimension=1x1
  data 1,1
    real function 2 scale 1
end
Function 2
  type linear
  data 1.000000e-16 1.000000e-16
  data 1.99999999 1.000000e-16
  data 2.0 1.000000e-01
  data 100.0 1.000000e-01
  data 100.00000001 1.000000e-16
  data 125.0 1.000000e-16
end
Ranloads
  matrix 1x1
  load = 1
  nodeset 2
  force 0 1 0
  scale 1
end
Outputs
  disp
  relative_disp
end
Frequency
  freq_min 1
  freq_max 150
  freq_step 0.1
  block '100 10 11 12 13 14 15 16 17 18 19'
```

```
  nodeset all
  relative_disp
  disp
end
Block 1
  material 1
end
Material 1
  e 1e7
  nu 0.3
  density 0.1
end
Block 100
  Joint2g
  kx elastic 0
  ky elastic 0
  kz elastic 0
  nsm 1e-4
end
Tied Joint
 normal definition none
 surface 100, 200
 connect to block 100
 side average
end

//{ind=0}
{loop(10)}

Block {10+ind}
  joint2g
  kx elastic 0
  ky elastic 0
  kz elastic 0
  nsm 1e-4
end
tied joint
 normal definition none
 surface {10+ind}, {20+ind}
 connect to block {10+ind}
 side average // do not stiffen the surface
end

//{ind++}
{endloop}
```

9.66. Projection of Gauss Point Stresses to Nodes

Refer to Section 6.2 for details of the tests.

9.66.1. Exact Projection of Linearly Varying Stresses

```
SOLUTION
     statics
END

FILE
  geometry_file {mesh_file}
END

BOUNDARY
  nodeset 1
    x = 0
  nodeset 2
    y = 0
  nodeset 3
    z = 0
END

LOADS
  body
  gravity = 1 2 3
END

USER OUTPUT
  compute nodal stress_xx_error as function stress_xx_error_func
  compute nodal stress_yy_error as function stress_yy_error_func
  compute nodal stress_zz_error as function stress_zz_error_func
  compute nodal stress_xy_error as function stress_xy_error_func
  compute nodal stress_xz_error as function stress_xz_error_func
  compute nodal stress_yz_error as function stress_yz_error_func
  compute nodal disp_x_error as function disp_x_error_func
  compute nodal disp_y_error as function disp_y_error_func
  compute nodal disp_z_error as function disp_z_error_func
  compute global max_stress_xx_error as maxabs of nodal stress_xx_error
  compute global max_stress_yy_error as maxabs of nodal stress_yy_error
  compute global max_stress_zz_error as maxabs of nodal stress_zz_error
  compute global max_stress_xy_error as maxabs of nodal stress_xy_error
  compute global max_stress_xz_error as maxabs of nodal stress_xz_error
  compute global max_stress_yz_error as maxabs of nodal stress_yz_error
  compute global max_disp_x_error as maxabs of nodal disp_x_error
  compute global max_disp_y_error as maxabs of nodal disp_y_error
  compute global max_disp_z_error as maxabs of nodal disp_z_error
END

FUNCTION stress_xx_error_func
  type = analytic
    expression variable coord = coord
    expression variable stress_xx = nodal nodalStressX
    evaluate expression = "stress_xx - (-0.288*coord[1] + 0.72)"
END

FUNCTION stress_yy_error_func
  type = analytic
    expression variable coord = coord
    expression variable stress_yy = nodal nodalStressY
    evaluate expression = "stress_yy - (-0.576*coord[2] + 0.3168)"
END

FUNCTION stress_zz_error_func
  type = analytic
```

```
    expression variable coord = coord
    expression variable stress_zz = nodal nodalStressZ
    evaluate expression = "stress_zz - (-0.864*coord[3] + 0.3888)"
END

FUNCTION stress_xy_error_func
  type = analytic
    expression variable stress_xy = nodal nodalStressXY
    evaluate expression = "stress_xy"
END

FUNCTION stress_xz_error_func
  type = analytic
    expression variable stress_xz = nodal nodalStressXZ
    evaluate expression = "stress_xz"
END

FUNCTION stress_yz_error_func
  type = analytic
    expression variable stress_yz = nodal nodalStressYZ
    evaluate expression = "stress_yz"
END

FUNCTION disp_x_error_func
  type = analytic
    expression variable coord = coord
    expression variable dispx = nodal DispX
    evaluate expression = "dispx - (-0.144*coord[1]*coord[1] + 0.72*coord[1] + 2.7)/30.0e6"
END

FUNCTION disp_y_error_func
  type = analytic
    expression variable coord = coord
    expression variable dispy = nodal DispY
    evaluate expression = "dispy - (-0.288*coord[2]*coord[2] + 0.3168*coord[2] + 0.26136)/30.0e6"
END

FUNCTION disp_z_error_func
  type = analytic
    expression variable coord = coord
    expression variable dispz = nodal DispZ
    evaluate expression = "dispz - (-0.432*coord[3]*coord[3] + 0.3888*coord[3] + 0.26244)/30.0e6"
END

OUTPUTS
  disp
  stress = nodes
  stress_xx_error
  stress_yy_error
  stress_zz_error
  stress_xy_error
  stress_xz_error
  stress_yz_error
  disp_x_error
  disp_y_error
  disp_z_error
  max_stress_xx_error
  max_stress_yy_error
  max_stress_zz_error
  max_stress_xy_error
  max_stress_xz_error
  max_stress_yz_error
  max_disp_x_error
  max_disp_y_error
  max_disp_z_error
END

ECHO
```

```
END

BLOCK 1
  material 1
//  cutet10
END

MATERIAL 1
  E 30e6
  nu 0.0
  density 0.288
END
```

9.66.2. Exact Projection of Principal and Von Mises Stresses

```
SOLUTION
     statics
END

FILE
  geometry_file {mesh_file}
END

// coordinate system for 30 degree rotation about axis in direction [1 2 3]
// ux = 1/sqrt(14), uy = 2/sqrt(14), uz = 3/sqrt(14), th = pi/6
// R_{11} = cos(th) + ux^2*(1 - cos(th))
// R_{22} = cos(th) + uy^2*(1 - cos(th))
// R_{33} = cos(th) + uz^2*(1 - cos(th))
// R_{12} = ux*uy*(1 - cos(th)) - uz*sin(th)
// R_{23} = uy*uz*(1 - cos(th)) - ux*sin(th)
// R_{31} = uz*ux*(1 - cos(th)) - uy*sin(th)
// R_{13} = ux*uz*(1 - cos(th)) + uy*sin(th)
// R_{21} = uy*ux*(1 - cos(th)) + uz*sin(th)
// R_{32} = uz*uy*(1 - cos(th)) + ux*sin(th)
// unit vector along new z-axis is (R_{13}, R_{23}, R_{33})
// unit vector along new x-axis is (R_{11}, R_{21}, R_{31})
BEGIN RECTANGULAR COORDINATE SYSTEM cs1
  ORIGIN = 0.0 0.0 0.0
  Z POINT = 0.295970083958616 -0.076212936863829 0.952151929923014
  XZ POINT = 0.875595017799836 0.420031090899431 -0.238552399866233
END

BOUNDARY
  nodeset 1
    x = 0
    coordinate cs1
  nodeset 2
    y = 0
    coordinate cs1
  nodeset 3
    z = 0
    coordinate cs1
END

LOADS
  body
  gravity = 1 10 100
  coordinate cs1
END

USER OUTPUT
  compute nodal xt as function xt_func
  compute nodal yt as function yt_func
  compute nodal zt as function zt_func
  compute nodal sigxx as function sigxx_func
  compute nodal sigyy as function sigyy_func
```

```
  compute nodal sigzz as function sigzz_func
  compute nodal sigvm_error as function sigvm_error_func
  compute nodal prin_stress_min_error as function prin_stress_min_error_func
  compute nodal prin_stress_mid_error as function prin_stress_mid_error_func
  compute nodal prin_stress_max_error as function prin_stress_max_error_func
  compute global max_sigvm_error as maxabs of nodal sigvm_error
  compute global max_prin_stress_min_error as maxabs of nodal prin_stress_min_error
  compute global max_prin_stress_mid_error as maxabs of nodal prin_stress_mid_error
  compute global max_prin_stress_max_error as maxabs of nodal prin_stress_max_error
END

FUNCTION xt_func
  type = analytic
    expression variable coord = coord
    evaluate expression = "0.875595017799836*coord[1] + 0.420031090899431*coord[2] + -0.238552399866233*coord[3]"
END

FUNCTION yt_func
  type = analytic
    expression variable coord = coord
    evaluate expression = "-0.381752634837842*coord[1] + 0.904303859846028*coord[2] + 0.191048305048596*coord[3]"
END

FUNCTION zt_func
  type = analytic
    expression variable coord = coord
    evaluate expression = "0.295970083958616*coord[1] + -0.076212936863829*coord[2] + 0.952151929923014*coord[3]"
END

FUNCTION sigxx_func
  type = analytic
    expression variable xt = nodal xt
    evaluate expression = "-1.0*(xt - 0.5)"
END

FUNCTION sigyy_func
  type = analytic
    expression variable yt = nodal yt
    evaluate expression = "-10.0*(yt - 0.5)"
END

FUNCTION sigzz_func
  type = analytic
    expression variable zt = nodal zt
    evaluate expression = "-100.0*(zt - 0.5)"
END

FUNCTION sigvm_error_func
  type = analytic
    expression variable sigxx = nodal sigxx
    expression variable sigyy = nodal sigyy
    expression variable sigzz = nodal sigzz
    expression variable sigvm = nodal nodalVonMises
    evaluate expression = "sqrt(0.5*(pow(sigxx-sigyy,2)+pow(sigyy-sigzz,2)+pow(sigzz-sigxx,2)))-sigvm"
END

FUNCTION prin_stress_max_error_func
  type = analytic
    expression variable sigxx = nodal sigxx
    expression variable sigyy = nodal sigyy
    expression variable sigzz = nodal sigzz
    expression variable prin_max = nodal nodal_max_principal_stress
    evaluate expression = "max_all = max(max(sigxx, sigyy), sigzz); max_all - prin_max"
END

FUNCTION prin_stress_min_error_func
  type = analytic
    expression variable sigxx = nodal sigxx
```

```
    expression variable sigyy = nodal sigyy
    expression variable sigzz = nodal sigzz
    expression variable prin_min = nodal nodal_min_principal_stress
    evaluate expression = "min_all = min(min(sigxx, sigyy), sigzz); min_all - prin_min"
END

FUNCTION prin_stress_mid_error_func
  type = analytic
    expression variable sigxx = nodal sigxx
    expression variable sigyy = nodal sigyy
    expression variable sigzz = nodal sigzz
    expression variable prin_mid = nodal nodal_intermediate_principal_stress
    evaluate expression = "max_all = max(max(sigxx, sigyy), sigzz); min_all = min(min(sigxx, sigyy), sigzz); ((sigxx != min_all) 
END

OUTPUTS
  disp
  stress = nodes
  xt
  yt
  zt
  sigxx
  sigyy
  sigzz
  sigvm_error
  prin_stress_max_error
  prin_stress_min_error
  prin_stress_mid_error
  max_sigvm_error
  max_prin_stress_min_error
  max_prin_stress_mid_error
  max_prin_stress_max_error
END

ECHO
END

BLOCK 1
  material 1
//  cutet10
END

MATERIAL 1
  E 30e6
  nu 0.0
  density 1.0
END

GDSW
  solver_tol = 1e-10
END
```

9.67. RigidSet Compared to Rbar

Refer to Section 7.3

```
SOLUTION
    solver=gdsw
  eigen
    nmodes 50
  shift -1e6
END

PARAMETERS
 wtmass=0.00259
END

FILE
   geometry_file 'rigidset.exo'
   omit block 2,3
END

OUTPUTS
  displacement
END

ECHO
  mass block
END

RIGIDSET set1
  sideset 1
END

RIGIDSET set2
  sideset 2
END

GDSW
  max_numterm_C1 500
  overlap 2
  krylov_method 1
//   orthog_option 3
END

BLOCK 1
   material 1
   hex8b
END

BLOCK 2:3
  beam2
END

MATERIAL 1
   density 0.3
   E = 3.0e7
   nu = 0.3
END
```

9.68. Multiple Tied-Surfaces and Curved Surfaces

Refer to Section 7.4

```
SOLUTION
    solver=gdsw
  title 'tied surface example with holes'
  eigen
  nmodes 15
  shift = -1e8
END
Parameters
  RemoveRedundancy=yes
  wtmass=0.00259
end

File
  geometry_file 'tied_surface.exo'
end
Outputs
  displacement
  vonmises
  Constraint_Info
end

Block 1
   material 1
end

Block 2
  material 1
end

Block 10
  material 1
end

Material 1
   E=1.0e7
   nu=0.33
   density=0.098
end

TIED DATA
   surface 2,1
   transverse tied
   search tolerance 1.e-3
   edge tolerance   1.e-5
end
TIED DATA
   surface 102,2
   gap removal = on
   search tolerance 1.e-3
   edge tolerance   1.e-6
end
TIED DATA
   surface 3,101
   search tolerance 1.e-1
   edge tolerance   1.e-6
end
GDSW
  con_tolerance 1e-2
  max_numterm_C1=6
end
```

9.69. Tied-Joint with Joint2G and Spring

Inputs for comparison of manually generated constraints with TiedJoint.

9.69.1. Manual Constraints

```
SOLUTION
    solver=gdsw
 eigen
 nmodes=20
 shift=-1e7
END

FILE
geometry_file lap_simple.exo
  omit block 3
END

OUTPUTS
disp
END

ECHO
mpc
END

BLOCK 1,2
material "mat"
END

BLOCK 3
  material "mat" // dead
  area = 3.1416E+00
  I1 = 7.8540E-01
  I2 = 7.8540E-01
  J = 1.5708E+00
END

Block 33
   spring
    kx=20776000
        ky=20776000
        kz=26080000
END

HISTORY
 sideset 1,2
 displacement
END

Rigidset 1
    sideset 1
end
Rigidset 2
    sideset 2
end

MPC
 254 x 1
 207 x -1
END

MPC
 254 y 1
 207 y -1
```

```
END

MPC
 254 z 1
 207 z -1
END

MPC
 253 x 1
 58 x -1
END

MPC
 253 y 1
 58 y -1
END

MPC
 253 z 1
 58 z -1
END

MATERIAL "mat"
E 200e9
nu 0.3
    density 7800
END

BOUNDARY
    sideset 3
       fixed
END

LOADS
    sideset 4
        pressure = -1e3
        function = 1
END

FUNCTION 1
       type LINEAR
       name "const_one"
       data 0.0 1.0
       data 2.0e4 1.0
END
```

9.69.2. Tied Joint Constraints

Refer to Section 7.1 for details of the test.

```
SOLUTION
    solver=gdsw
 eigen
 nmodes=20
 shift=-1e7
restart=write
END

FILE
    geometry_file lap_tied_spring_slip.exo
    omit block 3
END

OUTPUTS
```

```
disp
END

ECHO
mpc
input
END

Tied Joint
    Normal Definition = slip
        surface 1,2
        side = rrod
        connect to Block 33
end

BLOCK 1:3
material "mat"
END

BLOCK 33
   spring
        kx=20776000
        ky=20776000
        kz=26080000
END



HISTORY
 sideset 1,2
 displacement
END


MATERIAL "mat"
E 200e9
nu 0.3
    density 7800
END


BOUNDARY
    sideset 3
       fixed
END

LOADS
    sideset 4
        pressure = -1e3
        function = 1
END

FUNCTION 1
       type LINEAR
       name "const_one"
       data 0.0 1.0
       data 2.0e4 1.0
END
```

9.70. Slide RBE2. Selected DOFs

Refer to Section 7.2 for details of the test.

```
//created with Nasgen from Nastran file 'sliderbe.nas'
SOLUTION
  title=' NEi Nastran Static Analysis Set'
  statics
END

FILE
  geometry_file 'sliderbe.exo'
END

PARAMETERS
  eigen_norm=visualization
END

BOUNDARY
  nodeset 11  x=0
  nodeset 12  y=0
  nodeset 13  z=0
  nodeset 14  Rotx=0
  nodeset 15  Roty=0
  nodeset 16  Rotz=0
END

LOADS
  nodeset 112
    force = 0 1 0
END

OUTPUTS
  disp
END

ECHO
  mass
END

BLOCK 13
  material=1
  thickness=0.5
END

BLOCK 23
  material=1
  thickness=0.375
END

BLOCK 24
  // 25 links
  RBAR
END

MATERIAL 1
  Isotropic
  E = 3e+07
  NU = 0.3
  density = 0.0007324
END
```

9.71. Contact Verification

Refer to Section 7.5 for details of the test.

```
Solution
case static_gap
  statics
end

File
  geometry_file bar_curve_r1000.g
end

Boundary
  nodeset 1
    z=0
  sideset 1
    x=0 y=0
End

Block 1
  material "steelish"
end
Block 2
  material "steelish"
end
Block 3
  material "steelish"
end

Material "steelish"
  isotropic
  density = 0.0343
  nu = 0.0
  E = 29.e6
end
Loads
  body
  gravity = 0 -1 0
  function = 1
end
Function 1
  name "impulse"
  type LINEAR
  data 0 1
  data 1 1
end
Outputs
  disp
  stress
  energy
end
Tied Data
  Surface 101, 100
  search tolerance 0.125
  gap removal = off
end
Tied Data
  Surface 200, 201
  search tolerance 0.125
  gap removal = off
end
```

9.72. Moving Mesh MPCs

Refer to Section 7.6 for details of the test.

```
SOLUTION
    solver=gdsw
  case trans
  transient
    time_step {time_step}   //1.0e-4
    nsteps {nsteps}  //150
    nskip {nskip}  //10
    nUpdateConstraints = 1
    predictorcorrector = 0
END

FILE
  geometry_file = {geometry_file}   //brick_gap.g
END

LINESAMPLE
  samples per line 1000
  endpoint -4  0.  0.   4 0. 0.
  format mfile
END


LOADS
  sideset 1
  acoustic_accel = 0.0
  function 1
END

INITIAL-CONDITIONS
  acoustics = by_block
END

FUNCTION 1
  type linear
  data 0 0
  data 1 0
END

BOUNDARY
  sideset 2
    absorbing
END


OUTPUTS
    apressure
END

ECHO
    input off
END

BLOCK 1
  acoustics 4
  material "air"
END

BLOCK 2
  acoustics 2
  material "air"
END
```

```
MATERIAL "air"
    density 1.293
        acoustic
        c0 332.0
END

begin contact definition
  gap removal = off
  skin all blocks = on
  begin interaction defaults
     general contact = on
  end
end

GDSW
    solver_tol = 1e-8
END
```

This page left blank

This page intentionally left blank.

BIBLIOGRAPHY

[1] Ertas. A., J. T. Krafcik, and S. Ekwaro-Osire. "Performance Of An Anisotropic Allman/DKT 3-Node Thing Triangular Flat Shell Element". In: *Composite Engineering* 2.4 (1992), pp. 269–280 (cit. on p. 241).

[2] D. J. Allman. "A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Problems". In: *Comput. and Struct.* 19.1-2 (1996), pp. 1–8 (cit. on p. 241).

[3] Kenneth F. Alvin. "A method for Treating Discretization Error in Nondeterministic Analysis". In: *AIAA Journal* 99.1611 (1999) (cit. on p. 233).

[4] V. Anes et al. "New approach for analysis of complex multiaxial loading paths". In: *International Journal of Fatigue* 62 (2014), pp. 21–33 (cit. on p. 94).

[5] Jean-Louis Batoz, Klaus-Jurgen Bathe, and Lee-Wing Ho. "A Study of Three-Node Triangular Plate Bending Elements". In: *Int. J. Numer. Meth. Engng.* 15 (1980), pp. 1771–1812 (cit. on p. 241).

[6] Boris Beizer. *Software Testing Techniques*. Intnl Thompson Computer Press, 1990 (cit. on p. 226).

[7] D. Blackstock. "Connection between the Fay and Fubini Solutions for Plane Sound Waves of Finite Amplitude". In: *JASA* 39 (1966), pp. 1019–1026 (cit. on p. 408).

[8] K. Blakely. *MSC/NASTRAN Users Guide: Basic Dynamic Analysis*. Vol. 68. The MacNeal-Schwendler Corporation, 1993 (cit. on p. 40).

[9] Hans Heinrich Bleich and Ivan Seth Sandler. "Interaction between structures and bilinear fluids". In: *International Journal Solids and Structures* 6.5 (1970), pp. 617–639 (cit. on p. 62).

[10] Robert D. Blevins. *Formulas for Natural Frequency and Mode Shape*. Malabar, FL, USA: Krieger, 1984 (cit. on pp. 59, 155, 158, 174, 179, 419).

[11] Gregory Bunting et al. "Parallel Ellipsoidal Perfectly Matched Layers for Acoustic Helmholtz Problems on Exterior Domains". In: *Journal of Computational Acoustics* (2018) (cit. on p. 266).

[12] T. Carne et al. *Finite Element Analysis and Modal Testing of a Rotating Wind Turbine*. Tech. rep. SAND82-0345. Sandia National Laboratories, 1982 (cit. on pp. 161, 162).

[13] J. Chung and G. M. Hulbert. "A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation - The Generalized Alpha Method". In: *JAM* 60.2 (1993), pp. 371–375 (cit. on p. 51).

[14] *COMSOL MEMS Module User Guide, version 5.4*. COMSOL Inc, 2020 (cit. on pp. 113, 115).

[15] R. R. Craig. *Structural Dynamics: An Introduction to Computer Methods*. John Wiley & Sons, 1981 (cit. on p. 424).

[16] J. M. Dickens, J. M. Nagawa, and M. J. Wittbrodt. “A critique of mode acceleration and modal truncation augmentation methods for modal response analysis”. In: *Comput. and Struct.* 62.6 (1997), pp. 985–998 (cit. on p. 38).

[17] Clark R. Dohrmann, S. Key, and M. Heinstein. “A Method for Connecting Dissimilar Finite Element Meshes in Two Dimensions”. In: *Int. J. Numer. Meth. Engng.* 48 (2000), pp. 655–678 (cit. on p. 350).

[18] Clark R. Dohrmann, S. Key, and M. Heinstein. “Methods for Connecting Dissimilar Three-Dimensional Finite Element Meshes”. In: *Int. J. Numer. Meth. Engng.* 47 (2000), pp. 1057–1080 (cit. on p. 350).

[19] A. Ertas, J. T. Krafcik, and S. Ekwaro-Osire. “Explicit Formulation of an Anisotropic Allman/DKT 3-Node Thin Triangular Flat Shell Elements”. In: *Composite Material Technology* 37 (1991), pp. 249–255 (cit. on p. 241).

[20] G. C. Everstine and F. M. Henderson. “Coupled finite element/boundary element approach for fluid/structure interaction”. In: *JASA* 87.5 (1990), pp. 1938–1947 (cit. on p. 358).

[21] C. Felippa and J. DeRuntz. “Finite Element Analysis of Shock-Induced Hull Cavitation”. In: *Computer Meth. in Appl. Mech. Eng.* 44 (1984), pp. 297–337 (cit. on p. 63).

[22] C. A. Felippa. *The SS8 Solid-Shell Element: A Fortran Implementation*. Tech. rep. CU-CAS-02-04. Univ. Colo. at Boulder, 2002 (cit. on p. 192).

[23] C. A. Felippa. *The SS8 Solid-Shell Element: Formulation and a Mathematica Implementation*. Tech. rep. CU-CAS-02-03. Univ. Colo. at Boulder, 2002 (cit. on pp. 192, 238, 240).

[24] F. Fuentes et al. “Orientation embedded high order shape functions for the exact sequence elements of all shapes”. In: *Computers and Mathematics with Applications* 70.1 (2015), pp. 353–458 (cit. on p. 28).

[25] N. Guo, P. Cawley, and D. Hitchings. “The finite element analysis of the vibration characteristics of piezoelectric discs”. In: *Journal of Sound and Vibration* 159.1 (1992), pp. 115–138 (cit. on pp. 117, 118).

[26] M. F. Hamilton and D. T. Blackstock. *Nonlinear Acoustics*. Academic Press, 1998 (cit. on p. 407).

[27] Daniel C. Hammerand. *Laminated Composites Modeling In Adagio/Presto*. Tech. rep. SAND2004-2143. Sandia National Laboratories, May 2004 (cit. on pp. 250, 252).

[28] J. Hoffelner, H. Landes, and R. Lerch. “Calculation of Acoustic Streaming Velocity and Radiation Force Based on Finite Element Simulations of Nonlinear Wave Propagation”. In: *Proceedings of IEEE Ultrasonics Symposium* 1 (2000), pp. 585–588 (cit. on p. 407).

[29] J. Hoffelner et al. “Finite Element Simulation of Nonlinear Wave Propagation in Thermoviscous Fluids Including Dissipation”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 48.3 (2001), pp. 779–786 (cit. on p. 407).

[30] Hanson Huang. “Transient Interaction of Plane Acoustic Waves with a Spherical Elastic Shell”. In: *Journal of the Acoustical Society of America* 45.3 (1969), pp. 661–670 (cit. on pp. 371, 376).

[31] Thomas J. R. Hughes. *The Finite Element Method–Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc, 1987 (cit. on p. 173).

[32] Kinsler et al. *Fundamentals of Acoustics*. John Wiley & Sons, 1982 (cit. on pp. 259, 347, 391, 415, 416).

[33] V. P. Kuznetsov. "Equations of Nonlinear Acoustics". In: *Sov. Phys. Acoust.* 16 (1971), pp. 467–470 (cit. on p. 407).

[34] Richard H. MacNeal and Robert L. Harder. "A Proposed Standard Set of Problems to Test Finite Element Accuracy". In: *Finite Elements in Analysis and Design* 1 (June 1985), pp. 3–20 (cit. on pp. 226, 227, 423).

[35] J. L. Meriam and L. G. Kraige. *Engineering Mechanics Volume 2 - Dynamics*. John Wiley & Sons, 1986 (cit. on p. 434).

[36] Morse and Ingard. *Theoretical Acoustics*. Mcgraw-Hill Book Company, 1968 (cit. on p. 393).

[37] Thomas Moyer et al. "Navy Enhanced Sierra Mechanics (NESM): Toolbox for Predicting Navy Shock and Damage". In: *Computing in Science and Engineering* 18.6 (2016), pp. 10–18 (cit. on p. 157).

[38] Glenford J. Myers. *The Art of Software Testing*. John Wiley & Sons, 1979 (cit. on p. 226).

[39] A. D. Pierce. *Acoustics: An Introduction to Its Physical Principles and Applications*. ASA, 1989 (cit. on pp. 283, 336, 347, 350, 361, 396, 398, 407, 408).

[40] Douglass Post et al. "The Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program". In: *Computing in Science and Engineering* 18.6 (2016), pp. 7–9 (cit. on p. 157).

[41] L. F. Richardson. "The Approximate Arithmetic Solution by Finite Differences of Physical Problems Involving Differential Equations, With Applications to the Stresses in a Masonry Dam". In: *Philosophical Transactions of the Royal Society of London*. A 210 (1910), pp. 307–357 (cit. on p. 233).

[42] S D Team. *SD – User's Manual*. Tech. rep. SAND2021-12518. living document with more recent versions. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratories, 2022 (cit. on p. 35).

[43] S D Team. *Sierra SD Example Problems Manual*. Tech. rep. SAND2023-09500. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratories, 2023 (cit. on p. 35).

[44] S D Team. *Sierra Structural Dynamics - Theory Manual*. Tech. rep. SAND2023-09413. living document with more recent versions. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratory, 2023 (cit. on p. 350).

[45] Daniel J. Segalman. *A Four-Parameter Iwan Model for Lap-Type Joints*. Tech. rep. SAND 2002-3828. Sandia National Laboratories, Nov. 2002 (cit. on p. 253).

[46] Francis J. Shaker. *Effect of Axial Load on Mode Shapes and Frequencies of Beams*. Tech. rep. D-8109. Washington, D.C.: Lewis Research Center, National Aeronautics and Space Administration, 1975 (cit. on pp. 161, 162).

[47] MSC Software. *MSC/NASTRAN 2017 Superelements User's Guide*. The MacNeal-Schwendler Corporation, 2016 (cit. on p. 217).

[48] M. A. Sprague and T. L. Geers. "Response of Empty and Fluid-Filled, Submerged Spherical Shells to Plane and Spherical, Step-Exponential Acoustic Waves". In: *Shock and Vibration* 6 (1999), pp. 147–157 (cit. on p. 378).

[49] S. Timoshenko. *Strength of Materials*. 3rd ed. D. Van Nostrand Company, Inc., 1955 (cit. on p. 238).

[50] S. Timoshenko. *Theory of Elastic Stability*. McGraw Hill Book Company, 1936 (cit. on p. 65).

[51] Timothy G. Trucano, Martin Pilch, and William L. Oberkampf. *On the role of Code Comparisons in Verification and Validation*. Tech. rep. SAND 2003-2752. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratories, 2003 (cit. on p. 423).

[52] W. Weaver, Jr. and S. P. Timoshenko. *Vibration Problems in Engineering*. 5th ed. John Wiley and Sons, 1990 (cit. on p. 38).

[53] Warren C. Young. *Roark's Formulas for Stress & Strain*. 6th ed. McGraw-Hill Book Company, 1989 (cit. on pp. 191, 238, 442).

[54] John D. Zepper et al. *ASCI Applications Software Quality Engineering Practices*. Tech. rep. SAND2002-0121. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratories, 2002 (cit. on p. 3).

[55] X D Zhang and C T Sun. "Formulation of an Adaptive Sandwich Beam". In: *Smart Materials and Structures* 5.6 (1996), pp. 814–823 (cit. on pp. 110, 111).

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	Technical Library	1911	0845

This page intentionally left blank.

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.